

SCADA Sistemi ile Yazılım Doğrulama Araç Entegrasyonu

Rahima Ibrahimova

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran 2023

Software Verification Tool Integration with SCADA Systems

Rahima Ibrahimova

MASTER OF SCIENCE THESIS

Department of Computer Engineering

June 2023

SCADA Sistemi ile Yazılım Doğrulama Araç Entegrasyonu

Rahima Ibrahimova

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Eğitim ve Öğretim Yönetmeliği Uyarınca

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Yazılımı Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Doç. Dr. Metin Özkan

Haziran 2023

ONAY

Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Rahima IBRAHİMOVA'nın YÜKSEK LİSANS tezi olarak hazırladığı "SCADA Sistemi ile Yazılım Doğrulama Araç Entegrasyonu" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oybirliği ile kabul edilmiştir.

Danışman: Doç. Dr. Metin Özkan

İkinci Danışman: -

Yüksek Lisans Tez Savunma Jürisi:

Üye: Doç. Dr. Metin Özkan

Üye: Dr. Öğr. Üyesi Uğur Gürel

Üye: Dr. Öğr. Üyesi Altan Onat

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet Erşahan

Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Metin Özkan danışmanlığında hazırlamış olduğum “**SCADA Sistemi ile Yazılım Doğrulama Araç Entegrasyonu**” başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 18/06/2023

Rahima
Ibrahimova

ÖZET

SCADA (Gözetleyici Kontrol ve Veri Toplama) sistemleri, otomatik sistemlerin uzaktan izlenmesi, kontrol edilmesi ve veri kaydı için yaygın olarak kullanılmaktadır. Ancak, bu sistemlerin güncel teknolojilerle desteklenmesi ve yeni kabiliyetler eklenmesi ile daha verimli kullanılması mümkün olabilir. Diğer taraftan otomatik sistemlerin ilk geliştirilmesi sürecinde ve kullanılmaya başlandıktan sonra yeni isterlere göre güncellemelerin yapılmasında yazılım doğrulaması ve onaylamasının yapılması önemli bir konudur. Doğrulama ve onaylama yazılım araçları, sistemle ilgili veriler üretmekte ve bu verilerin uygun grafiksel gösterimlerle sistem geliştiricilerine sunulması gerekmektedir.

Bu tez çalışmasında yeni bir yaklaşım ortaya konulmaktadır. Yeni geliştirecek bir otomatik sistemle birlikte kullanılacak olan SCADA sistemine, sistemin yazılım doğrulaması ve onaylamasında kullanılacak araçları entegre etmek önemli bir katkı sağlayabilir. Yeni sistemin ilk geliştirme sürecinde, doğrulama ve onaylama araçlarının veri toplama ve izlemesinde SCADA sistemi kullanılabilir. Aynı zamanda, sistemde ilk kurulum sonrasında yapılacak güncellemeler için doğrulama ve onaylama bileşeni hazır olarak bulunabilir. Belli zaman periyodlarında sistem doğrulaması ve onaylaması gerçekleştirilerek sistemin güvenilir kalması sağlanabilir.

Yapılan çalışmalarla, önerilen yaklaşımın gerçekleştirme yöntemi anlatılmış ve gerçek sistem testleri yapılmıştır. Gerçek sistem testlerinde, laboratuvar ortamında kullanılan robotik denetleme sisteminde faydalanılmıştır.

SCADA sistemi, robotik denetleme sisteminin doğrulama ve onay sürecinde izleme amacıyla kullanılmıştır. Çalışma ROS ortamında gerçekleşmiştir. Verilerin SCADA sisteminin kabul ettiği protokole dönüştürülmesi için yapılan çalışmada OPC UA kullanılmıştır.

Anahtar kelimeler: SCADA, ROS, OPC, OPC UA, Yazılım Doğrulama

SUMMARY

SCADA (Supervisory Control and Data Acquisition) systems are widely used for remote monitoring, control and data logging of automated systems. However, it may be possible to use these systems more efficiently by supporting them with up-to-date technologies and adding new capabilities. On the other hand, software verification and validation is an important issue in the initial development of automated systems and in making updates according to new requirements after they are put into use. Verification and validation software tools generate data about the system and these data should be presented to the system developers with appropriate graphical representations.

In this thesis, a new approach is presented. Integrating the tools to be used in the software verification and validation of the system into the SCADA system to be used with a new automated system can make a significant contribution. During the initial development process of the new system, the SCADA system can be used for data collection and monitoring of verification and validation tools. At the same time, the verification and validation component can be readily available for updates to the system after initial installation. System verification and validation can be performed at certain time periods to ensure that the system remains reliable.

In the studies, the implementation method of the proposed approach is explained and real system tests are performed. In the real system tests, the robotic inspection system used in the laboratory environment was utilized.

The SCADA system was used to monitor the robotic inspection system during the verification and validation process. The study was carried out in the ROS environment. OPC UA was used in the study to convert the data into the protocol accepted by the SCADA system.

Keywords: SCADA, ROS, OPC, OPC UA, Software Verification

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde, değerli bilgilerini benimle paylaşan, kendisine ne zaman danışsam bana kıymetli zamanını ayırıp sabırla ve büyük bir ilgiyle bana faydalı olabilmek için elinden gelenden fazlasını sunan ve danışman hoca statüsünü hakkıyla yerine getiren değerli tez danışmanım sayın Doç. Dr. Metin Özkan'a teşekkür ederim.

Bu yüksek lisans programının her aşamasında beni yalnız bırakmayan, hep destekleyen aileme sonsuz sevgi ve saygılarımı sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xi
ÇİZELGELER DİZİNİ.....	xiii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiv
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	3
2.1. SCADA Sistemleri	3
2.2. Otomatik Sistemlerin Doğrulaması.....	6
3. SİSTEM TASARIMI.....	9
3.1. SCADA Sistemi	10
3.1.1. SCADA Sistemleri Temel Bileşenleri	12
3.1.1.1. <u>Kontrol Merkezi Birimi</u>	12
3.1.1.2. <u>Uzak Terminal Birimi</u>	12
3.1.1.3. <u>Programlanabilir Mantık Denetleyici</u>	13
3.1.1.4. <u>İnsan Makine Arayüzü</u>	14
3.1.2. Açık Kaynaklı SCADA Yazılımları	15
3.2. Robot İşletim Sistemi (Robot Operating Systems, ROS)	17
3.3. Açık Platform Haberleşmesi – Birleşik Mimarisi (OPC-UA)	21
3.4. OPC UA ile ROS Arası Veri Aktarım Köprüsü (OPC UA- ROS Bridge)	23
3.5. Robotik Sistemler için Yazılım Doğrulama Aracı (Model-Aided Runtime Verification for Robotic Systems, MARVer).....	23

İÇİNDEKİLER

Sayfa

4. MATERYAL VE YÖNTEM.....	25
4.1. Robotik Denetim Sistemi	26
4.2. Doğrulama ve Onaylama Sistemi	27
4.3. ROS ile OPC UA Arasında Adaptör Düğüm Geliştirme	29
4.4. SCADA Tablo ve Şemaların Oluşturulması	32
5. BULGULAR VE TARTIŞMA.....	42
6. SONUÇ VE ÖNERİLER.....	53
KAYNAKLAR DİZİNİ	54

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.1. Sistem Mimarisi	9
3.2. SCADA Sisteminin Genel Blok diyagramı	11
3.3. PLC Genel Mimarisi	14
3.4. Endüstriyel IoT Uygulamalarında Standartlaştırılmış Haberleşme Protokolü olarak OPC UA Kullanımı.....	22
3.5. MARVer Mimarisi.....	24
4.1. Sistem kurulumunda izlenen adımlar.....	25
4.2. Robotik Denetim Sistemi.....	26
4.3. MARVer Doğrulama Yazılımı Arayüzü.....	28
4.4. ROS OPC UA haberleşme paketi ve adaptör düğümün şematik gösterimi	29
4.5. ROS ile OPC UA arası adaptör düğüm akış diyagramı	31
4.6. Rapid SCADA uygulama adımları	32
4.7. Yeni Proje Yaratmak.....	32
4.8. Hat ve Cihaz yaratmak	33
4.9. Cihazda atılacak adımlar	33
4.10. Cihaz özellikleri	34
4.11. Verinin sisteme girilmesi.....	34
4.12. Tablo ve Şema klasörleri yaratmak	36
4.13. Interface'e klasörlerin girilmesi	37
4.14. Şemada dinamik resmin yaratılması.....	37
4.15. Şemada dinamik resmin girilmesi	38
4.16. Resmin girilme ayrıntıları.....	38

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
4.17. Tabloya verilerin girilmesi	39
4.18. ROS'dan gelen verinin girilmesi	39
4.19. Kanal yaratmak.....	40
4.20. Kanal yaratılmasında adımla	40
4.21. Rapid SCADA'yı güncellemek	41
4.22. Localhost'a bağlanmak.....	41
5.1. Robotik Denetleme Sisteminin, Denetleme Görevi Esnasında Çekilen Video Kareleri	45
5.2. SCADA Tasarı Görünümü	46
5.3. SCADA Tablo Görünümü: (a) Tüm ekran görünümü, (b) Tablo Kısmı Yakın.....	47
5.4. Robot kolun eklem pozisyonlarının zamana göre grafiği	48
5.5. Robot kolun eklem hızlarının zamana göre grafiği	49
5.6. Robot kolun eklem ivmelerinin zamana göre grafiği	49
5.7. Robot kolun eklem jerklerinin zamana göre grafiği.....	50
5.8. Mobil platformun pozisyonunun zamana göre grafiği	50
5.9. Doğrulama Sonucunun Zamana Göre Diyagramı	51

ÇİZELGELER DİZİNİ

Cizelge

Sayfa

4.1. Robotik Denetim Sistemi SCADA sisteminde kullanılacak verileri.....	27
4.2. MARVer yazılım doğrulama yazılımının SCADA sisteminde kullanılacak verileri.....	29
4.3. Verilerin girilmesinde başlıkların anlamı.....	35
5.1. Çalışma Anı Doğrulama İsterleri	52



SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

Acıklama

SCADA	Gözetleyici Kontrol ve Veri Toplama (Supervisory Control and Data Acquisition)
ROS	Robot İşletim Sistemi (Robot Operating System)
OPC UA	Açık Platform Haberleşmesi Birleşik Mimari (Open Platform Communications Unified Architecture)
SOA	Hizmet Odaklı Mimari (Service Oriented Architecture)
IDS	Saldırı Tespit Sistemi (Intrusion Detection System)
IDE	Entegre Geliştirme Ortamı (Integrated Development Environment)
IAS	Endüstriyel Uygulama Yazılımı (Industrial Application Software)
IOT	Nesnelerin İnterneti (Internet of Things)
IIOT	Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things)
IP	İnternet Protokolü (Internet Protocol)
HTTP	Üst Metin Transfer Protokolü (Hyper Text Transfer Protocol)
LAN	Yerel Alan Ağı (Local Area Network)
WAN	Geniş Alan Ağı (Wide Area Network)
ICS	Endüstriyel Kontrol Sistemleri (Industrial Control Systems)
MTU	Kontrol Merkezi Birimi (Main Terminal Unit)
RTU	Uzak Terminal Birimi (Remote Terminal Unit)
PLC	Programlanabilir Mantık Denetleyici (Programmable Logic Controller)
HMI	İnsan Makine Arayüzü (Human-Machine Interface)
HRI	İnsan Robot Etkileşimleri (Human Robot Interactions)
V&V	Doğrulama ve Onaylama (Verification and Validation)
SDN	Yazılım Tanımlı Ağ (Software Defined Networking)

1. GİRİŞ VE AMAÇ

Günümüzde neredeyse tüm şirketler, devlet kurumları, müesseseler ve fabrikalarda yürütülen imalat ve hizmet süreçlerinin uzaktan gözetimi, denetimi ve verilerin toplanması önemli konulardan biri olarak görülmektedir. Bu süreçlerin gözetimi ve denetimi, ortaya çıkan ya da çıkabilecek sorunların hızlı bir şekilde giderilmesini sağladığı gibi sistemlerin denetimini de kolaylaştırmaktadır. Ayrıca toplanan veriler, sistemlerin analiz edilmesi, kalite artırıcı önlemlerin alınmasına da katkı sağlamaktadır. Sistemlerin gözetimi ve denetimi için birçok yöntem ve uygulamalar kullanılmaktadır. Özellikle, endüstriyel süreçlerde üretim ve imalat, su arıtma, enerji dağıtım, iletişim sistemleri, ısıtma ve havalandırma, tüneller ve trafik sistemleri gibi çeşitli alanlarda otomatik sistemlerin gözetimi ve denetiminde SCADA (Gözetleyici Kontrol ve Veri Toplama) sistemleri kullanılmaktadır.

SCADA sistemlerinin kullanıldığı otomatik sistemlerde yazılım ve donanım kombinasyonları kullanılmaktadır. Bu sistemlerin donanım ve yazılımları, insan operatörlerine ihtiyaç duymaksızın otomatik çalışmak üzere tasarlanmakta ve programlanmaktadır. Bu sistemler, birkaç sistem ve çevresel olayları girdi olarak alır; koşullu karar verme ve belirli denetim mantığı bağlı olarak işlemler gerçekleştirir. Otomatikleştirilmiş olan sistemlerin sağladığı önemli bazı faydalar bulunmaktadır. Öncelikle, insan hatalarından kaynaklı ortaya çıkan riskler ve zararlar ortadan kaldırılmış olur. Üretkenlikte önemli derecede bir artış söz konusu olur. Otomatikleştirilmiş sistemin kurulduğu uygulama alanında, standartlaştırılmış operasyonların gerçekleştirilmesi mümkün olur. Operasyonların daha iyi yönetimi ve kayıt altına alınması mümkün olur. İş gücü, zaman ve maliyet açısından önemli derecede kazanım olması yanında, yapılan işlerin doğruluk ve kesinliği artmaktadır. Ayrıca, sunulan hizmetlerin performansı ve güvenilirliği de artar.

Otomatikleştirilmiş sistemlerin sağladığı çok fazla fayda vardır. Ancak, bu sistemlerin tasarımları ve gerçeklemeleri esnasında ve sonrasında doğrulamaları gerekmektedir. Her ne kadar sistemin operasyonu esnasında, elle çalışan sistemlere kıyasla insan faktörüne bağlı ortaya

ıkabilecek hatalar ortadan kalkmıř olsa da sistemin geliřtirilmesi esnasında yapılan ve fark edilemeyen hatalar, sistem devreye alındıktan sonra ortaya ıkararak hem emniyet sorunları hem de nemli maddi zararlara neden olabilir. Bu sorunlarla karřılařmamak iin, sistem yazılımına doęrulama ve onaylama yntemleri uygulamak gerekir. Bylece, sistem henz devreye alınmadan nce ortaya ıkabilecek sorunlar giderilebilir ya da en aza indirilebilir.

Bu tez alıřmasında, yeni bir yaklařım ortaya konulmakta ve yapılan uygulamalı alıřmalar ile bu yaklařımın uygulanabilirlięi ve saęlayabileceęi faydalar gsterilmektedir. Bu yaklařımda, yeni geliřtirilen veya mevcut otomatik sistemler iin SCADA sistemi kurulumu yapılırken, aynı zamanda sistem yazılım doęrulama ve onaylama aralarının da entegre edilmesinin nemli avantaj saęlayacaęı hipotezini ortaya koymaktadır. Bylece, yeni kurulan sistemin doęrulama ve onaylama srelerinde SCADA sistemi gzetim amalı kullanılabilecektir. Bunun yanında, yeni isterlere ve iyileřtirme taleplerine istinaden sistem yazılımı zerinde yapılacak revizyon ve gncellemelerin doęrulama ve onaylama sreci iin, mevcut doęrulama aralarının entegre olduęu SCADA sistemi kullanılabilecektir.

nerilen bu yeni yaklařımın uygulamasını gerekleřtirmek zere, Otomatik Sistemlerin Emniyet ve Gvenlięinin Doęrulama ve Onaylaması (VALU3S, 2023) olarak adlandırılan ECSEL JU projesindeki bir uygulama ortamı kullanılmıřtır. Bu uygulama ortamında, retim hattında otobs iskelet yapısının kalite denetlemesi iin kullanılan, robotik denetleme sistemi kullanılmaktadır. Bu sistemi referans alarak, laboratuvar ortamında kurulan robotik denetleme sistemi kullanılmıřtır (IFARLAB, 2023). Aık kaynak kodlu bir SCADA yazılımı ve MARVer (Demirci vd., 2022; Demirci, 2023) adlı doęrulama yazılım aracının entegrasyonu gerekleřtirilmiřtir.

Bu tez alıřmasının Blm 1’de alıřmanın amacı ve kısa bilgi verilmiřtir. Blm 2’de nerilen yaklařımla iliřkili literatrde yapılan alıřmalar sunulmaktadır. Blm 4’de nerilen yaklařımın detaylı anlatımı verilmektedir. nerilen yaklařımla ilgili yapılan uygulama ve sonuları Blm 5’de verilmektedir. Blm 6’da elde edilen sonularla ilgili deęerlendirme ve gelecekte yapılabilcek alıřmalar bulunmaktadır.

2. LİTERATÜR ARAŞTIRMASI

2.1. SCADA Sistemleri

SCADA sistemleri ile ilgili literatürde çok sayıda çalışma vardır. Bu çalışmalar, SCADA sistemlerinde güvenlikten, öngörülü bakım gibi fonksiyonelliğin artırılması gibi geniş bir alanda yürütülmüştür.

SCADA sistemleri, otomatik sistemlere uzaktan erişim sağlayarak, gözetleyici kontrol ve veri toplama gerçekleştirilmektedir. Uzak erişim söz konusu olduğu için, ağ üzerinde güvenlik sorunları ortaya çıkabilir. Bu nedenle, güvenlik ile ilgili çok sayıda güncel yayın bulunmaktadır.

Gao vd. (2014) gerçekleştirdikleri çalışmada, literatürdeki çok sayıdaki belgelenmiş SCADA standartlarını, iletişim ve güvenlik anlamında son teknoloji seviyesini incelemekte, güvenlik sorunlarını ve çözümlerini tartışmışlardır. McLaughlin vd. (2016) yaptıkları çalışmada, endüstriyel kontrol sistemlerinin geleneksel elektromekanik tabanlı sistemlerden modern bilgi ve iletişim teknolojileri tabanlı sistemlere dönüştüğü vurgulanmaktadır. Bu durumda siber güvenlik önemli bir konu haline geldiği ve SCADA sistemlerinin de bu konu kapsamında ele alınması gerektiğini belirtmektedir. Nazir vd. (2017) çalışmalarında, SCADA sistemleri için siber atak kırılganlığını değerlendirmek için kullanışlı benzetim, modelleme ve benzer tekniklerin araştırmasını yapmışlardır. Giraldo vd. (2017), siber-fiziksel sistemler alanında güvenlikle ilgili ataklar, savunmalar, araştırma istikametleri, ağ güvenliği ve güvenlik seviyesi gerçekleştirme olarak sınıflandırma yaparak literatür detaylı taramasını sunmaktadır. Bu kapsamda, bu sistemlerin bir bileşeni olarak SCADA sistemleri için de ele almaktadır. Ghosh ve Sampalli, (2019), uzaktan izleme amaçlı kullanılan SCADA sistemlerde açık erişim ağlarının kullanımının artması ile birlikte güvenlik tehditleri karşı kaşıya kalındığı belirtilmektedir. Yaptıkları çalışmada, SCADA sistemlerinde devam eden güvenlik araştırmalarının karşılaştırmalı incelemesini sunmaktadır. Güvenlik istekleri ve ağ protokol katmanlarına bağlı saldırıların

sınıflandırmasını sunmaktadır.

Bere ve Muyingi (2015), SCADA sistemleri ve dağıtık kontrol sistemleri ile birlikte endüstriye kontrol sistemlerinin, endüstriyel süreç kontrolünde kullanıldığını belirtmektedir. Bu sistemleri, artık diğer bilgi sistemleri ile bağlantılı hale getirildiği söylenmektedir. Bu durumda bu sistemlerin ileri seviye kalıcı tehditlere maruz kaldığı belirtilmektedir. Bu probleme karşı olası çözümlerin analizini yapmaktadırlar. Ayrıca, yapay bağışıklık sistemlerinin kullanımını önermektedirler.

Chan ve Zhou (2022), akıllı ağ yapılarının gelişmesi ile birlikte SCADA sistemlerinin açık sistemlere bağlı olma ihtiyacı her geçen gün arttığını ve bunun sonucunda güvenlik önemli hale geldiğini belirtmektedir. Ancak, SCADA sistemlerinin daha önce güvenlik dikkate alınmadan geliştirdiğini söylemektedirler. Güvenlik dikkate alınmadan dağıtım yapılan SCADA sistemleri için çözümün eklentiler olduğu söylenmektedir. Yaptıkları çalışmada bunu dikkate almaktadırlar.

Erez ve Wool (2015) çalışmalarında, ModBus/TCP SCADA kontrol yazmaçlarında (registers) kurlsız değişiklikleri algılayan bir anomali tespit sistemi tasarladı. Bu kurlsız değişiklikler için, sensör, sayıcı ve sabit yazmaçlarına odaklanmıştır. Öğrenme tabanlı sınıflandırıcı ile bu sınıflar belirlenmektedir.

SCADA sistemleri için izinsiz girişler de önemli güvenlik sorunu oluşturmaktadır. Bu konuda da yapılan çalışmalar bulunmaktadır. Shitharth ve Winston (2017) çalışmalarında, SCADA ağlarındaki izinsiz girişleri tespit etmek için iki algoritma sunmuşlardır. İsmi geçen ilk algoritma Guguk Kuşu Aramasına Dayalı Saldırı Ağırlıklı Parçacık Optimizasyon (IWP-CSO), veri setinden elde edilen öznitelikleri çıkarmak ve optimize etmek için kullanılır. İkinci Hiyerarşik Nöron Mimarisi tabanlı olarak adlandırılan algoritma Yapay Sinir Ağı (HNA-NN), optimize edilmiş özelliklere göre sınıflandırma yapmak için kullanılır. Performans değerlendirme simülasyon ortamında gerçekleştirilmiş ve farklı veri kümeleri olarak kabul edilmiştir. Önerilen kombinasyonu algoritmalar %93,1 doğruluk oranına ulaşmışlardır.

Wang vd. (2014) tarafından, SCADA sistemlerini tehdit eden kötü amaçlı yazılımlarca izinsiz giriş için hatalı veri enjeksiyon atakları yapılabildiği belirtilmekte ve gerçek zamanlı bir tespit sistemi önerilmektedir. Önerilen sistemin değerlendirmesini yapmak üzere, gerçek dünya elektrik santrali simülatörü kullanılmaktadır.

Yang vd. (2014) yaptıkları çalışmada, yeni jenerasyon SCADA sistemleri için ihlal algılama sistemi için yeni bir yaklaşım sunmaktadırlar. SCADA spesifik siber güvenlik test ortamında önerilen yaklaşımı doğrulamaktadırlar. Wong vd. (2017), güvenlik ihlal algılama sistemlerinde SCADA protokolleri için katkı sağlamaktadır.

Ayrıca, literatürde SCADA sistemleri ile ilgili uygulamalarda bulunmaktadır. Qays vd. (2022) yaptıkları çalışmada, yenilenebilir enerji sistemlerinin izlenmesinde, IoT destekli SCADA sistemi kullanmışlardır. SCADA sistemine sahadaki verileri taşımak üzere maliyet etkin tasarım için düşük maliyetli elektronik bileşenler kullanılmış ve Arduino uzak terminal birimi haline getirilmiştir. Benzetim ve gerçek ortam deneysel sonuçları ile önerilen yaklaşımın kullanılabilirliği gösterilmiştir.

Babay vd. (2018) çalışmalarında, Spire adlı yeni bir mimari geliştirmişlerdir. SCADA kontrol merkezinin kopyalarını dağıtan aynı zamanda da siber saldırılara karşı dayanıklılığı artıran bir sistem kurmuşlardır. Geliştirdikleri Spire mimarisi, iki kontrol merkezi ve iki veri merkezinden oluşmuştur.

Aghenta ve Iqbal (2019), güneş enerjisi sistemlerin izlenmesi ve uzaktan kontrolü için düşük maliyetli, açık kaynaklı bir SCADA sistemi önermişlerdir. Önerilen sistem, IoT tabanlı SCADA mimarisine dayanmaktadır. Geleneksel SCADA sistemleri ile web servisleri üzerinden iletişim kurmaktadır. Arduino ve raspberry pi gibi düşük maliyetli bilgisayar kartları uzak terminal birimleri olarak kullanılmaktadır.

Ndukwe vd. (2022) çalışmalarında, veri iletimi için bir iletişim kanalı olarak LoRa iletişim protokolünü kullanarak dağıtılmış yenilenebilir enerji üretim sistemlerini izlemek ve

kontrol etmek için IoT mimarisine dayalı açık kaynaklı, düşük maliyetli bir SCADA sisteminin tasarımını ve geliştirilmesini önermişlerdir. Sistem, tüm parametrelerin çok uzun süreli depolanmasına izin veren bir zaman serisi veri tabanı olan InfluxDB'deki bir IOT veri tabanı tarafından desteklenmektedir. Bu sistem, veri alımı, iletimi, veri yönetimi, veri gösterimi ve sistem kontrolü dahil olmak üzere SCADA işlevlerinin gerçekleştirildiğini göstermektedir.

SCADA sistemleri, veri toplama ve denetim amaçlı kullanımı nedeniyle, toplanan veriler ile sistem iyileştirmesi için faydalı sonuçlar çıkarılabilir. Buna istinaden, literatürde bazı çalışmalar, SCADA sistemlerin kullanımı ile arıza tespitine odaklanmaktadır. Ikhsan vd. (2019) çalışmalarında, okul binalarında elektriksel hataları tespit etmek üzere güç dağıtım sistemini izleyen ve kontrol eden SCADA sistemi kurarak çözüm üretmektedir. Önerile yaklaşımın deneysel doğrulaması için Arduino ATmega2560 kullanmışlardır.

2.2. Otomatik Sistemlerin Doğrulaması

Kurulan yeni bir sisteme ya da geliştirmeler yapılan mevcut sistemlere, doğrulama ve onaylama süreçlerinin uygulanması gerekmektedir. Literatürde, otomatik sistem yazılımlarının doğrulanması ve onaylanmasına yönelik çok sayıda yayın bulunmaktadır. Çalışmaların bazılarında, yeni teknikler geliştirilirken, bazı çalışmalarda da yazılım araçları ve farklı araç ve yöntemleri içeren melez yaklaşımlar geliştirilmektedir. Bu bölümde, farklı doğrulama ve onaylama yöntemleri, araçları ve yaklaşımları incelenmektedir.

Zheng ve Julien (2015), siber fiziksel sistemlerdeki doğrulama ve onaylama ile ilgili araştırma güçlüklerini ve çalışmaların yönelimlerini değerlendirmektedir. Ingrand, (2019), otonom sistemlerin yazılımlarındaki başarısızlıkların dramatik bir duruma gelebildiğini belirtmektedir. Ancak, bunun karanlık bir resim oluşturmasına gerek olmadığı, otonom sistemlerin en kritik bileşenleri yazılımlarının gürbüz ve güvenilir olduğunu göstererek ve belgeleyerek ispatlamanın yeterli olacağı ifade edilmektedir. Formal doğrulama ve onaylama kullanımı ile bu belgelemenin yapılabileceği belirtilmektedir. Bu çalışmada, otonom robot yazılımlarında formal doğrulama ve onaylamadaki güncel eğilim ortaya konulmaktadır. Antao

vd. (2018), mimarisine ve endüstriyel gerçeklik tarafından belirlenen gereksinimlere dayalı olarak IIoT (Endüstriyel Nesnelerin İnterneti) platformlarının doğrulanması için bir gereksinim listesi önermektedir. Ölçeklenebilirlik, güvenilirlik, güvenlik ve gizlilik, zamanlama, güvenlik, kurtarma, birlikte çalışabilirlik ve yeniden yapılandırılabilirlik gibi yürütülmeleriyle ilgili ana zorlukları tanımlayarak, bunları IoT katmanlarıyla ilişkilendirdikleri 8 Siber Fiziksel Üretim Sistemleri (CPPS) gereksiniminin doğrulanmasını ve test edilmesini önermektedirler.

Luckcuck vd. (2019), otonom sistemlerin karmaşık, melez ve genellikle emniyet açısından kritik olduğunu ve bu durumun formal tanımlama ve doğrulama için güçlük oluşturduğunu belirtmektedir. Çoğunlukla başvuru test yapma ve benzetim tekniklerinin tek başına doğrulama ve belgeleme için yetersiz olduğunu söylemektedir. Bu çalışmada, otonom robotik sistemlerin forma doğrulaması için güncel durumun incelenmesi ve uygun sınıflandırma ile sunumu yapılmaktadır.

Webster vd. (2019), İnsan-Robot Etkileşimleri (HRI) bağlamında robot asistanların doğrulayıcı V&V'si olarak adlandırılan ve birkaç farklı V&V tekniğini birleştiren bir yaklaşım sunmaktadır. Yaklaşımlarında, gerçek bir robotla yapılan deneylerde biçimsel doğrulama, simülasyon tabanlı test ve kullanıcı doğrulaması olmak üzere üç farklı teknik kullanmışlardır. Bu makalede, HRI bağlamında gerçek bir robotla yapılan deneyler aracılığıyla biçimsel yöntemleri, simülasyon tabanlı testleri ve kullanıcı doğrulamasını birleştirmektedirler.

Sinha vd. (2019), güvenilir Endüstriyel Otonom Sistemler (IAS) oluşturmak için statik, resmi tekniklerin bir incelemesini sunmaktadır. Biçimsel yöntemleri, güvenilir sistemlerin geliştirilmesine büyük ölçüde yardımcı olabilecek ve gereksinim mühendisliği, sistem tasarımı ve uygulaması, doğrulama ve geçerleme (test), bakım ve hatta belgelendirme dahil olmak üzere sistem geliştirme yaşam döngüsünün (SDLC) tüm aşamalarında kullanılabilen matematiksel teknikler olarak tanımlamaktadırlar. Sistemin işlevsel olarak doğru olması ve aynı zamanda güvenilirlik, emniyet ve güvenlik gibi işlevsel olmayan gereksinimleri (NFR'ler) karşılaması gerektiğini belirtirler. Bu çalışmanın kapsamını, çevrimiçi veya çalışma zamanı yaklaşımlarıyla değil, yalnızca güvenilirlik için statik veya çevrimdışı yaklaşımlarla sınırlandırmışlardır. Testleri üç konu olan biçimsel yöntemler, simülasyon ve biçimsel doğrulama ile incelemiş ve

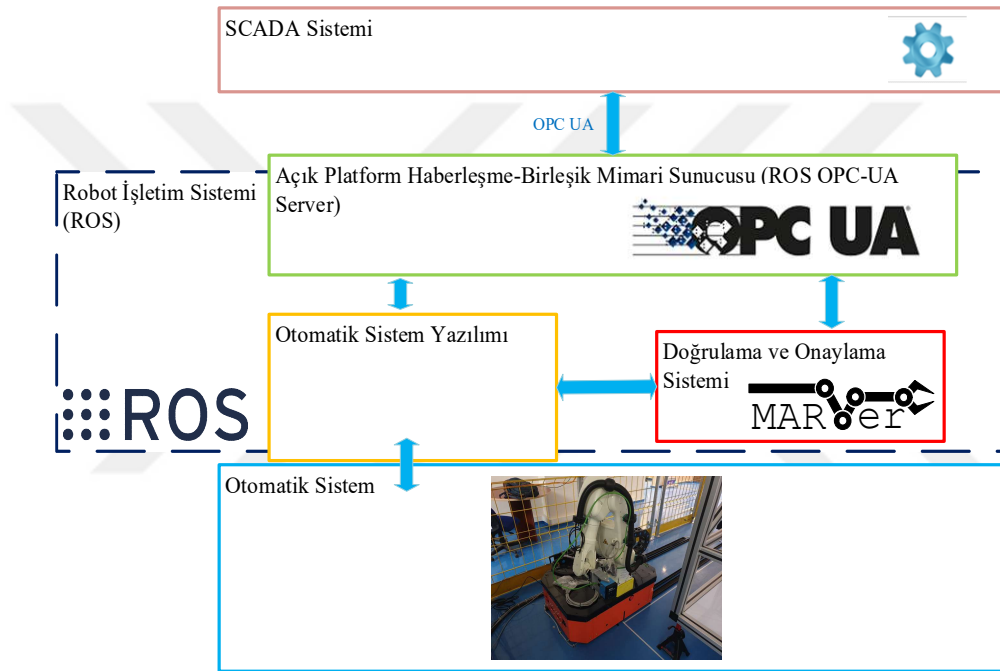
bunların son durumunu kapsamlı bir şekilde çalışmalarında sunmuşlar.

Literatürde bulunan SCADA sistemleri ve yazılım doğrulama ve onaylama çalışmaları incelenmiştir. Yapılan taramada, SCADA sistemleri ile yazılım doğrulama araçlarının birlikte kullanımına yönelik bir yaklaşım ile karşılaşılmamıştır.



3. SİSTEM TASARIMI

Önerilen yaklaşımda, otomatik sistemlerde uzaktan gözetleyici kontrolü ve veri toplama amaçlı yaygın olarak kullanılan SCADA sistemlerinin, yazılım doğrulama araçları ile kullanımı öngörülmektedir. Buna göre, sistem mimarisi Şekil 3.1’de gösterildiği gibi oluşturulmuştur.



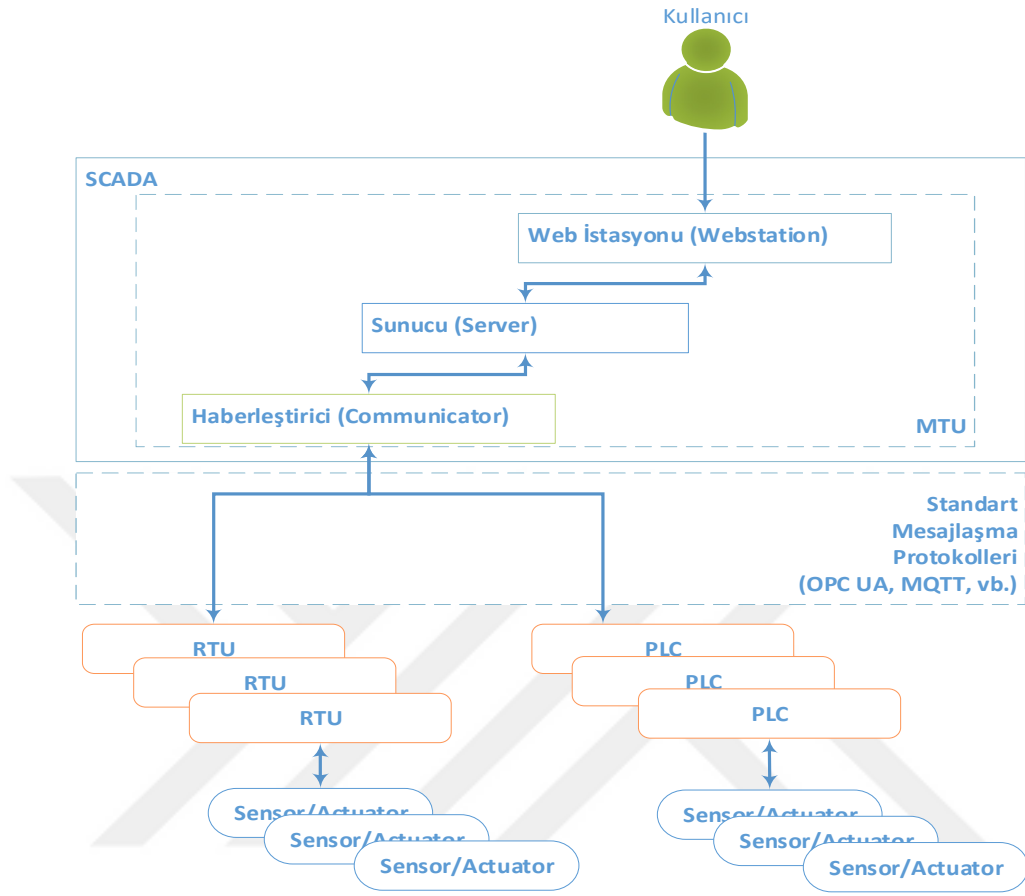
Şekil 3.1. Sistem Mimarisi

Otomatik sistemlerde, genellikle ara katman yazılımlar kullanılmaktadır. Robotik sistemlerde de bir ara katman yazılım olarak robot işletim sistemi (Robot Operating System, ROS) yaygın olarak kullanılmaktadır. Robotik sistemin denetim yazılımı ROS ara katmanı ile çalışmaktadır. ROS ortamında koşan diğer yazılımlar ile ROS mesajlaşma protokolleri kullanılarak veri alışverişi sağlanmaktadır. Doğrulama ve onaylama sistemi, geliştirme ve güncelleme süreçlerinde otomatik sistemin yazılımının doğrulaması ve onaylaması için yazılım araçlarını içermektedir. ROS ortamında koşan bu araçlar, otomatik sistem yazılımından verileri ROS mesajlaşma protokolleri ile almaktadır. Böylece, test altındaki sistem yazılımı ile

doğrulama aracının iletişimi, ROS ara katman yazılımı üzerinden sağlamaktadır. Otomatik sistemin uzaktan izlenmesi ve denetimi için SCADA sistemi kullanılmaktadır. SCADA sistemleri çeşitli endüstriyel iletişim protokolleri üzerinden otomatik sistem ile veri alışverişini sağlayabilmektedir. Bu protokollerden birisi de OPC UA olarak adlandırılan protokoldür. Otomatik sistemin veri alışverişinde kullandığı ROS protokolleri ile SCADA sisteminin kullanabildiği OPC UA protokolü arasında çevrimiçi dönüşüm sağlayacak ROS OPC UA sunucusu kullanılmaktadır. Böylece, ROS protokol üzerinden haberleşebilen doğrulama ve onaylama sistemindeki araçlar da SCADA sistemi ile veri alışverişini sağlayabilmektedir. Böylece, doğrulama ve onaylama sonuçlarının görselleştirilmesi, veri kaydı ve analizinde SCADA sistemi kullanılabilir duruma gelmektedir. SCADA ve robotik sistem arasında doğrulama ve onaylama sisteminin entegre edilmesiyle, sistem güncelleme ve iyileştirme çalışmalarının doğrulaması da yapılabilecektir.

3.1. SCADA Sistemi

SCADA sistemi, kapsamlı ve entegre bir veri tabanlı kontrol ve izleme sistemidir. SCADA yazılımları ile endüstriyel üretim süreçlerinden enerji üretimine, altyapı işlemlerinden (su arıtma, atık su arıtma, gaz boru hatları, rüzgâr santralleri, vb.) tesis tabanlı işlemlere (havaalanları, uzay istasyonları, gemiler, vb.) kadar çok çeşitli alanlarda izleme, kontrol, veri toplama, verilerin kaydı ve saklanması işlevleri gerçekleştirilmektedir. SCADA sistemlerinin, veri tabanlı olması ve bir işletmedeki verilerin toplanması nedeniyle, veriye dayalı birçok teknolojinin entegre edilmesi mümkün olabilir. Örneğin, kestirici bakıma yönelik eklentiler ile arızalar oluşmadan önlem alınması mümkün olabilir



Şekil 3.2. SCADA Sisteminin Genel Blok diyagramı

SCADA sisteminin genel mimarisi, Şekil 3.2’de gösterilmektedir. Gözetimli denetimi yapılacak olan fiziksel sistemle algılayıcılar ve sürücüler vasıtasıyla veri sağlanmakta ve denetlenmektedir. Bu enstrümanlar, uzak terminal birimi (Remote Terminal birimi, RTU) ve programlanabilir mantık denetleyici (Programmable Logic, Controller, PLC) gibi birimler ile bağlantılıdır. Bu birimler, farklı marka ve modelde temin edildikleri için SCADA sistemi ile veri aktarımında standart mesajlaşma protokollerine ihtiyaç duyulmaktadır. Bunun için, bu birimler tarafından da desteklenen protokoller mesajlaşma için kullanılmaktadır.

Kontrol merkezi birimi (Master Terminal Unit, MTU), SCADA sisteminin çekirdeğini oluşturur. MTU, kurulu olduğu bilgisayar üzerinde RTU ve PLC’ler ile haberleşerek veri alışverişini gerçekleştirir. SCADA sistemi de Haberleştirici, Sunucu ve Web istasyonu olarak 3 kısımdan oluşmaktadır. Haberleştirici, denetleyiciler ile etkileşimi gerçekleştirir ve verileri

sunucuya iletir. Sunucu, veri arşivlerini yönetir, matematiksel hesaplamalar yapar ve müşteri uygulamalara bilgi sağlar. Web istasyonu ise, web tarayıcıları üzerinden, çizelgeler, şemalar ve diyagramlar gibi farklı formlarda bilginin gösterimini ve sistemlere komutların iletilmesini sağlar. Bir anlamda, kullanıcıların sistemi görebildikleri ve müdahalede bulunabildikleri bir arayüz sağlar.

3.1.1. SCADA Sistemleri Temel Bileşenleri

SCADA sistemlerinde, gözetimli denetimi yapılacak fiziksel otomatik sistemlerden verilerin toplanması, işlenmesi ve kullanıcıya sunulması aşamalarında kullanılan bileşenler bulunur. Aşağıda bu bileşenler kısaca anlatılmaktadır.

3.1.1.1. Kontrol Merkezi Birimi (Master Terminal Unit, MTU)

Merkezi denetleyici veya merkezi terminal birimi olarak da isimlendirilebilen MTU, yerel bir ağ (LAN) ile veya geniş alan ağı (WAN) ile bir sunucu veya bir grup bilgisayarın ana sunucusuyla bağlanma formudur. MTU, RTU'lardan veri sağlamaktan sorumlu bir bilgisayar sistemidir. MTU görevleri SCADA bileşenlerinin haberleşmesini izlemek ve denetlemek, İnsan makine arayüz (HMI) yazılımı kullanarak SCADA haberleşmesi ile ilgili bilgi ve verileri grafiksel arayüz ile görüntülemektir. Ayrıca, saha cihazlarına komut göndermek ve almak da görevlerinin arasında yer almaktadır.

3.1.1.2. Uzak Terminal Birimi (Remote Terminal Unit, RTU)

Uzak Terminal Birimi veya Uzaktan İzleme ve Denetleme Ünitesi (RTU) bulunduğu merkezin sistem değişkenlerine ait bilgileri toplayan, depolayan ve gerektiğinde de bu bilgileri kontrol merkezine bir iletişim ortamı ile gönderen kontrol merkezinden gelen komutları uygulayan bir terminaldir. Uzak Terminal Birimi, ölçüm, kontrol, durum izleme ve diğerleri sonuçlarından terminaller (bir tür veri istasyonu) olarak işlev gören bir SCADA alt sistemidir. RTU görevleri elektrik akımı, statik basınç, sıcaklık ve diğerlerini ölçmek gibi dönüştürücülerden gelen sinyalleri çevirme, dönüştürme ve hesaplama işlevi görür. Daha sonra

ise verileri MTU'ya iletir veya RTU'daki sistem kontrol için deęilse doęrudan MTU'ya iletir.

Böylece RTU'lar, aian cihazlarını izleyen ve denetleyen, tesis denetimine ya da SCADA sistemine bağlayan mikrodnetleyici tabanlı bir cihazdır. Bir RTU, merkezi işlemci birini, güç kaynağı, haberleşme portları, analog ve dijital fiziksel giriş çıkış birimleri içerir. SCADA sistemlerinde, farklı tipte alan cihazlar bulunur ve bu cihazlar bağlı oldukları RTU aracılığıyla veri iletimi gerçekleştirir. Farklı marka ve model RTU'lar bulunmaktadır. Farklı üreticiler tarafında sağlanan RTU'ların SCADA sistemi ile haberleşmesinde OPC UA, MQTT, Modbus gibi standart protokoller kullanılmaktadır.

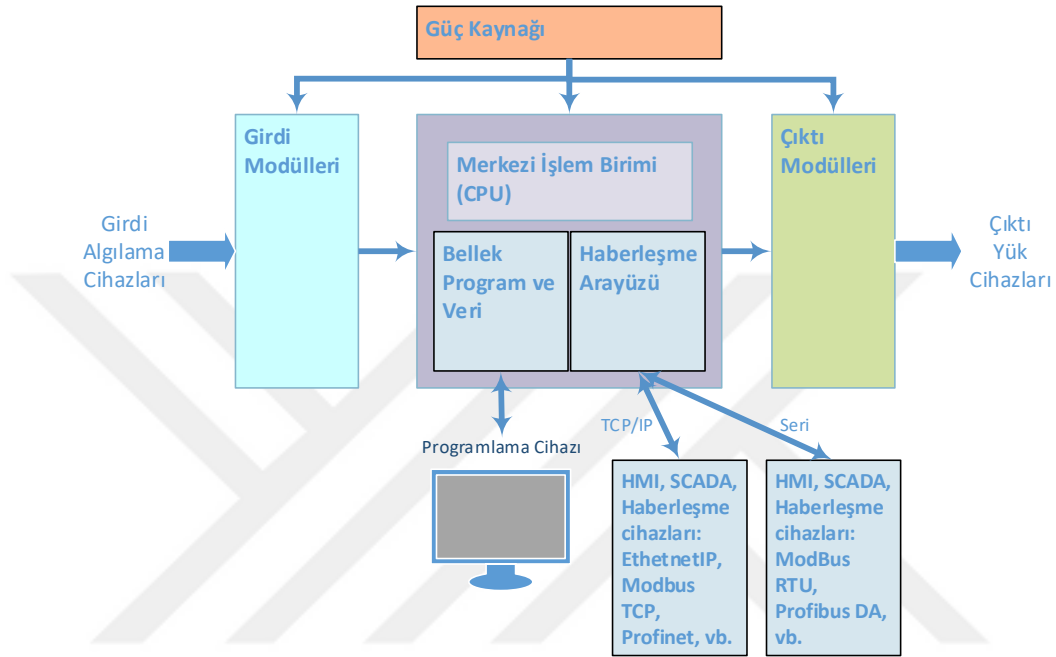
3.1.1.3. Programlanabilir Mantık Denetleyici (Programmable Logic Controller, PLC)

PLC, endüstriyel kontrol sistemleri için kullanılan özel bir bilgisayar cihazıdır. Petrol rafinerileri, üretim hatları, makineler, robotik sistemler ve benzeri. Birçok endüstride yaygın olarak kullanılır. Bu cihazlar, belirli bir işlemi örneğin, makine fonksiyonunu ve tüm üretim hattın bile otomatikleştirmesini sağlar. PLC, aynı zamanda SCADA sistemine de veri sağlayabilir. SCADA sunucuları, PLC den veri alıp ve kontrol sistemini komuta edebilir.

İlk PLC, 1968 yılında icat edilmiş ve o yıldan sonra üretim süreçlerinin otomatikleştirilmesinde önemli rol oynamıştır. Günümüzde de tüm üretim sistemlerinde kullanılmaktadır. PLC'ler az sayıda entegre girdi ve çıktı birimleri ile küçük modüler cihazlardan, kızak sistemi ile eklenerek genişletilmiş binlerce girdi ve çıktı içeren ve başka PLC'ler ağ oluşturan çok büyük cihazlara kadar kullanımı bulunmaktadır. Ayrıca, günümüzde PLC'lerin çoğu modüler olup kullanıcının ayırık kontrol, analog kontrol, PID kontrol, pozisyon kontrolü, motor kontrolü, seri haberleşme ve yüksek hızlı ağ gibi çeşitli fonksiyonlar ekleyebilmesine izin vermektedirler. Binlerce işlemlerin yerini alacak tek bir PLC programlanabilir.

Şekil 3.3'te görüldüğü gibi, PLC merkezi işlem birimi ve entegre program ve veri depolayan bellek ve haberleşme arayüzü içermektedir. PLC programları genellikle kişisel bir bilgisayarda özel bir uygulamadan yazılır ve ardından doğrudan bağlantı kablosu ile veya bir ağ

üzerinden PLC'ye aktarılır. Ayrıca, diğer çevresel cihazlarla TCP/IP ve seri haberleşme gibi farklı haberleşme tipleri ile veri alışverişi yapabilmektedir. Girdi ve çıktı birimleri, denetimini gerçekleştireceği tesisteki algılama ve yük cihazlarına veri erişimini sağlamaktadır.



Şekil 3.3. PLC Genel Mimarisi

3.1.1.4. İnsan Makine Arayüzü (Human-Machine Interface, HMI)

Operatör panelleri ve dokunmatik panel gibi adlarla da nitelendirilen HMI sistemler insan ve makine etkileşimini sağlamaktadır. Panel üzerinde işletim sistemi çalıştıran bir çeşit bilgisayardır. İşletim sistemleri kullanıcıya kapalıdır ve özel editörler yardımı ile programlanırlar. Farklı işletim sistemleriyle uyumlu kullanılabilirler. Temel görevleri ise operatörün verdiği komutları otomasyon sistemine iletmek olan, otomasyon sisteminden aldığı proses verilerini de ekranında görüntüleyen cihazlardır.

HMI sistemlerinin birçok kullanım alanı bulunmaktadır. Neredeyse tüm endüstriyel şirketlerde makineler ile etkileşime geçmek ve endüstriyel süreçlerini geliştirmek için kullanılmaktadır. HMI ekranların başlıca kullanım alanlarına (enerji, imalat, geri dönüşüm,

petrol ve gaz, taşımacılık vb.) örnekleri verebiliriz.

SCADA Sistemleri, gerektiğinde uzaktan ve yakından kontrol edilebilen, otomasyon sistemlerindeki insan kaynaklı hataların azaltılmasını sağlayan ve mevcut problemlerin çözülebilmesini sağlayan yazılım sistemleridir.

3.1.2. Açık Kaynaklı SCADA Yazılımları

Çoğu endüstri, üretim süreçlerinde kullanmak üzere SCADA sistemine ihtiyaç duyar. Ancak, ticari SCADA yazılımları yüksek lisans ücretleri ile işletmeler için önemli bir maliyet kalemi oluşturur. Bu maliyet kalemi, küçük ve orta ölçekli endüstrilerin SCADA sistemlerine erişimi önünde bir engel teşkil eder. Bundan dolayı, bu endüstriler üretim performansının artmasına katkı sağlayacak bu sistemin kurulumunu gerçekleştiremez. Ancak, açık kaynak kodlu yazılımlar bulmak mümkündür. Endüstriler bu yazılımları, kendi sistemlerinde kullanabilir ve kendi sistemlerine göre özelleştirebilir.

Yapılan araştırma sonucu, bazı açık kaynak kodlu SCADA yazılımları incelenmiş ve aşağıda verilmektedir.

OpenSCADA

OpenSCADA, açık kaynaklı SCADA ve HMI gerçekleştirmesidir (OpenSCADA, 2013). Açık kaynaklı, modüler, ölçeklenebilir ve çok platformlu olması temel özellikleridir. Sistemin son sürümü 0.9 LTS (17.06.2018) çıkmıştır.

Free SCADA

FreeSCADA (2023), ücretsiz SCADA yazılımıdır. PLC ya da Modbus TCP/IP cihazları ile haberleşebilmektedir. Yazılım, web uygulama olarak çalışmakta, IP adresi girildikten sonra cihazların sisteme eklenmesi, görüntülemek ve PLC üzerinden kontrol etmek üzere ekranlar

oluşturulmasına imkan sağlamaktadır.

IndigoSCADA

IndigoSCADA (2023), açık kaynaklıdır. Windows işletim sisteminde çalışabilmektedir. Birden fazla HMI penceresini destekler. Desteklediği protokoller, OPC DA 2.05, A&E 1.1, HDA 1.20, DNP 3.0, RFC 1006 ve Modbus yer almaktadır.

S.E.E.R. 2

S.E.E.R. 2 (2023), web tabanlı kullanıcı arayüzüdür. Bu uygulama farklı dillerde mevcuttur. Sistemin son sürümü 30.11.2016 çıkmıştır.

ScadaBR

ScadaBR (2023), bir SCADA sisteminin tüm fonksiyonlarını içeren açık kaynak yazılımdır. Günlük kayıtları, raporlama ve kontrolleri sunan SCADA sistemidir. Grafik görünümünü, resimler ve alarmlar tanımlayabilir. Desteklediği protokoller, Modbus RTU, Modbus TCP, BACnet, OneWire, Http, Zwave yer almaktadırlar. Uyumlu olduğu işletim sistemleri, - Windows XP Pro Service Pack 3; Windows 7 tüm sürümler (32/64 bit); Windows 8.1 tüm sürümler (32/64 bit) işletim sistemleridir.

SZARP

Szarp (2023), Linux için tüm özelliklere sahip açık kaynaklı SCADA sistemidir. Burada istemci/sunucu mimarisi ve HTTP kullanarak birden fazla verilere şeffaf erişim sağlanır.

RapidSCADA

RapidSCADA (2023), açık kaynaklıdır ve ücretsizdir. Ancak, farklı geliştiriciler tarafından sunulan eklentiler ücretli olarak alınabilmektedir. Bu eklentiler olmadan, yapılabilecekler

oldukça kısıtlı kalmaktadır. RapidSCADA, sunucularda, gömülü bilgisayarlarda ve bulutta çalışır. Hızlı SCADA düğümleri, kendi aralarında bilgi alışverişinde bulunur ve gerçek zamanlı olarak harici veri tabanlarıyla etkileşime girer. Sistemin son sürümü (15 Mayıs 2023) çıkmıştır.

Fernhill SCADA

Fernhill (2023), istemci/sunucu mimarisine dayalı ölçeklenebilir SCADA sistemidir. Birden fazla istemciyi destekler. Kullanımı ve kurulumu kolaydır. Açık veri erişim arayüzlerini destekler. Tek bir IDE kullanılarak yapılandırılmıştır. Sistemin yeni sürümü 3.93, 2 Nisan 2023 tarihinde çıkmıştır.

3.2. Robot İşletim Sistemi (Robot Operating System, ROS)

ROS (2023), robotik uygulamaları geliştirmek üzere faydalanan yazılım kütüphaneleri ve araçlarından meydana gelir. Sürücülerden güncel algoritmalara kadar güçlü geliştirme araçları ile bir robotik projede ortaya çıkabilecek tüm ihtiyaçları karşılar. ROS, bileşenler, arayüzler ve araçları tanımlayan ücretsiz ve açık kaynak kodlu bir yazılımdır. Robotlar, hareket için gereken eyleyiciler, içinde bulunduğu dünyayı okuyabilmesi için sensörler ve otonom çalışması için kontrol sistemi ya da robot beyni içerir. ROS, bu bileşenlerin kolaylıkla kurulması ve ROS araçları kullanımı ile bunların birbirine bağlantılanmasını sağlar. Bu bağlantıyı, konu (topic) ve mesaj olarak adlandırılan yapı ile gerçekleştirir. Tüm bileşenler, tanımlanan temel bir fonksiyonu gerçekleştirmek üzere ayrı birer program olarak çalışmakta ve mesajlaşma altyapısı ile tüm robot sisteminin işlerliğini sağlayacak şekilde birlikte çalışabilmektedir. Kaydol/yayınla (publish/subscribe) protokolü ile mesajlaşma gerçekleştirdiği için, robotik sistemin amaca uygun olarak işleyişini sağlayan programlar yanında farklı araçlarda mesajlara erişebilmektedir. Bu araçlar ile robot uygulamasının 3B simülasyon ortamında test edilmesi, robotik sistemin dijital ikizinin oluşturulması, sistem verilerinin görselleştirilmesi gibi fonksiyonlar gerçekleştirilmektedir. Böylece, test yapma ve eğitim gerçekleştirmek çok daha etkin olmaktadır. ROS, kamera, motor sürücüsü vb. birçok robotik sistemlerde kullanılan bileşen için donanım arayüzlerini desteklemektedir. Böylece, geliştiriciler donanım sağlayıcılarından

bağımsız olarak farklı olarak bileşenlere aynı arayüz üzerinden erişim sağlayabilmektedir. Programlama dili bağımsız olarak nitelendirilen bu sistem ile aynı robotla farklı programlama dillerinde iletişim kurulabilir. ROS sayesinde robotun sensörler aracılığıyla aldığı veriler işlenir ve robota yeniden komut olarak gönderebilir. Robotların kontrolü için kullanıldığını bildiğimiz sistem mevcut kodlar üzerinde değişiklikler yaparak çalışan bir yazılım geliştirmeye olanak tanır. Daha önceden geliştirilmiş olan algoritmaların robot üzerinde hızlı bir şekilde kullanılmasını sağlar ve kıyaslama, gözlem, geliştirme durumları için avantaj sağlar.

ROS, üç kavramsal seviye içerir: Dosya sistem seviyesi, hesaplama grafiği seviyesi ve topluluk seviyesi. Bu seviyeler altında çeşitli kavramlar geliştirilmiştir.

ROS dosya sistem seviyesinde, disk üzerinde bulundurulmuş ROS kaynakları kapsar.

- Paket: ROS yazılımlarının organizasyonu için kullanılır. Bir paket, ROS çalışma anı süreçleri (düğüm, node), ROS bağımlı kütüphaneleri, veri setlerini, yapılandırma dosyalarını ve organize edilecek tüm kaynakları içerir.
- Depolar: Ortak sürüm takip sistemine sahip paketlerin bir koleksiyonudur. Paketler aynı sürüm sistemini paylaşır ve birlikte yükseltilirler.
- Mesaj tipleri: Düğümler arasında iletilen mesajların yapısı, paket içindeki bir dizinde dosya olarak tutulur. Gönderilen ve alınan mesajın içerik çözümlemesinde bu yapılar kullanılır.
- Servis tipleri: Düğümler, servis olarak belirtilen mesaj yapıları ile servis talebinde bulunabilir ya da servis talebini alabilir. Servis mesajlarının yapısı paket içindeki bir dizinde dosya olarak tutulur.

ROS hesaplama grafiği seviyesinde, hesaplama grafiği birlikte veri işleme gerçekleştiren ROS proseslerinin eşler arası ağıdır. Bu ağın paydaşları bir dizi kavramı beraberinde getirir.

- Düğümler (Nodes): Düğümler, hesaplamaları yapan bilgisayar prosesleridir. ROS, modüler bir yapıya sahiptir. Bu modülerlik için önemli bir katkı bu düğüm yapısı ile sağlanır. Bir robotik uygulamanın gerçekleşmesinde, basit fonksiyonel işlemleri yürüten çok sayıdaki düğümün birlikte çalışması ile sağlanır. Örneğin, bir düğüm lazer mesafe sensörünü

kontrol eder, bir düğüm konumlandırma hesabını yapar, bir düğüm yol planlaması yapar ve tüm bunlar birbiri ile etkileşim içinde çalışır. Bir düğüm programlanırken, diğer düğümlerle haberleşme için ROS kütüphanesi kullanılır.

- Master: ROS master, isim sunucusu gibi çalışır. Birden çok düğümün birbirine erişiminde aracılık eder.
- Mesajlar: Düğümler, birbirine veri aktarımında mesajları kullanır.
- Başlıklar (Topics): Mesajların yönlendirilmesinde, kaydol/yayınla (publish/subscribe) yaklaşımı kullanılır. Bir düğüm, diğer düğümlere sağlayabileceği veriler için, belli bir mesaj formatında tanımlı bir başlık oluşturur. Bu başlığı, Master üzerinden kaydettirir. Diğer düğümler, bu başlık için kaydolduklarında, her bu başlık altında mesaj iletildiğinde otomatik olarak alırlar.
- Servisler: Başlık üzerinden veri aktarımı birden çoğa (one-to-many) mantığı ile gerçekleşir. Servislerde, bir düğüm gerçekleştireceği servisi master üzerinde kaydettirir. Diğer düğümler, bu servis talebinde bulunduğu düğüm servisi gerçekleştirir. Böylece, iste/cevapla (request/reply) yaklaşımı ile işletilir.
- Çantalar (Bags): Mesaj verileri, çantalar vasıtasıyla kaydedilip, tekrar oynatılabilmektedir. Böylece, geliştirme ve test yapma süreçlerinde çok yararlı olmaktadır.

ROS topluluk seviyesinde, farklı toplulukların yazılım ve bilgi paylaşılabildiği kavramları içerir.

- Dağıtım: Dağıtımlar, kurulum gerçekleştirilebilecek sürüm koleksiyonunu belirtir. Linux işletim sistemlerinin dağıtımlarına benzer bir rol gerçekleştirir. Böylece, bir yazılım koleksiyonun kurulumunu kolaylaştırır ve aynı zamanda bir yazılım grubu içinden tutarlı sürümleri sağlamayı kolaylaştırır.
- Depolar: Farklı kuruluşlar, kendi robot yazılım bileşenlerini geliştirir ve piyasaya sürer.
- ROS Wiki: ROS ile ilgili bilgilerin belgelendiği ana forumdur.

Mesajlar çeşitli görselleştirme ve uzaktan kontrol araçlarına kolayca gönderilebilir. ROS ile gerçek robot yerine "dijital ikiz" adı verilen simüle edilmiş bir robotu çalıştırabilirsiniz. Bu,

test ve eğitimi daha kolay ve daha etkili hale getirir. ROS, kameralar ve motor denetleyicileri gibi birçok yaygın robot bileşeni için donanım arabirimlerini destekler.

Çalışmamızda ROS'u kullanmamızdaki etmenleri belirtmek gerekirse, bunlar kullanımın kolay olması ve sistemin kolayca anlaşılabilir olmasıdır. Diğer etmen olarak simülasyon ortamı sağlaması ve bunun getirdiği ayrıcalıklardan yararlanabilmektir. Burada ayrıca debug-hata ayıklama mekanizmasından da faydalanabiliyoruz. Testler yapıldığı zaman veriler kaydedebiliyoruz ve bu bizlere geliştirme aşamasında kolaylık sağlıyor.

ROS'un modüler mimarisi, geliştiricilerin robotları satıcı, kilitlenme veya lisans ücretlerinden bağımsız olarak oluşturmaya olanak tanır. ROS ayrıca yazılım arayüzüne sahip olan hemen hemen her bileşenle çalışabilir. ROS'un robot geliştirmeye yönelik ortak araçlar, tekerleği yeniden icat etmeye değil, sorununuzu çözmeye odaklanmanızı sağlar. Bu nedenle küçük startup'lardan Fortune 500 kuruluşlarına, akademik araştırma laboratuvarlarından devlet kurumlarına kadar yüzlerce şirket robot geliştirme platformu olarak ROS'u seçmiş ve yalnızca 2020 yılında yarım milyardan fazla ROS paketi indirmiştir.

Ancak, ROS bir yazılımdan çok daha fazlasıdır, ROS aynı zamanda ileri robot teknolojilerini akla gelebilecek her türlü uygulamaya uygulayan ve çözümlerini birbirleriyle paylaşan yüksek eğitilmiş ve pragmatik robotiklerden oluşan bir topluluktur. ROS'u seçtiğinizde, dünya çapında bir yetenek havuzundan faydalanmış olursunuz. ROS sürekli büyüyen bir hizmet topluluğudur. Bu hizmetler arasında günlük kayıt ve teşhis, derin öğrenme veri toplama, test ve kalite güvencesi ve daha fazlası yer almaktadır.

ROS, robotlarda yaygın olarak kullanılan bir yazılımdır. Kodda küçük değişiklikler yaparak diğer robotlar üzerinde çalışabilen yazılımlar geliştirmeyi hedefler. ROS sayesinde, standart robotlar için yazılan standart algoritmalar yükle ve çalıştır hızında çalışır. Robota bağlı sensörlerden alınan veriler ROS aracılığıyla bilgisayara gelir ve bilgisayar tarafından işlen veriler robota görev olarak geri döner.

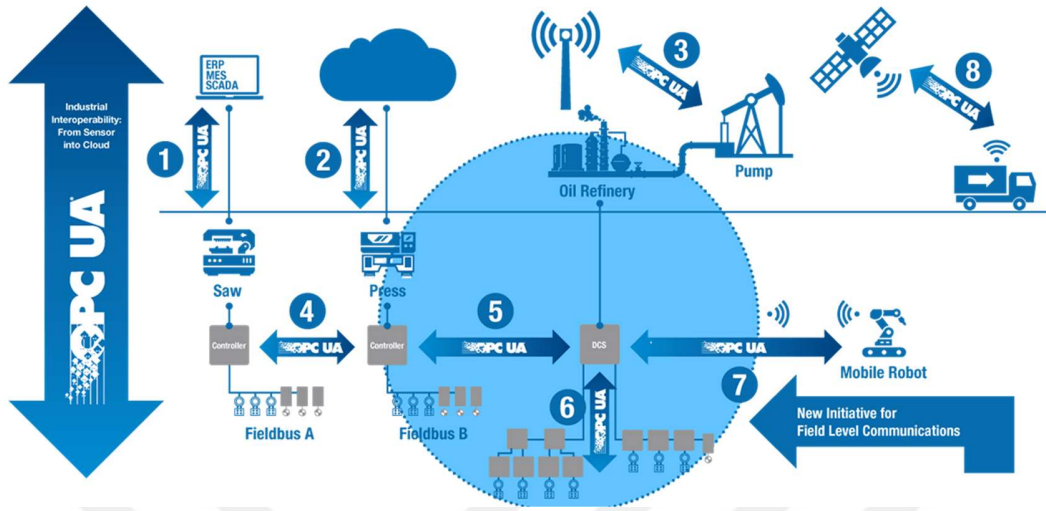
3.3. Açık Platform Haberleşmesi – Birleşik Mimarisi (OPC-UA)

OPC UA, genellikle endüstriyel alanlarda tüm cihazlar arasında güvenli veri alışverişinde kullanılan açık bir standarttır (OPC,2023). Açık Platform İletişimleri (OPC) tarafından 2008 yılında piyasaya sürülmüştür. Platformdan bağımsızdır. Böylece, OPC UA Windows, Mac, Android ve Linux ile uyumludur. Farklı cihaz üreticileri, cihazlarının bu standarda uygun olarak haberleşmesi için yazılım sağladığında, farklı tip cihazların birbirleri ile konuşabilmesi mümkün olabilmektedir. Böylece, PLC, SCADA ve ERP gibi donanım ve yazılımlar tüm birbirleri ve sahadaki tüm cihazlarla tek dilde güvenli ve kayıpsız bir şekilde haberleşebilir.

Bu standart, önemli avantajlar sağlaması nedeniyle yaygın olarak kullanılmaktadır. Düşük seviyeli cihazlardan yüksek seviyeli uygulamalara bilgi akışının sağlanır ve OPC UA kullanılarak geçmiş veri kaynaklarından ham veriler elde edilebilir. OPC UA, platformdan bağımsızdır; endüstriyel sistemler, herhangi bir işletim sistemini kullanarak herhangi birinin yazılımını sıkıntısız bir şekilde entegre edebilir. Bir diğer avantajı da birlikte çalışabilirliğidir, bu ise özel endüstriyel sistemler oluşturmaya olanak tanır. SCADA sistemleri geliştirmesini sağlar, böylece mevcut tesis ekipmanı ek yapılandırma olmadan yeni yazılım modülleriyle entegre olabilir.

OPC UA'nın dezavantajlarına ise karmaşık yapısı ve veri yapılarını işlerken esnek olmaması başta gelir. Diğer dezavantajı, cihaza özgü sınırlamalardır. Bu sınırlamalar, elektronik imza ve geçmiş veri kaynakları gibi belirli özellikler için destek eksikliğini içerir.

OPC UA hizmet odaklı bir mimaridir (SOA). OPC UA-da OPC UA sunucuları ve OPC UA istemcileri vardır. Bir OPC UA sunucusu, bir OPC UA istemcisine SCADA uygulamalarında verilerin işlenmesi, saklanması ve toplanması için OPC UA modellerini kullanarak endüstriyel otomasyon verilerine güvenli erişim sağlar.



Şekil 3.4. Endüstriyel IoT Uygulamalarında Standartlaştırılmış Haberleşme Protokolü olarak OPC UA Kullanımı (Kaynak: <https://page.advantech.com/en/global/industrial-automation/industrial-io/opc-ua>)

Şekil 3.4’te görüldüğü gibi, endüstriyel uygulamalarda veri akışının olduğu tüm birimler arasında OPC UA kullanılmaktadır. Bunlar, şekilde görünen numaralarla ilişkili olarak,

1. IT/OT Haberleşmesi,
2. Bulut Entegrasyonu,
3. Güvenli Uzak Erişim,
4. Yerel OT Haberleşmesi,
5. Kontrolörden Kontrolöre Haberleşme,
6. Kontrolörden Alan Cihazlarına Haberleşme,
7. Kablosuz Entegrasyon (5G),
8. Gelecek Hazırlıkları

verilebilir. Üretimde Endüstri 4.0 temel zorluklarından biri, düşük seviye cihazlardan gerçek zamanlı veri toplamaktır. OPC UA, endüstrilerin OPC UA sunucularını tüm cihazlarda dağıtmasına olanak tanır. Bu, büyük miktarda gerçek zamanlı verinin analiz, sıralama ve çeşitli tüketici uygulamalarıyla paylaşılması için kurumsal düzeydeki kontrol sistemlerine ve uygulamalarına yönlendirilebileceği anlamına gelir.

3.4. OPC UA ile ROS Arası Veri Aktarım Köprüsü (OPC UA- ROS Bridge)

ROS-OPC UA (2023), ROS ve OPC UA iletişim standartları arasında iletişim arayüzü sağlar. Bir ROS paketi olarak temin edilebilmektedir. ROS ortamında başlık ve servis üzerinden yapılan mesajların OPC UA protokolü üzerinden iletimini sağlar. Açık kaynak kodlu OPC UA protokolünün gerçekleştirmesini sağlayan yazılım kütüphanelerinden faydalanır. Bunlar, C++ programlama dilinde freeopcua (freeopcua, 2023) ve Python dilinde python-opcua (python-opcua, 2023) olarak adlandırılan kütüphanelerdir.

3.5. Robotik Sistemler için Yazılım Doğrulama Aracı (Model-Aided Runtime Verification for Robotic Systems, MARVer)

Siber fiziksel sistemlerin (Cyber-physical systems, CPS) hata ayıklama süreçleri oldukça zorlayıcıdır. Aynı zamanda, ayıklanamayan hatalar bu sistemlerin kullanımında önemli emniyet sorunlarına neden olabilir. Bundan dolayı, bu sistemlerde kullanılabilecek kapsayıcı, güvenilir ve uygulaması az zaman ve çaba gerektiren doğrulama ve onaylama yöntemleri ve yazılım araçlarına ihtiyaç duyulur.

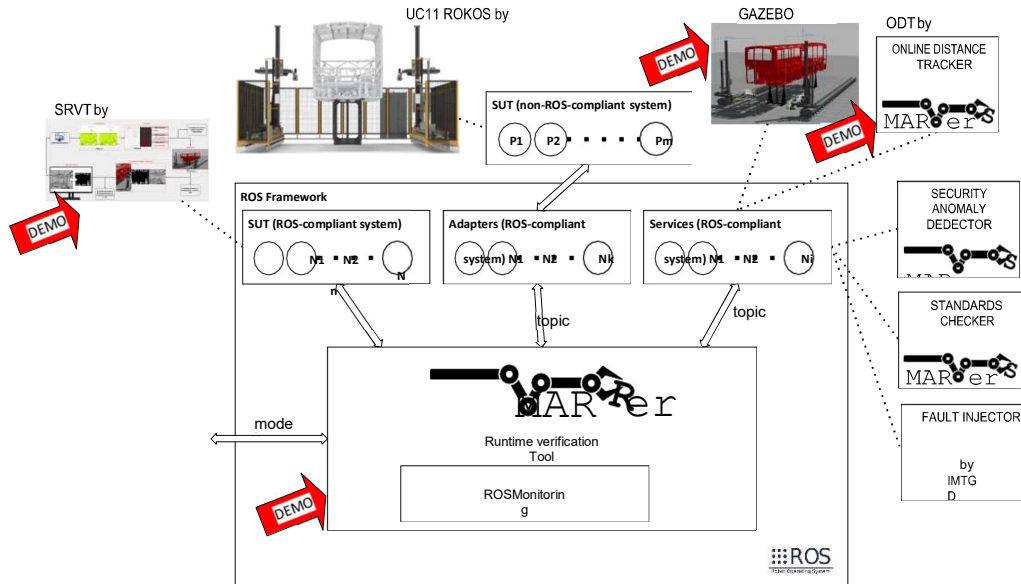
Doğrulama, geliştirilen bir sistem için belirlenmiş olan isterlerin sağlandığının gösterilmesidir. Diğer bir ifadeyle, bir geliştirme fazında bir ürünün başlangıçta belirlenen koşulları sağlandığını belirlemek üzere gerçekleştirilen değerlendirme süreci olarak açıklanabilir. Bu tez çalışması kapsamında, Model Destekli Çalışma Zamanı Doğrulama Aracı (Model-Aided Runtime Verification for Robotic Systems, MARVer) olarak adlandırılan ve robotik sistemlerin doğrulaması için geliştirilen bir araç kullanılmaktadır (Demirci vd., 2022; Demirci, 2023).

MARVer, açık kaynaklı, ROS ortamında çalışan robotik sistemlerin doğrulaması için geliştirilen model destekli Çalışma Zamanı Doğrulama Aracıdır. MARVer, iki doğrulama yazılımından faydalanılarak geliştirilmiştir. Bunlar, UPPAAL (Behrmann vd., 2004) ve ROSMonitoring (Ferrando vd., 2020) olarak adlandırılan araçlardır. UPPAAL, model doğrulama metodunu gerçekleştiren entegre bir araçtır. ROSMonitoring ise, ROS sistem içinde

veri alışverişi yapılan mesajları dinleyerek çalışma anında doğrulama gerçekleştirmek için kullanılan bir yazılım aracıdır.

MARVer, C++ ve Python programlama dilleri kullanılarak geliştirilmiş ve Linux işletim sisteminde koşturmaktadır. Doğrulama sürecinin ayarlanmasında ve gerçekleştirilmesinde kullanılan bir grafik kullanım arayüzüne sahiptir. Şekil 3.5’da gösterildiği gibi temelde üç bileşenden oluşmaktadır.

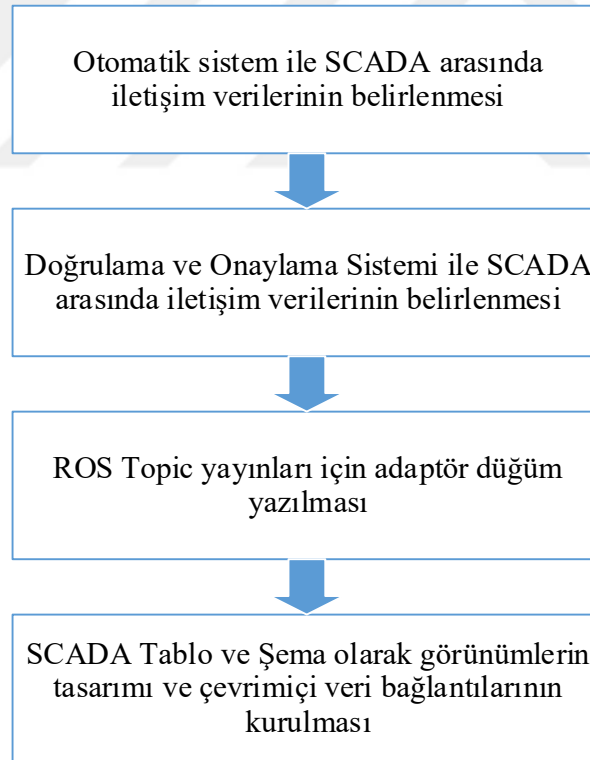
- MARVer-M: UPPAL ile oluşturulan modellerin görselleştirilmesi, veri yapısı ilişkilerinin tutulması ve modeldeki tanımlamaların çalışma anı doğrulamasında kullanılmak üzere dönüştürülmesini gerçekleştirir.
- MARVer-R: Tüm ayarlamalar, doğrulama için kullanılacak isterlerin tanımlaması ve çalışma anı doğrulama sürecinin yönetimi bu bileşen tarafında gerçekleştirilir.
- MARVer-S: Bir robotik sistemin doğrulaması için gereken, hesaplamalar, analizler, veri tedariki gerçekleştirerek doğrulama sürecine hizmet sunan düğümler bu bileşen altında bulunur.



Şekil 3.5. MARVer Mimarisi (Demirci vd., 2022)

4. MATERYAL VE YÖNTEM

Sistem kurulumunda izlenecek adımlar, Şekil 4.1’de verilmektedir. Buna göre öncelikle, otomatik sistemde SCADA sistemi üzerinden uzaktan izlenmesi ve denetimi yapılacak verileri belirlenmesi ve analizini yapılması gerekmektedir. Ayrıca, doğrulama ve onaylaması amacıyla kullanılacak yazılım aracının SCADA sisteminde görselleştirileceği verilerinin de belirlenmesi ve analiz edilmesi gerekmektedir. Otomatik sistem ve doğrulama sisteminin, ROS topic olarak yayınladığı verilerin SCADA sisteminin kabul edeceği protokollerde olması gerekmektedir. Bu çalışmada, OPC UA kullanımı tercih edilmiştir. ROS topiclerin, OPC UA protokolü ile gönderilebilmesi için ve yapısında değişiklik yapılması ihtiyacı ortaya çıkmaktadır. Bundan dolayı, ROS topic yayınlarının dönüşümü için adaptör düğüm yazılması gerekmektedir.



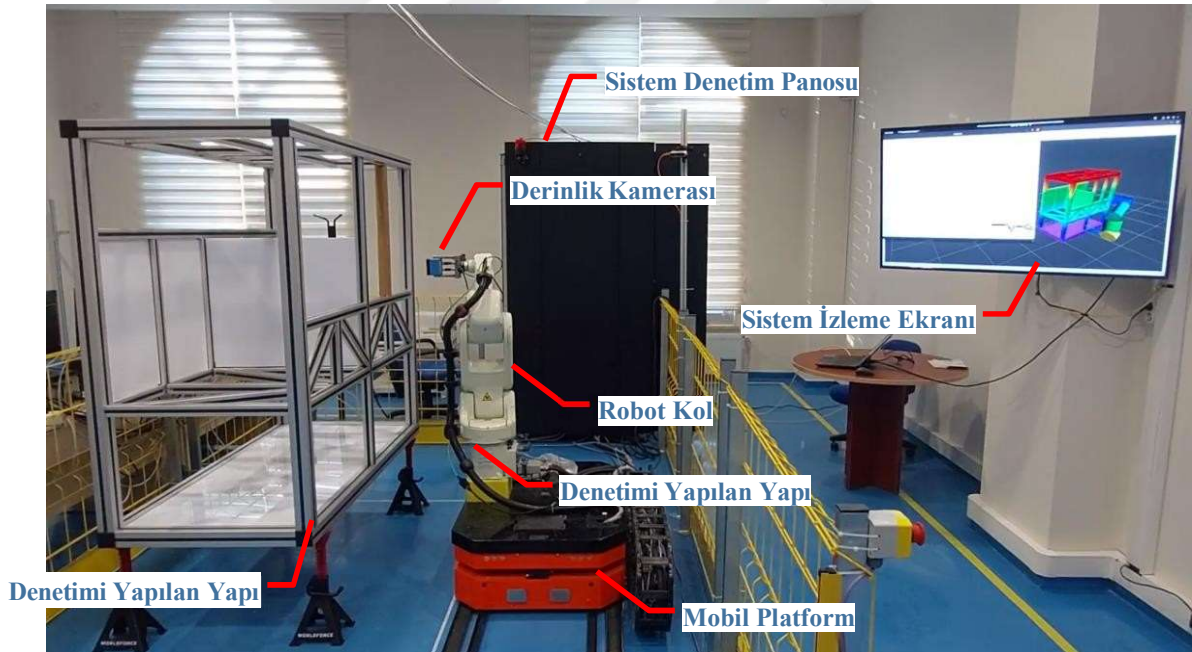
Şekil 4.1. Sistem kurulumunda izlenen adımlar

SCADA sisteminde, otomatik sistem ve doğrulama sisteminden aktarılan verilerin

uygun bir formda görselleştirilmesi için tablo ve şemaların tasarımı yapılmaktadır. Çevrimiçin verileriyle bağlantıların kurulması ve testleri ile sistem kurulumu gerçekleştirir.

4.1. Robotik Denetim Sistemi

SCADA sistemi ile entegre edilecek olan otomatik sistem olarak, robotik denetim sistemi kullanılmaktadır. Şekil 8’de görülen bu sistem, VALU3S projesi (<https://valu3s.eu/industrial-robotics-automation-use-cases/> , 18 Mayıs 2023) kapsamında, kullanım durumlardan birisi olan Otobüslerin iskelet sistemlerinin kalite kontrolünde kullanılan robotik denetim sisteminin bir TRL5 seviyesi laboratuvar prototipidir. Bu prototip, Eskişehir Osmangazi Üniversitesi Akıllı Fabrika ve Robotik Laboratuvarında (İFARLAB) (<https://ifarlab.ogu.edu.tr/> , 18 Mayıs 2023) kurulmuştur.



Şekil 4.2. Robotik Denetim Sistemi

Bu sistemde, 6 eksenli Kawasaki RS005L model bir robot kolu, İnovasyon Müh. üretimi olan ve sadece tek eksenle hareket edecek şekilde modifiye edilen mobil robot platformu üzerinde hareket ettirilmektedir. Robot kolun ucunda, denetim görevinde kullanılan Sick Visionary T-Mini derinlik kamerası bulunmaktadır. Robotik sistemin denetimini gerçekleştiren

denetim bilgisayarlarının bulunduğu pano bulunmaktadır. Büyük bir ekran, sisten verilerinin görselleştirilmesinde kullanılmaktadır. Büyük bir otobüs şasisinin bir bölümünün, ½ ölçekte küçültülen bölümü, çalışma uzayı içinde denetimi yapılacak yapı olarak bulunmaktadır.

Sistemde uzaktan izlenmesine ihtiyaç duyulacak veriler Çizelge 4.1’de verilmektedir. Sistem ROS altyapısı üzerinde çalıştığı için, tabloda kullanılacak verinin adı, verinin tipi, hangi ROS topic üzerinden yayınlandığı, yayın yapılan ROS topik tipi ve verinin kısa tanımı bulunmaktadır. Bu verilerin, SCADA sistemlerinin uyumlu olduğu veri transfer protokollerinden birine dönüştürülerek iletilmesi gerekmektedir.

Veriler arasında, robot kol ve robot platformun konum bilgileri bulunmaktadır. Ayrıca, mobile platformun hızı ve durumunu belirten veriler de mevcuttur. Sistemin denetleme görevinin durumunu bildiren veri de sistemden sağlanmaktadır.

Çizelge 4.1. Robotik Denetim Sistemi SCADA sisteminde kullanılacak verileri

Veri Adı	Veri Tipi	ROS Topic	ROS Topic Tipi	Tanımı
posJ1 – posJ6	double	/right_rokos/joint_states	sensor_msgs/JointState	6 eksen robot eklem konumları
mobPos	double	/odom (/odom/pose/pose/position/x)	nav_msgs/Odometry	Mobile platform konumu
mobVel	double	/odom (/odom/twist/twist/linear/x)	nav_msgs/Odometry	Mobile platform hızı
mobStatus	string	/mobile_status	std_msgs/string	Mobile platform durumu
taskStatus	string	/task_status	std_msgs/string	Sistem denetleme görevi durumu

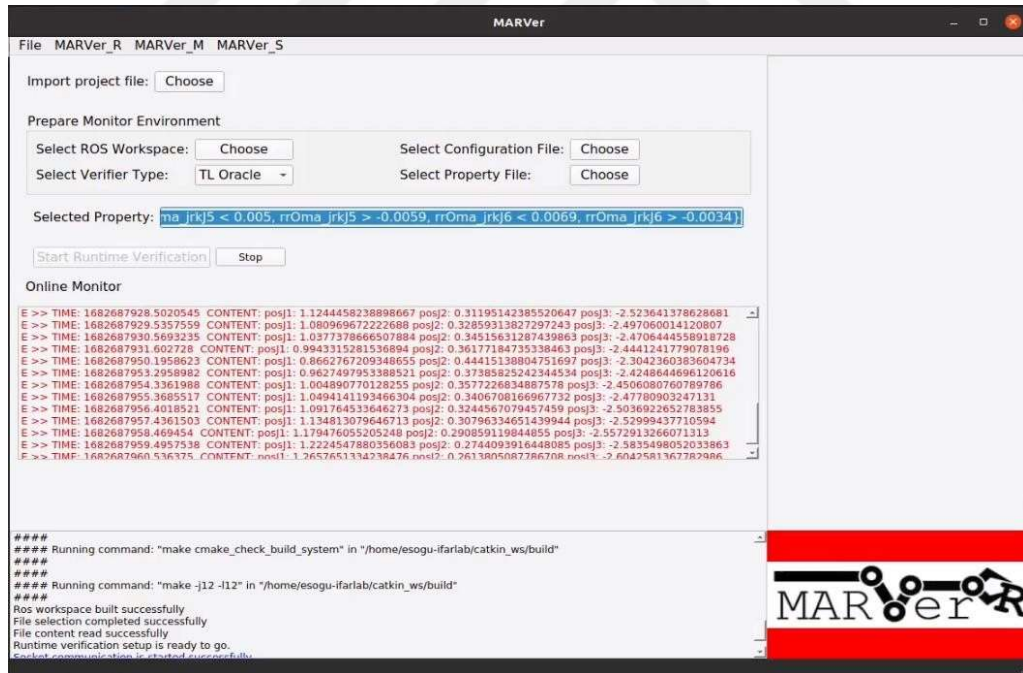
4.2. Doğrulama ve Onaylama Sistemi

Otomatik sistemlerin doğrulanması ve onaylanmasında farklı ölçütlere göre çalışan yazılım araçları bulunmaktadır. Bu çalışmada, robotik sistemler için kullanılmak üzere geliştirilen MARVer yazılım aracı kullanılmaktadır. Bu yazılım aracının, uygulama esnasında

alınan ekran görüntüsü Şekil 4.3’de görülmektedir.

Ayrıca, MARVer doğrulama yazılımının eklentisi olan ve OMA (Online Motion Analyzer) olarak adlandırılan yazılım, pozisyon değerlerinden nümerik olarak hız, ivme ve jerklerin hesaplamasını gerçekleştirmektedir. MARVer doğrulama aracı da bu değerlerin istenen koşullarda olup olmadığına bakarak doğrulama sürecini gerçekleştirmektedir.

Bu doğrulama yazılımından sağlanan ve SCADA sisteminde kullanılacak veriler, Çizelge 4.2’de verilmektedir. Bu tabloda, OMA tarafından yayınlanan robot kolun eklem hızları, ivmeleri ve jerkleri bulunmaktadır. Ayrıca, doğrulama sonucu da gene veri olarak sağlanmaktadır. Doğrulama sonucu, sistem çalışırken MARVer tarafından akan veriler üzerinde yapılan sorguların neticesi olarak ortaya çıkmaktadır. Buna göre, istenen koşulların sağlanmadığı anlarda false, aksi durumda true değer göndermektedir.



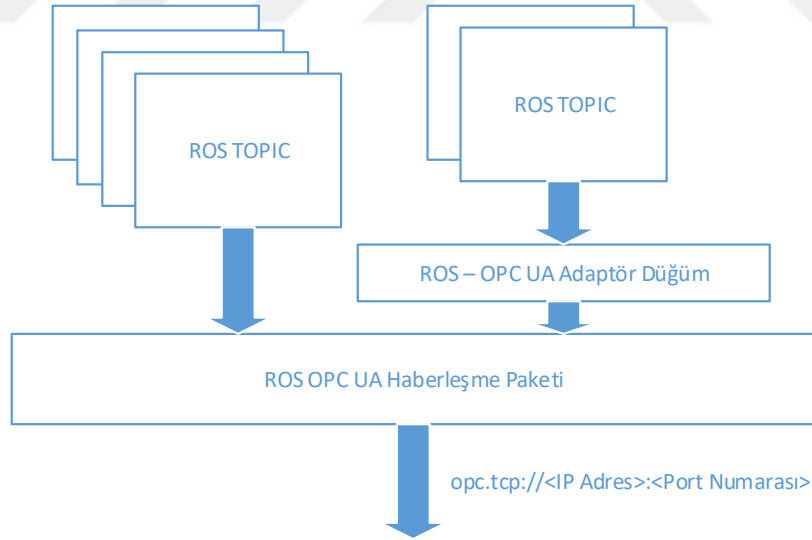
Şekil 4.3. MARVer Doğrulama Yazılımı Arayüzü

Çizelge 4.2. MARVer yazılım doğrulama yazılımının SCADA sisteminde kullanılacak verileri

Veri Adı	Veri Tipi	ROS Topic	ROS Topic Tipi	Tanımı
velJ1 – velJ6	double	/mam_oma	std_msgs/Float64MultiArray	6 eksen robot eklem hızları
accJ1 – accJ6	double	/mam_oma	std_msgs/Float64MultiArray	6 eksen robot eklem ivmeleri
jrkJ1 – jrkJ6	double	/mam_oma	std_msgs/Float64MultiArray	6 eksen robot eklem jerkleri
marverOMA	string	/mon_oma/monitor_verdict	std_msgs/string	Doğrulama sonucu

4.3. ROS ile OPC UA Arasında Adaptör Düğüm Geliştirme

ROS OPC UA haberleşme paketi (ros_opcua_communication), yayınlanan topiklerin hepsini OPC UA protokolüne göre birer nesne olarak, TCP üzerinden yayınlamaktadır. Genel şematik gösterim Şekil 4.4’de verilmektedir.

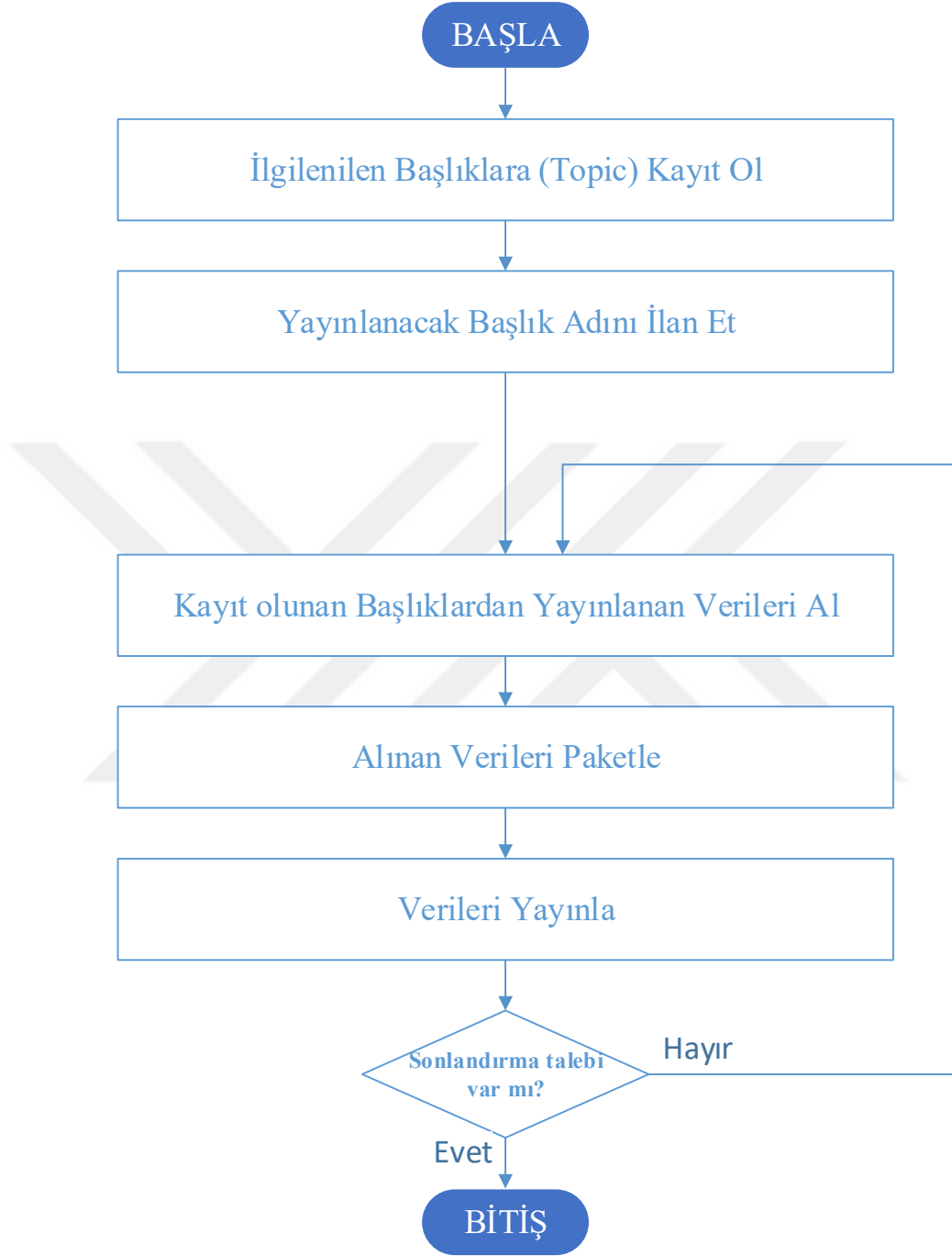


Şekil 4.4. ROS OPC UA haberleşme paketi ve adaptör düğümün şematik gösterimi

Bazı ROS Topic tiplerinde, özellikle vektörel veri tiplerinde, haberleşme paketi OPC UA formuna dönüştürmekte sorun çıkarmaktadır. Bu sorunu gidermek üzere, ROS- OPC UA adaptör düğüm geliştirilmiştir. Bazı, SCADA sistemine aktarılacak olan topicler bu adaptör

düğüm tarafından haberleşme paketi tarafından kullanılabilir formda topic olarak yayınlanmaktadır. Gelecek çalışma olarak, SCADA sisteminde kullanılması planlanan tüm topiclerin istenen sınıflandırma ile istenen forma dönüştürülmesini sağlayan ve bir grafik arayüz üzerinden kullanıcının seçim yapmasına olanak verecek bir yazılım geliştirilmesi katkı sağlayacaktır.

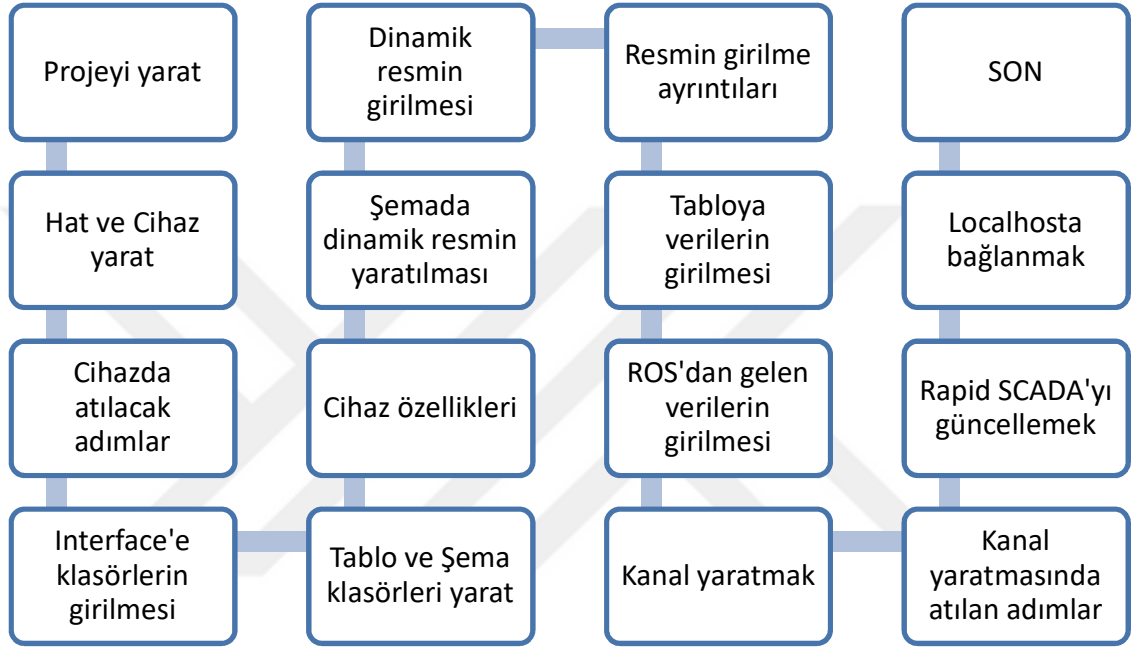
Şekil 4.5'te adaptör düğüm için akış diyagramı verilmektedir. Öncelikle, ilgilenilen topiclere kaydolunur. Daha sonra, birleştirilen verilerin yayınlanacağı topik için yayın ilanı oluşturulur. Bundan sonraki döngüsel süreç içinde, ilgilenilen topiclerdeki veriler alınır, paketlenir ve ilan edilen topic üzerinden yayınlanır.



Şekil 4.5. ROS ile OPC UA arası adaptör düğüm akış diyagramı

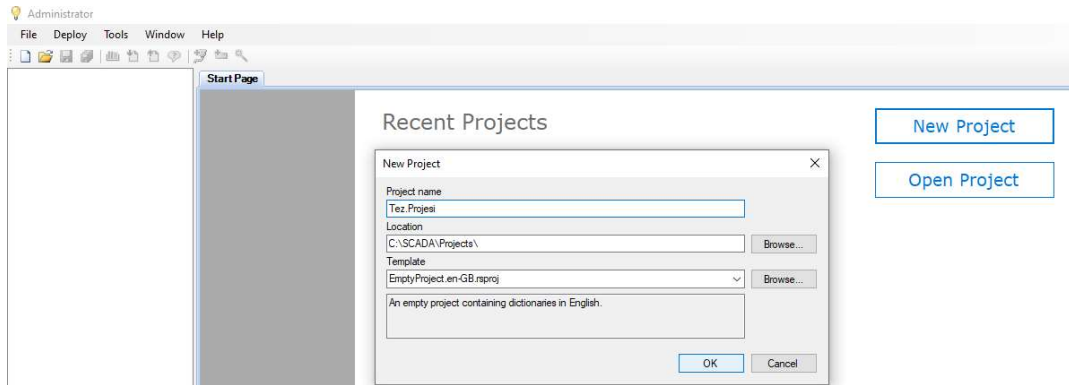
4.4. SCADA Tablo ve Şemaların Oluşturulması

Çalışmamızda Rapid SCADA’da hangi adımlar atılmıştır onları anlatacağız. Aşağıda Şekil 4.6’de yapılan adımların bütün hali verilmiştir:



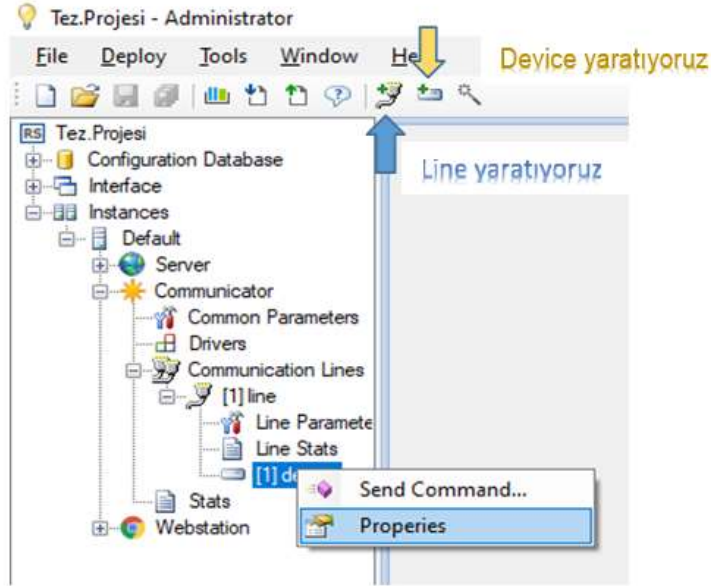
Şekil 4.6. Rapid SCADA uygulama geliştirme adımları

İlk olarak proje ismi verilmiş ve daha sonra hat ve cihaz yaratılmıştır. Şekil 4.7’te Proje isimlendirip yeni bir proje yaratıyoruz.



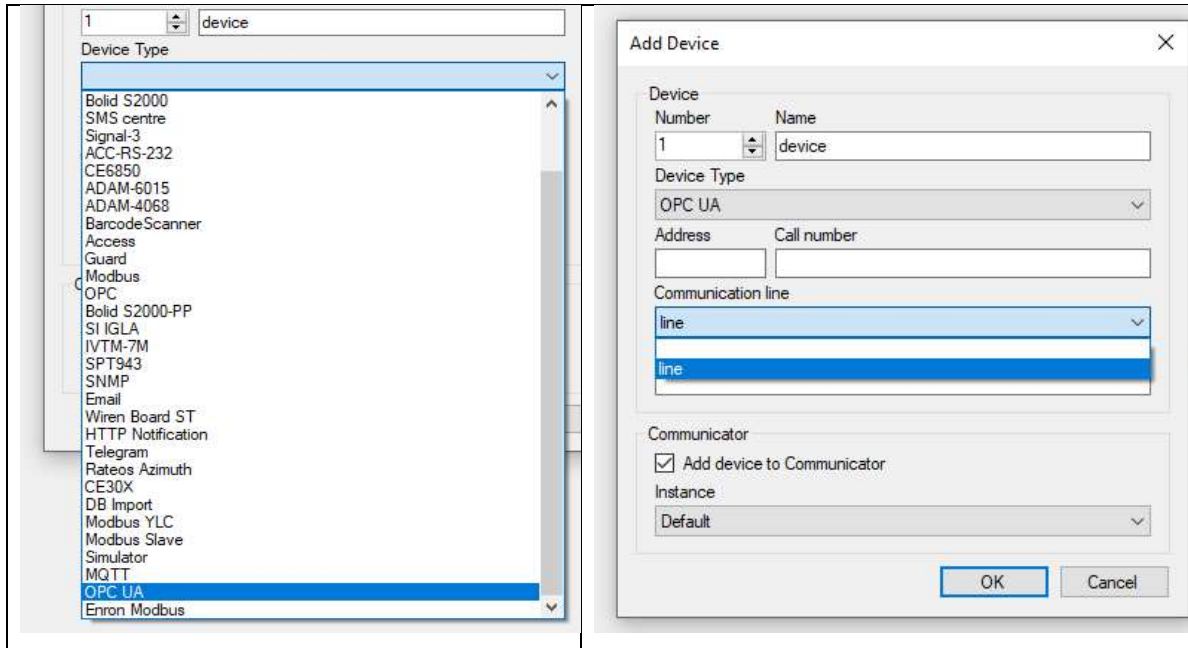
Şekil 4.7. Yeni Proje Yaratmak

Şekil 4.8’te mavi okla işaret edilen yere cihaz yaratılması için giriyoruz. Burada hat ve cihaz yarattığımız kısımları belirttik. Cihaza veri girmemiz için ‘Properties’ tıklıyoruz.



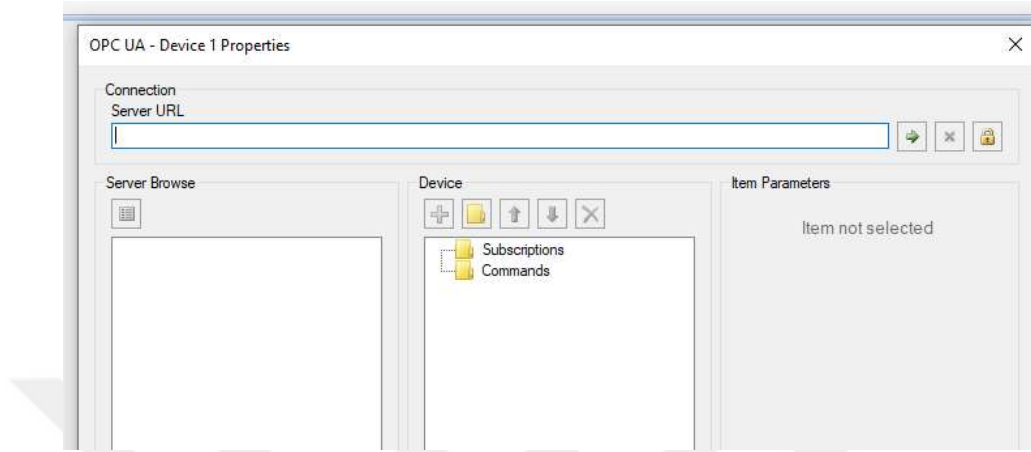
Şekil 4.8. Hat ve Cihaz yaratmak

Daha sonra Şekil 4.9’te cihaz tipini OPC UA seçip, hat kısmına yarattığımız hattımızı giriyoruz.



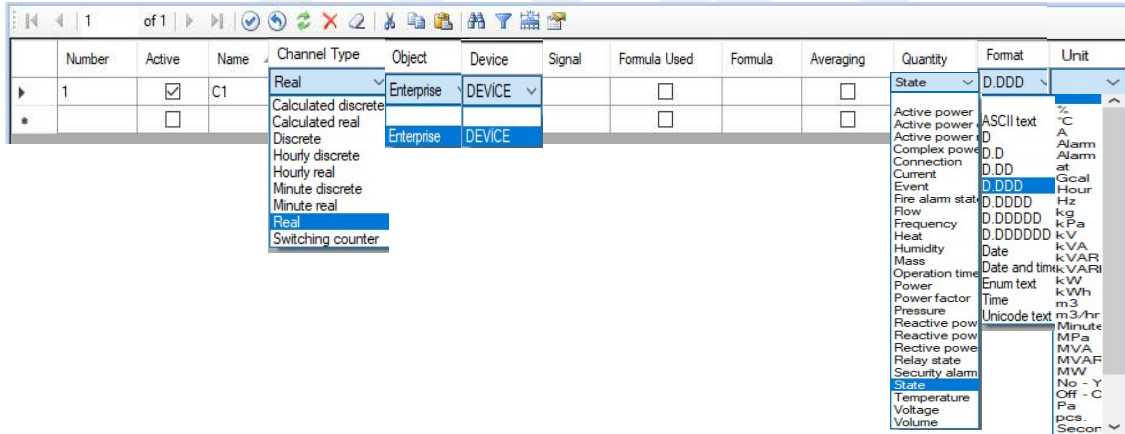
Şekil 4.9. Cihazda atılacak adımlar

Şekil 4.10'da cihaza veri girmek link veya aşağıda olan kısımda Şekil 4.11'deki gibi oluyor.



Şekil 4.10. Cihaz özellikleri

Burada URL bölümüne aldığımız verilerin adresini giriyoruz.



Output Channel	Write Events	Event Sound	Event on Change	Event on Undefined	Lower Alarm Limit	Lower Limit	Upper Limit	Upper Alarm Limit
	☐	☐	☐	☐				
	☐	☐	☐	☐				

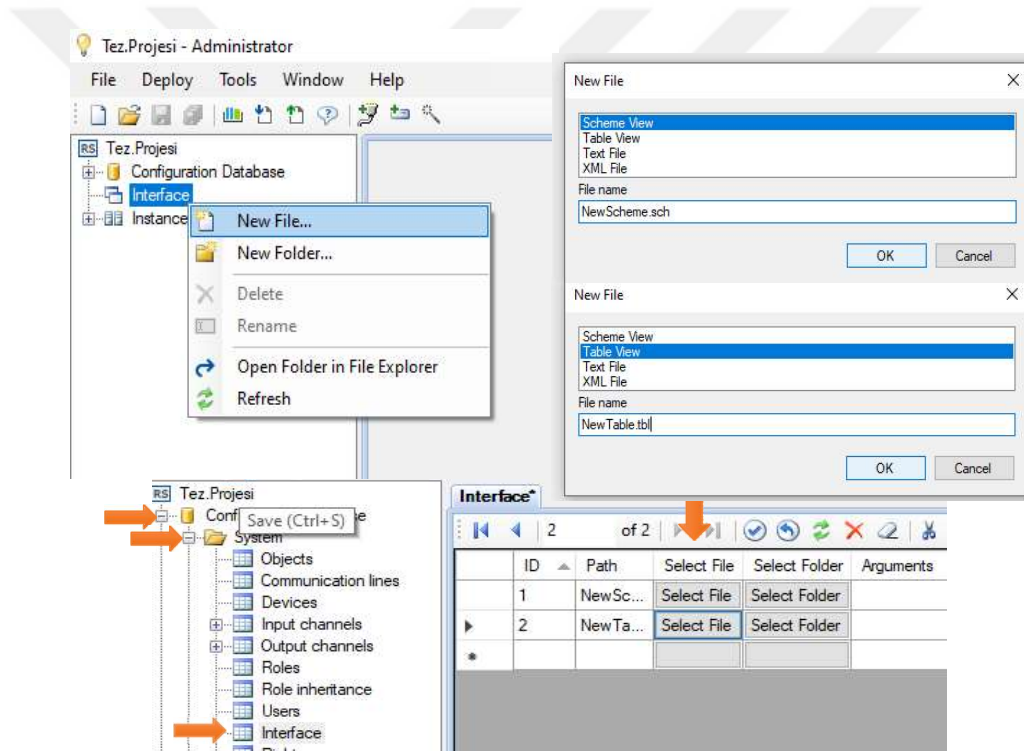
Şekil 4.11. Verinin sisteme girilmesi

Şekil 4.11'de gösterilen her başlığın anlatımı Çizelge 4.3'te verilmiştir:

Çizelge 4.3. Verilerin girilmesinde başlıkların anlamı

Adı	Tanımı
Active	Girdiğimiz datanı aktif etmediğimizde localhost'da tablo bize o veriyi göstermez
Name	Girdiğimiz veriyi isimlendirmek için vardır
Channel Type	Girdiğimiz verinin tipini gösterir
Object	Burada Enterprise seçilir
Device	Device seçilmediğinde yani boş bırakıldığında veri oynamıyor aynı şekilde kalıyor.
Signal	Device'dan girilen veriyi girmemiz için kullanılıyor
Formula Used	Bunu tıkladığımızda 'Server is unavailable' hatasını verir. Formula'ya bir komut değerlerinin hesaplanmasında kullanılan formüller sözlüğü yazılan zaman aktif edilebilir.
Formula	Giriş kanalı verilerinin ve çıkış kanallarının komut değerlerinin hesaplanmasında kullanılan formüller sözlüğü
Averaging	Girdiğimiz verinin ortalamasını ekrana yansıtır
Quantity	Girdiğimiz verinin durumunu localhost'da gösterir
Format	Girdiğimiz veriyi zaman veya tarih ya da text formatında gösterir
Unit	Girdiğimiz verinin yanına %, hz vd., şeklinde ekler
Output Channel	Localhost'da karşımıza pencere çıkarır ve bu değeri girmemizi ister
Write Events	Bunu seçmiş olursak, durumu aşağıda bize gösteriyor aksi taktirde Events tablosunda gözükmez
Event sound	Her artışta ses sinyali çıkarıyor, girdiğimiz limit değerleri aştığında sesle bizi uyarmaktadır ve aynı zamanda events tablosunda kırmızı renkle gösterir
Event on Change	Girdiğimiz değerin üzerine çıkanda renk değiştiriyor ve Write Events aktif edersek Events tablosunda da veriler renk değiştirecektir
Event on Undefined	Aktif edildiğinde olaya tanımsız kalır. Yani, EVENT SOUND aktif edilmiştir yalnız ses sinyalleri durmayız bunun sebebi ise Event On Undefined aktif edilmiş olmasıdır.
Lower Alarm Limit	Burada girdiğimiz sayıyı geçerse renk değiştirir tabloda low alarm

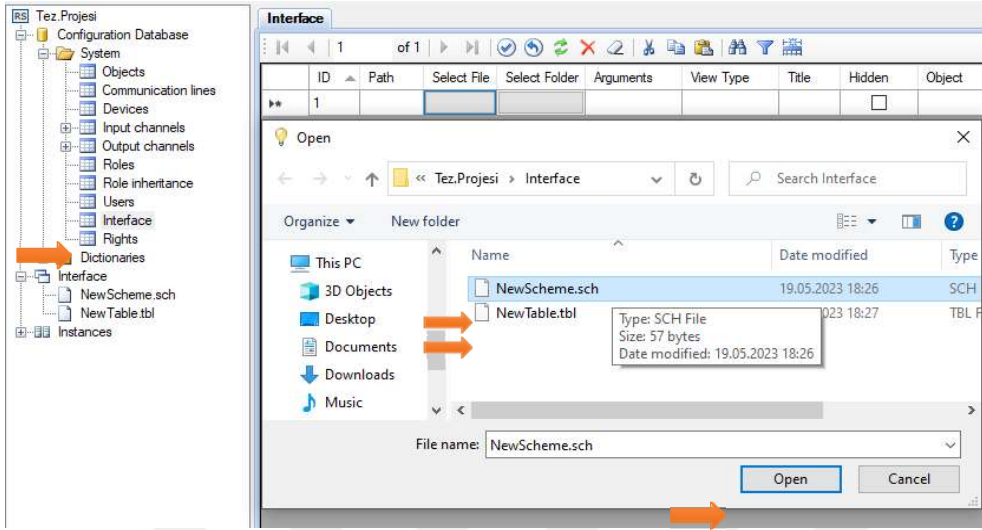
	limitini geçtiğini yazar eğer WRITE EVENTS aktif olundaysa ve bundan ilave olarak EVENT SOUND aktif etmişsek durmadan sinyaller verecektir
Lower Limit	Burada girdiğimiz sayı Events 'da gözükür ve renk değiştirir eğer EVENT ON CHANGE aktif ettiyse
Upper Limit	Burada girdiğimiz sayıyı geçerse turuncu renk gösterir tabloda
Upper Alarm Limit	Burada girdiğimiz sayıyı geçerse kırmızı renk gösteriyor tabloda ve eğer EVENT SOUND aktif etmişsek durmadan sinyaller verecektir



Şekil 4.12. Tablo ve Şema klasörlerini yaratmak

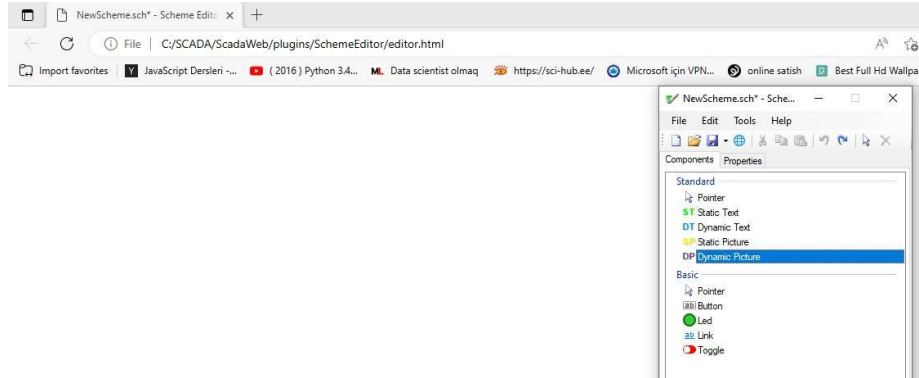
Şekil 4.12’de Interface’e girip yeni fayla tıklıyoruz ve Şema ve Tablo klasörlerini isimlendirerek yaratıyoruz. Şimdi ise yarattığımız klasörleri sisteme girmek için Select file’a tıklayarak bilgisayardan giriyoruz.

Şekil 4.13’te Interface’e tıklayarak oradan da Select file’a giriyoruz ve böylelikle klasörleri girmiş oluyoruz.



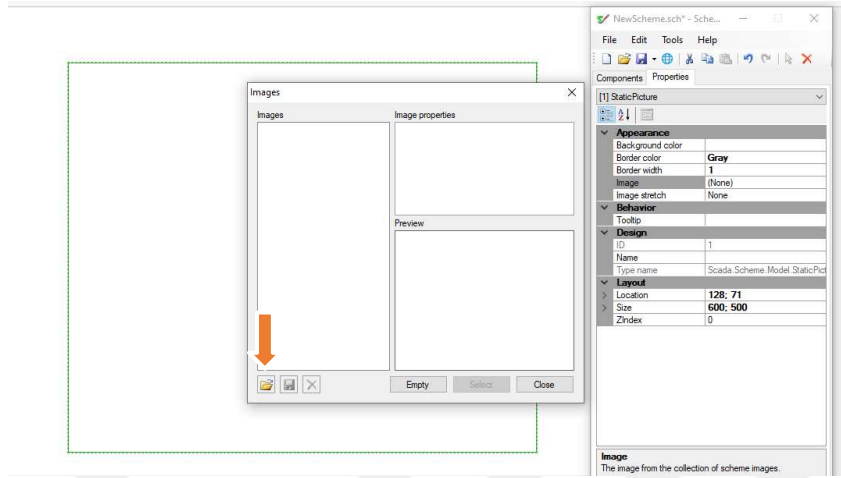
Şekil 4.13. Interface'e klasörlerin girilmesi

Resmi ise aşağıdaki adımları atarak SCADA sistemine yüklüyoruz.



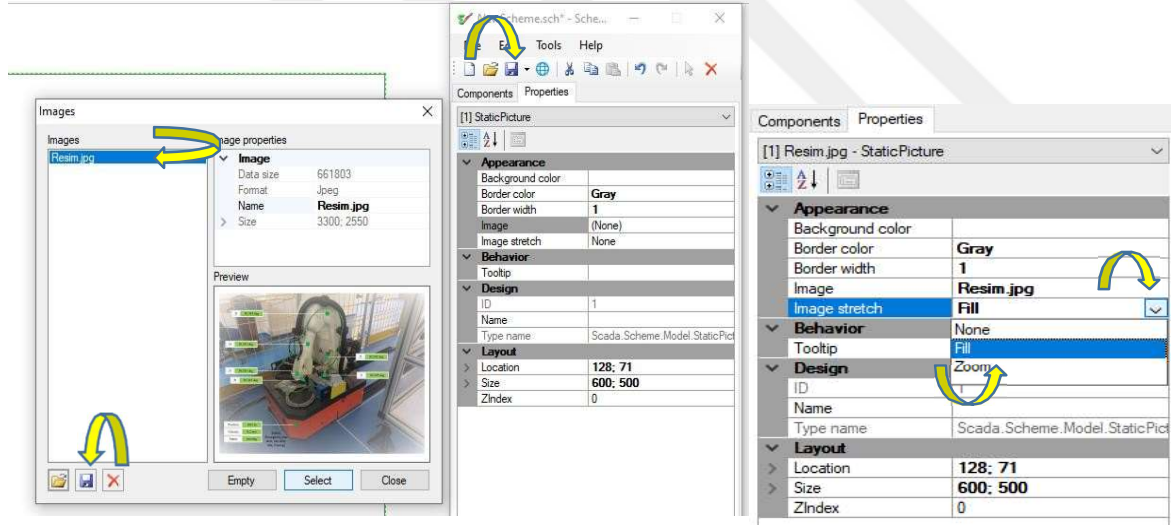
Şekil 4.14. Şemada dinamik resmin yaratılması

İlk adım yarattığımız Şema faylını açmaktır, daha sonra dynamic pic'e tıklıyoruz Şekil 4.14'te gösterildiği gibi



Şekil 4.15. Şemada dinamik resmin girilmesi

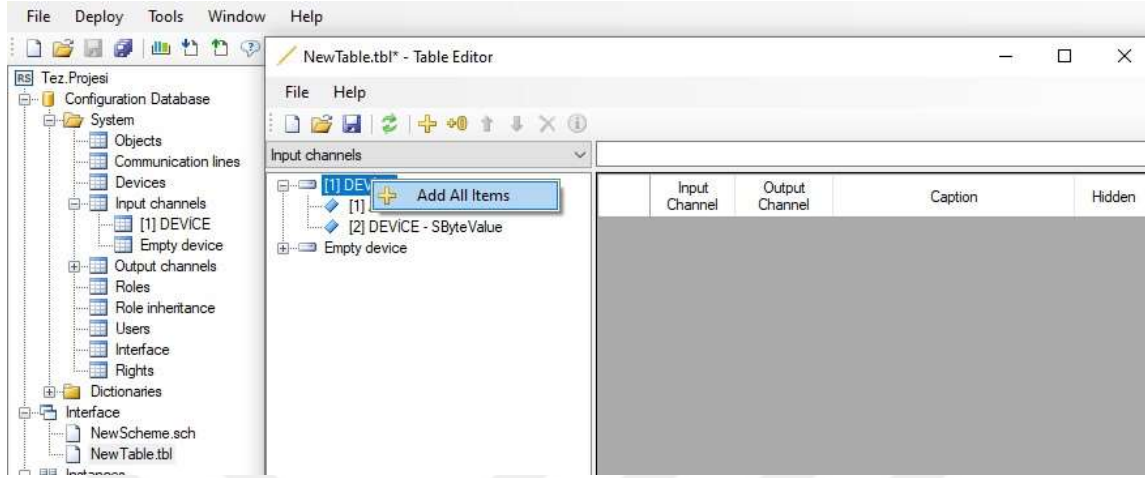
Şekil 4.15'te önceden gireceğimiz resmin ölçülerini belirleyip, resmi ok olan kısımdan seçerek save ediyoruz.



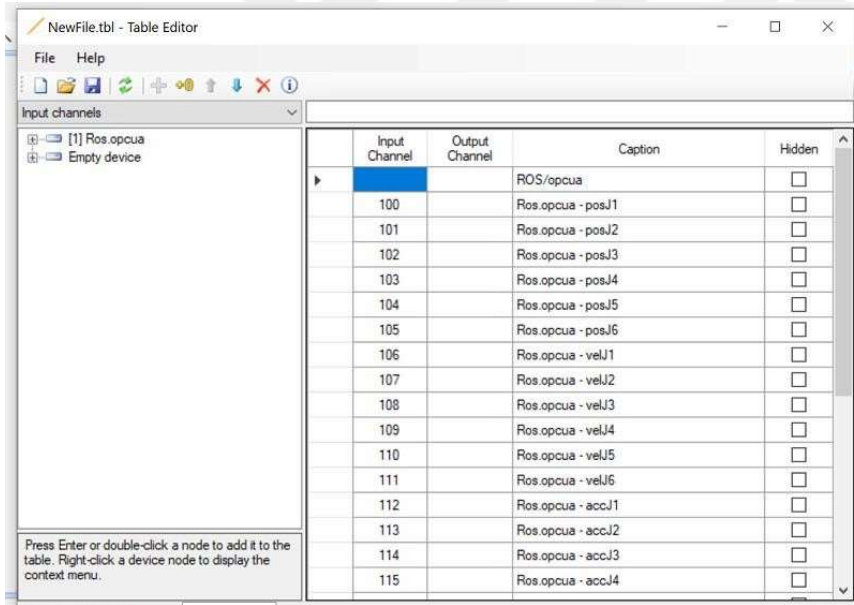
Şekil 4.16. Resmin girilme ayrıntıları

Şekil 4.16'de resmi ekledikten sonra ekrana doğru kapsamı için image stretch'den fill'i seçip save ediyoruz.

Elimizle girdiğimiz verileri tabloya eklenmesi Şekil 4.17'de gösterilmektedir

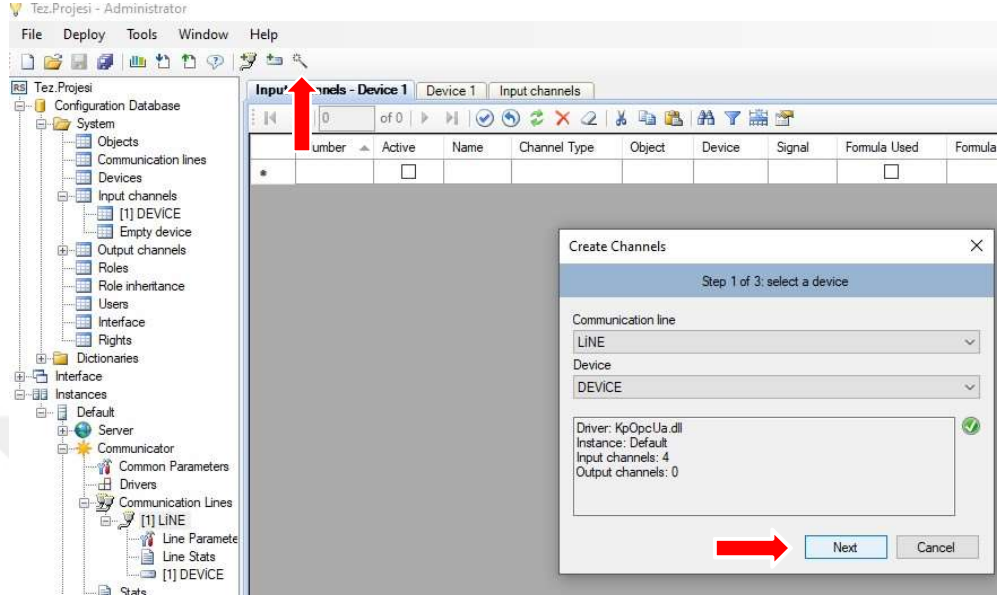


Şekil 4.17. Tabloya verilerin girilmesi



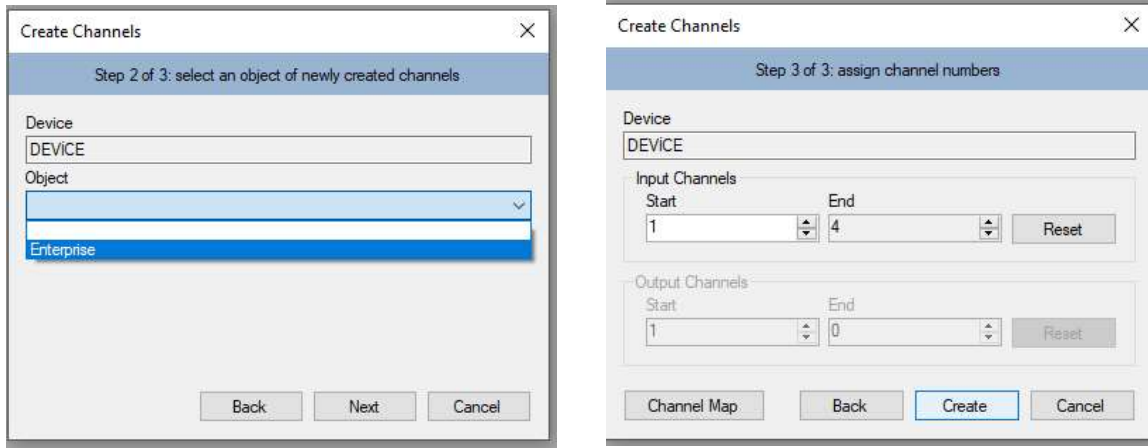
Şekil 4.18. ROS'dan gelen verinin girilmesi

Şekil 4.18'te veriler tabloya eklenmiştir ve localhost'da görüntülenmektedir.



Şekil 4.19. Kanal yaratmak

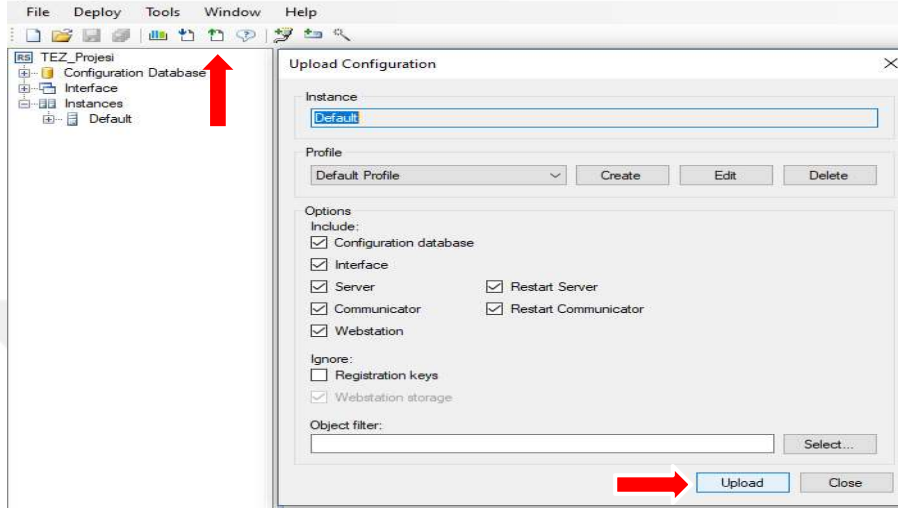
Şekil 4.19’te oklarla işaret ettiklerimize tıklayarak ‘Create Channel’ penceresini açıyoruz ve yarattığımız hatı ve cihazı girerek ilerliyoruz



Şekil 4.20. Kanal yaratılmasında adımlar

Şekil 4.20’da Object’de Enterprise seçiyoruz ve kaçtan kaç olmasını belirleyerek kanalımızı yaratmış oluyoruz.

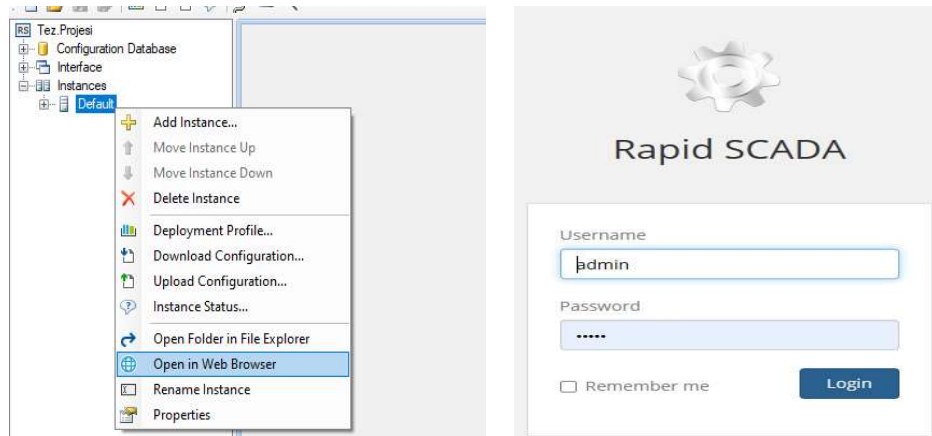
Her bir deęişiklik yaptığımızda, bu deęişiklikleri localhost’da görüntölemek için Upload yani güncellemek zorunluğumuz var.



Şekil 4.21. Rapid SCADA’yı güncellemek

Şekil 4.21’de güncellenmenin nasıl yapıldığı oklarla gösterilmiştir.

Localhosta giriş nasıl sağlandığını aşağıdaki şekil 4.22’de gösterilmektedir.



Şekil 4.22. Localhosta bağlanmak

Şekil 4.22’de Default’a sol tıklayarak Open in Web Browser’i açıyoruz burada şifre yerine ‘12345’ yazarak Localhosta girişi sağlıyoruz.

5. BULGULAR VE TARTIŞMA

Önerilen yaklaşımın kullanımını göstermek adına, deneysel çalışma yapılmış ve sonuçları paylaşılmıştır. Bu çalışmada, Şekil 4.1’de verilen akış şemasındaki adımlar işletilmiştir. Kullanılan deneysel sistemin resmi Şekil 4.2’de gösterilmektedir.

Deneysel sistemde bir kalite denetleme görevi çalıştırılmıştır. Bu görevde, robot kol ucundaki derinlik kamerası denetim yapılan yapı üzerinde 4 farklı konuma taşınmaktadır. Uygulama videosundan alınan kareler Şekil 5.1’da gösterilmektedir. Yapı üzerindeki dört pencereden içeri sokarak görevi tamamlamaktadır. 0s ile 15s arasında mobil platform ilk konumuna gelerek durmaktadır. Daha sonra, sadece robot kolu hareket etmekte, önce alt (15s-75s) ve sonra üst (90-165s) pencereden kamerayı yapı içine sokmaktadır. 180s içinde mobil platform ikinci konumuna giderek durmaktadır. Robot kol, önce üst (195s-255s) ve sonra alt (270s-330s) pencereden kamerayı yapı içine sokmaktadır. En son mobil platform göreve başladığı ilk konumuna dönmektedir (345s-360s).



0s



15s



30s



45s



60s



75s



90s



105s



120s



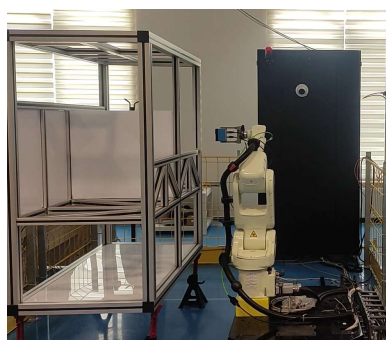
135s



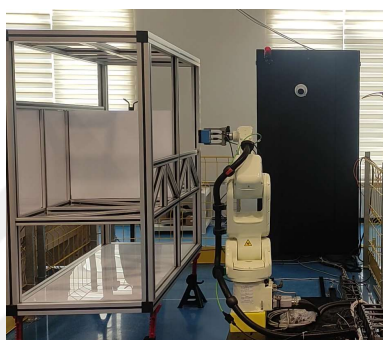
150s



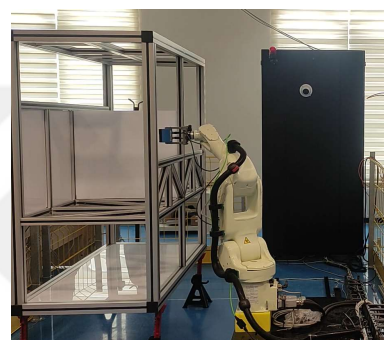
165s



180s



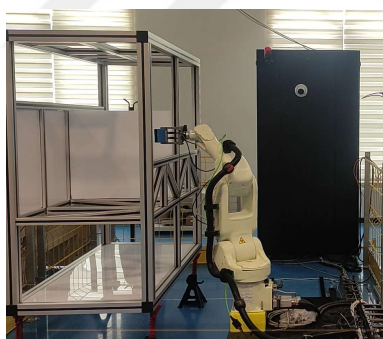
195s



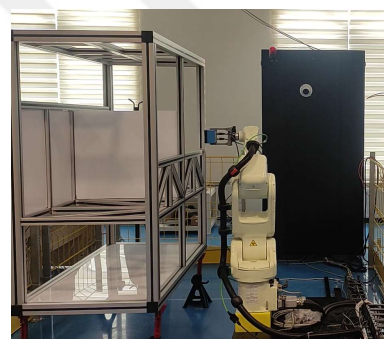
210s



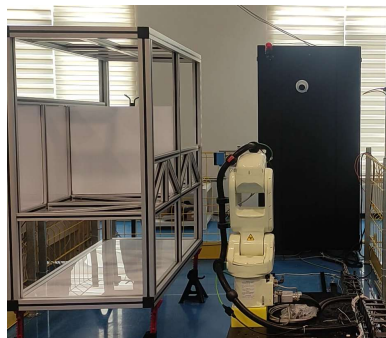
225s



240s



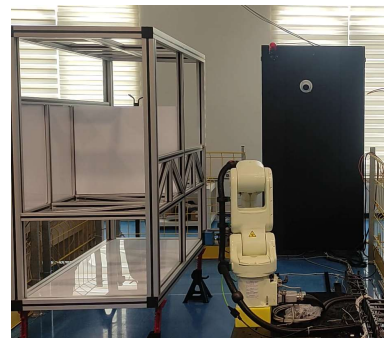
255s



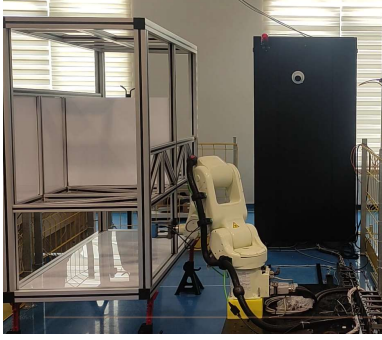
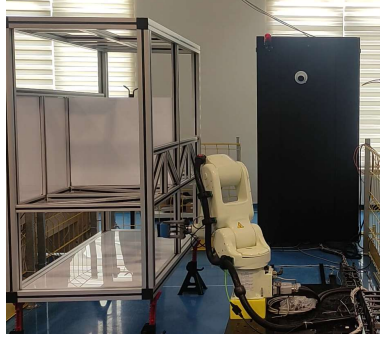
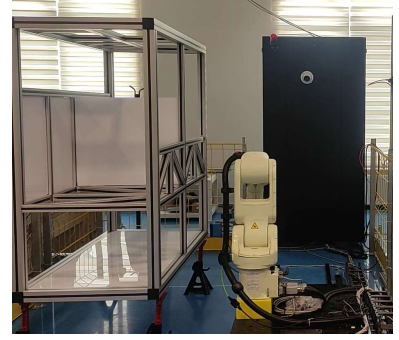
270s



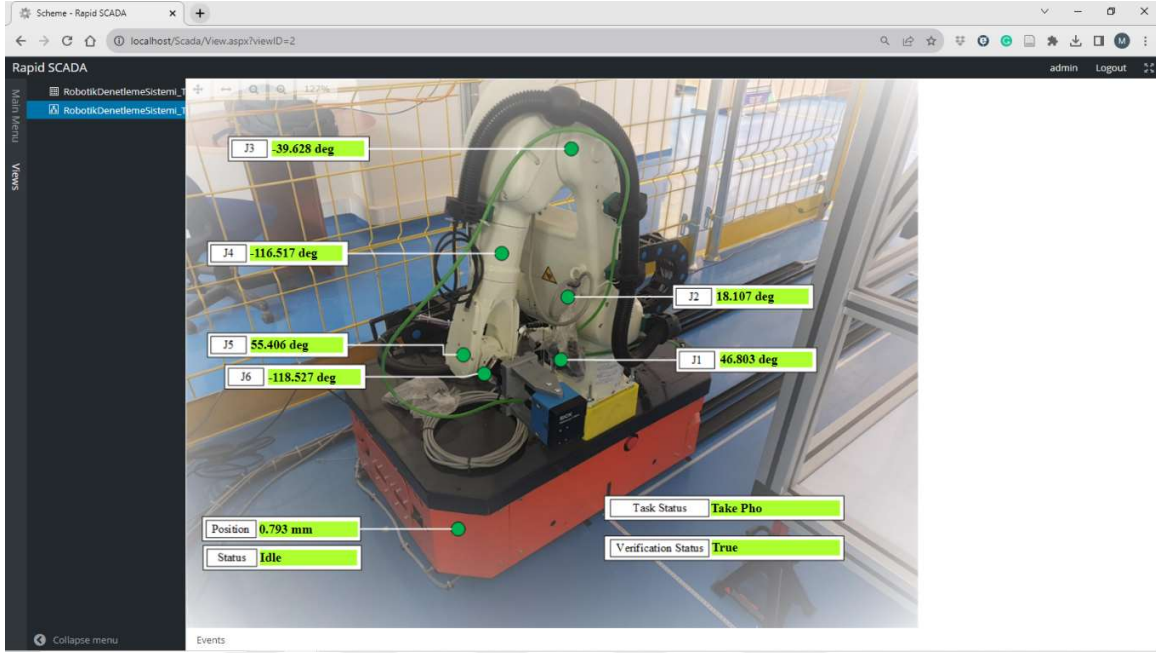
285s



300s

**315s****330s****345s****360s**

Şekil 5.1. Robotik Denetleme Sisteminin, Denetleme Görevi Esnasında Çekilen Video Kareleri



Şekil 5.2. SCADA Tasarı Görünümü

SCADA sisteminde, Çizelge 4.1’de verilen izlenecek olan robotik denetleme sisteminden temin edilen veriler ve Çizelge 4.2’de verilen doğrulama sisteminden sağlanan veriler görselleştirilmektedir. Bu verilerin görselleştirilmesinde iki farklı görünüm tasarlanmakta ve sonuçları paylaşılmaktadır. Şekil 5.2’de tasarı (scheme) görünümü verilmektedir. Bu görünümde, robotik denetleme sistemi resmi üzerinde ilişkili verilerin çevrimiçi değerleri izlenebilmektedir. Robot kolun eklem pozisyonları (J1-J6), burada verilmektedir. Döner eklem pozisyonları, radyan (rad) cinsinden verilmektedir. Ayrıca, mobil platformun, konumu, hızı ve durumu, bu görünümde verilmektedir. Platformunda lineer konumu ve hızı, milimetre (mm) ve milimetre/saniye (mm/s) cinsinde gösterilmektedir. Denetleme sisteminin, sistem tarafından işletilen görev durumu ve doğrulama sonuçları verilmiştir. Doğrulama sonuçları, doğru ve yanlış şeklinde olabilmektedir. Eğer, sistem isterleri çalışma anında sağlanıyorsa, doğru; sağlanmıyorsa, yanlış değer görünmektedir. Bu görünüm daha çok sistemde anlık olarak önemli verilerin izlenmesini sağlamaktadır.

Item	Current	14:00	15:00
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Konumu (Birim: deg)	90.169	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Konumu (Birim: deg)	-8.521	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Konumu (Birim: deg)	-71.068	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Konumu (Birim: deg)	-89.330	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Konumu (Birim: deg)	90.337	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Konumu (Birim: deg)	-131.694	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Hızı (Birim: deg/s)	-1.860	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Hızı (Birim: deg/s)	0.466	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Hızı (Birim: deg/s)	0.89	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Hızı (Birim: deg/s)	-0.844	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Hızı (Birim: deg/s)	-1.660	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Hızı (Birim: deg/s)	0.522	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş İnmesi (Birim: deg/s2)	-0.116	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş İnmesi (Birim: deg/s2)	-0.037	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş İnmesi (Birim: deg/s2)	-0.147	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş İnmesi (Birim: deg/s2)	-0.059	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş İnmesi (Birim: deg/s2)	-0.100	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş İnmesi (Birim: deg/s2)	-0.101	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Jerki (Birim: deg/s3)	0.052	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Jerki (Birim: deg/s3)	-0.058	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Jerki (Birim: deg/s3)	-0.165	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Jerki (Birim: deg/s3)	0.020	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Jerki (Birim: deg/s3)	0.050	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Jerki (Birim: deg/s3)	-0.103	---	*
RobotikDenetlemeSistemi - Mobil Platform Doğrusal Konumu (Birim: mm)	0.793	---	*
RobotikDenetlemeSistemi - Mobil Platform Durumu	Idle	---	*
RobotikDenetlemeSistemi - Sistem Görev Durumu	Manipula	---	*
RobotikDenetlemeSistemi - Sistem Görev Numarası	352,692	---	*
RobotikDenetlemeSistemi - Doğrulama Durumu (true: verified, false: not verified)	True	---	*

(a)

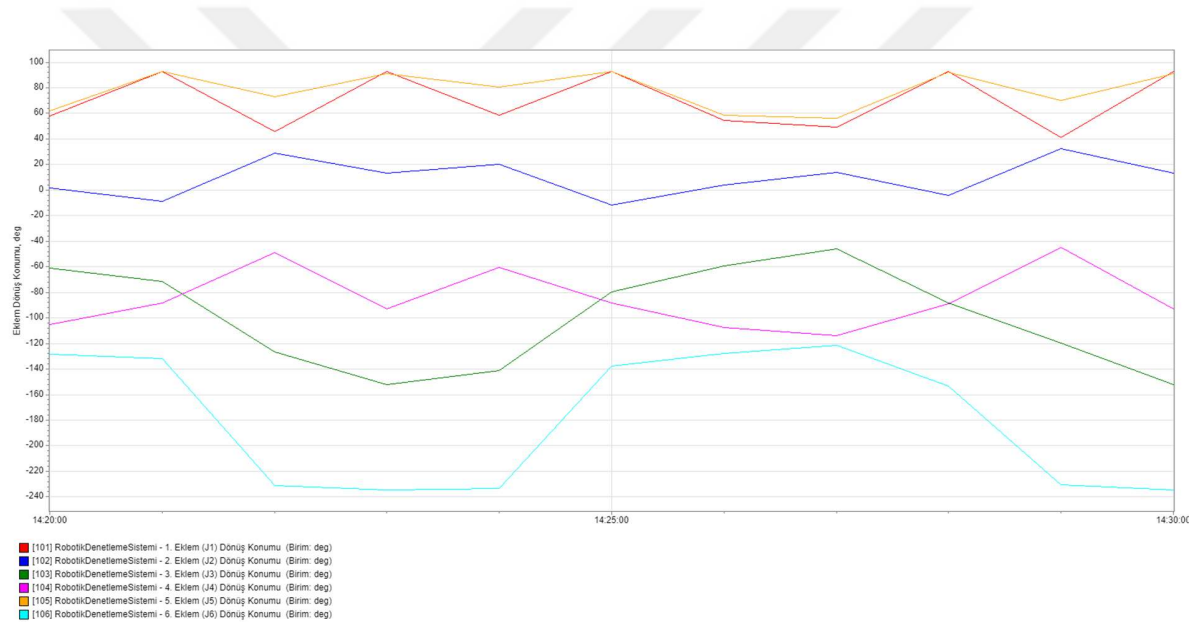
Item	Current	14:00	15:00
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Konumu (Birim: deg)	90.169	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Konumu (Birim: deg)	-8.521	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Konumu (Birim: deg)	-71.068	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Konumu (Birim: deg)	-89.330	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Konumu (Birim: deg)	90.337	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Konumu (Birim: deg)	-131.694	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Hızı (Birim: deg/s)	-1.860	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Hızı (Birim: deg/s)	0.466	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Hızı (Birim: deg/s)	0.89	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Hızı (Birim: deg/s)	-0.844	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Hızı (Birim: deg/s)	-1.660	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Hızı (Birim: deg/s)	0.522	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş İnmesi (Birim: deg/s2)	-0.116	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş İnmesi (Birim: deg/s2)	-0.037	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş İnmesi (Birim: deg/s2)	-0.147	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş İnmesi (Birim: deg/s2)	-0.059	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş İnmesi (Birim: deg/s2)	-0.100	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş İnmesi (Birim: deg/s2)	-0.101	---	*
RobotikDenetlemeSistemi - 1. Ekleme (J1) Dönüş Jerki (Birim: deg/s3)	0.052	---	*
RobotikDenetlemeSistemi - 2. Ekleme (J2) Dönüş Jerki (Birim: deg/s3)	-0.058	---	*
RobotikDenetlemeSistemi - 3. Ekleme (J3) Dönüş Jerki (Birim: deg/s3)	-0.165	---	*
RobotikDenetlemeSistemi - 4. Ekleme (J4) Dönüş Jerki (Birim: deg/s3)	0.020	---	*
RobotikDenetlemeSistemi - 5. Ekleme (J5) Dönüş Jerki (Birim: deg/s3)	0.050	---	*
RobotikDenetlemeSistemi - 6. Ekleme (J6) Dönüş Jerki (Birim: deg/s3)	-0.103	---	*
RobotikDenetlemeSistemi - Mobil Platform Doğrusal Konumu (Birim: mm)	0.793	---	*
RobotikDenetlemeSistemi - Mobil Platform Durumu	Idle	---	*
RobotikDenetlemeSistemi - Sistem Görev Durumu	Manipula	---	*
RobotikDenetlemeSistemi - Sistem Görev Numarası	352,692	---	*
RobotikDenetlemeSistemi - Doğrulama Durumu (true: verified, false: not verified)	True	---	*

(b)

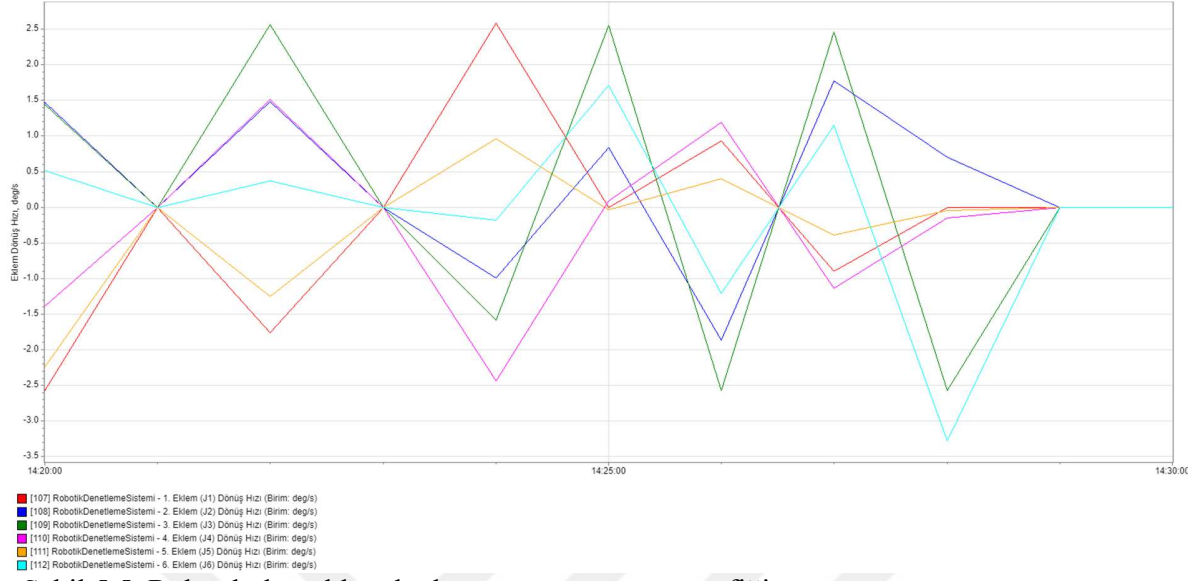
Şekil 5.3. SCADA Tablo Görünümü: (a) Tüm ekran görünümü, (b) Tablo Kısmı Yakın

SCADA sistemindeki bir diğer görünüm tablo görünümüdür. Şekil 5.3'te tablo görünümü verilmektedir. Tablo görünümünde, denetleme sistemi ve doğrulama sisteminden temin edilen ve kayıt altına alınmak istenen tüm veriler bulunmaktadır. Verilerin güncel değerleri ve belli saatlerdeki değerleri görünmektedir. Bu tablodaki veriler aynı zamanda kayıt altına da alınmaktadır.

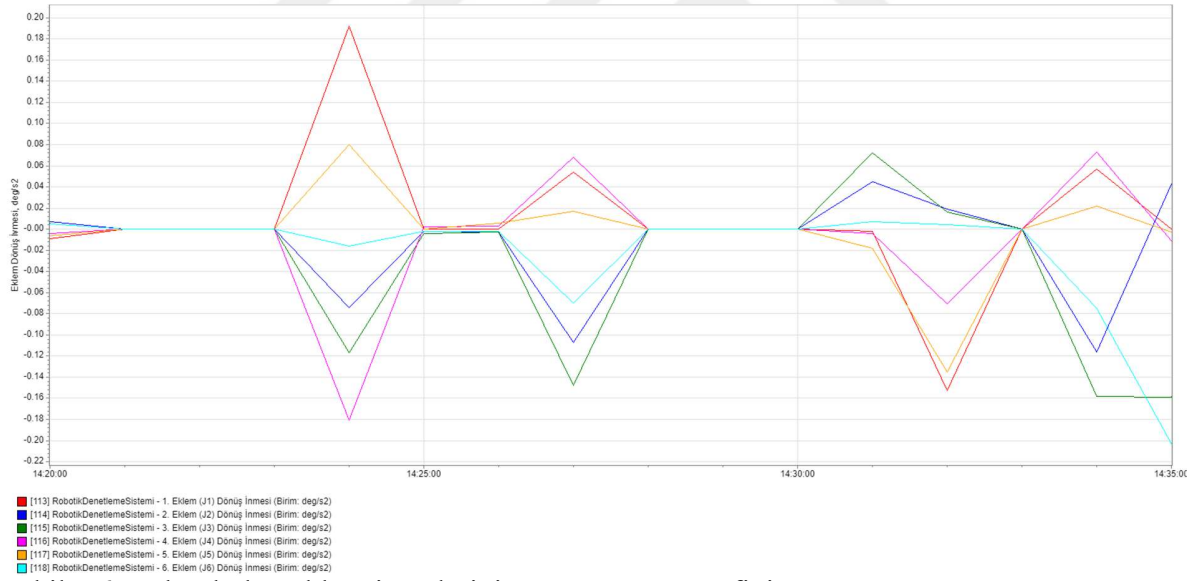
Tablodaki verilerin değerleri zamana göre grafik olarak da görüntülenebilmektedir. Şekil 5.4'te robot kol eklem değişkenleri (J1-J6) grafik olarak gösterilmektedir. Sırasıyla, Şekil 5.5-7'te, doğrulama sisteminden temin edilen eklem hızları, ivmeleri ve jerkleri görülmektedir.



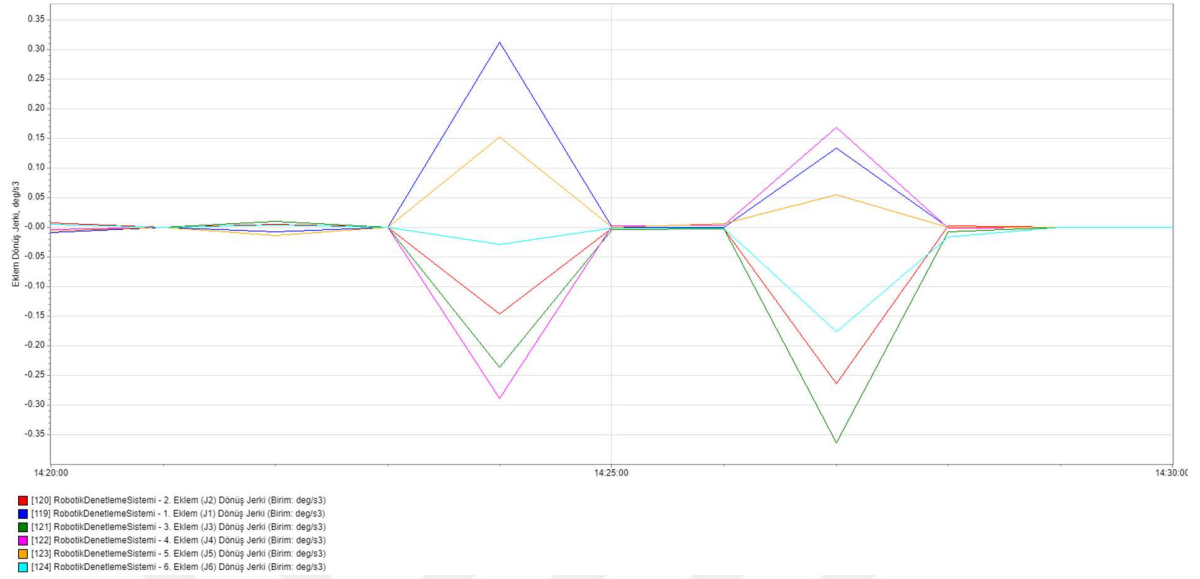
Şekil 5.4. Robot kolun eklem pozisyonlarının zamana göre grafiği



Şekil 5.5. Robot kolun eklemler hızlarının zamana göre grafiği

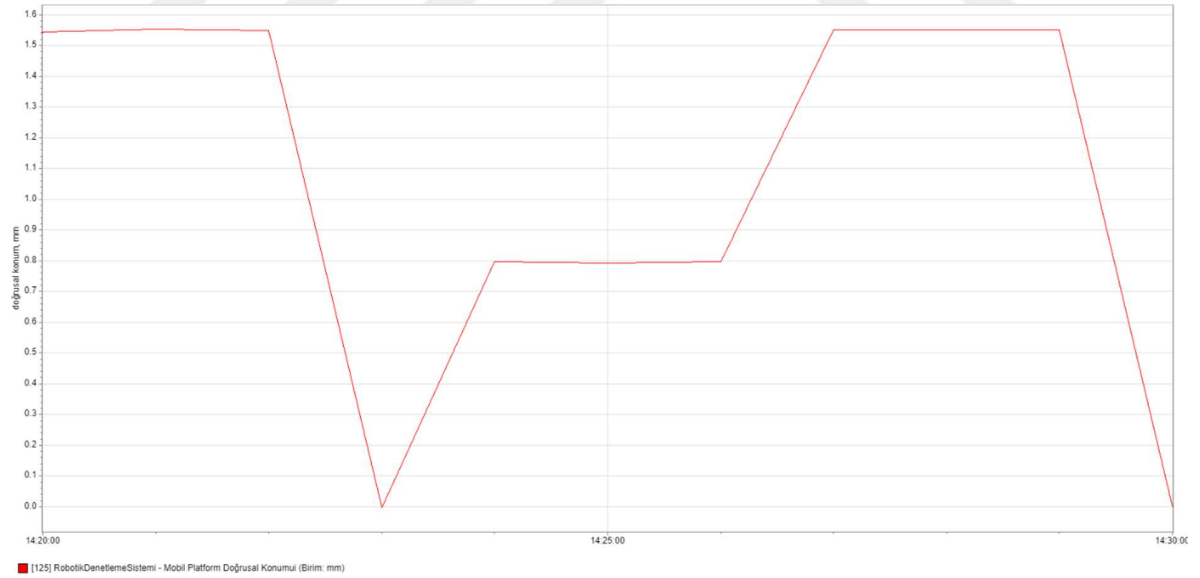


Şekil 5.6. Robot kolun eklemler ivmelerinin zamana göre grafiği

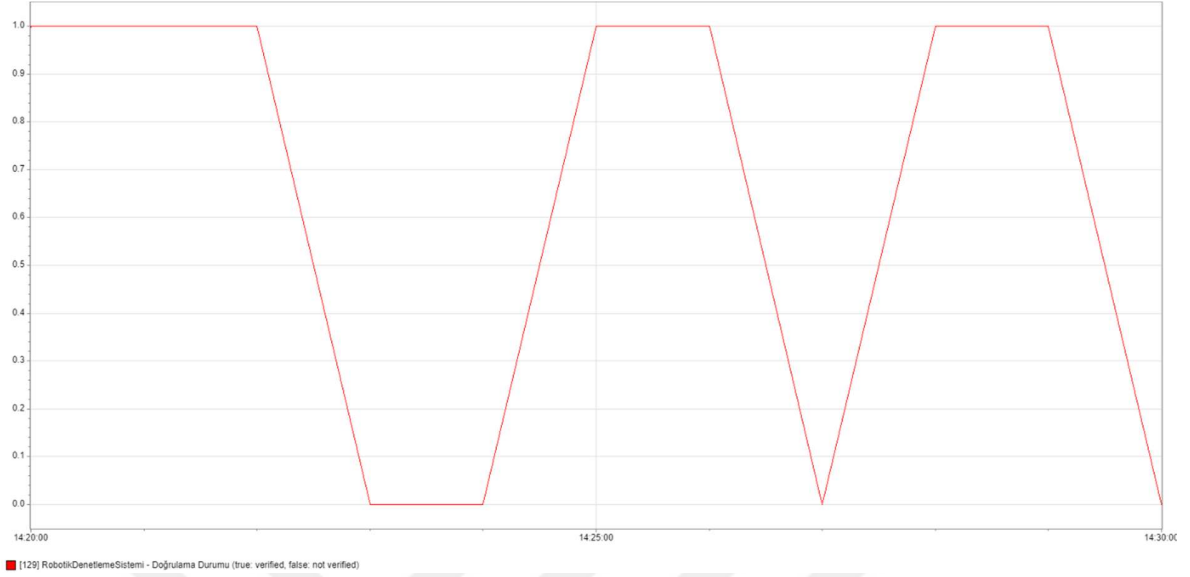


Şekil 5.7. Robot kolun eklem jerklerinin zamana göre grafiği

Ayrıca, mobil platform üzerinden elde edilen platform pozisyon da Şekil 5.8’ de görülmektedir.



Şekil 5.8. Mobil platformun pozisyonunun zamana göre grafiği



Şekil 5.9. Doğrulama Sonucunun Zamana Göre Diyagramı (0: İsterler sağlamıyor, 1: İsterler sağlanıyor)

Yazılım doğrulama ile ilgili MARVer yazılım aracından gelen doğrulama sonuçları, Şekil 5.9'da görüldüğü gibi SCADA sistem görünümünde gösterilmektedir. MARVer, doğrulama aracındaki MARVer-S içindeki servis bileşeni olan çevrimiçi hareket analizatörü (OMA) Şekil 5.5-7'te verilen verileri sağlamaktadır. MARVer-R bileşeni kapsamında isterlerin sorgulandığı çalışma anı doğrulama sürecinde, robot eklemleri pozisyon, hız, ivme ve jerk değerlerinin, Çizelge 5.1'de verilen limitler içinde olup olmadığı sorgulanmaktadır. Bu sorgulama sonuçları, Şekil 5.9'da gösterilmektedir. Mantıksal olarak sıfır (0) olduğunda isterler sağlanmamakta, yani limitler aşılmaktadır. Mantıksal olarak bir (1) olduğunda ise isterler sağlanmaktadır. Bu uygulamada, herhangi bir değer limitler dışına çıktığında, isterlerin sağlanmadığı sonucu çıkmaktadır.

Yapılan uygulama sonuçlarında görüldüğü gibi, otomatik bir sisteme entegre edilen yazılım doğrulama araçlarının sonuçları, SCADA sisteminden izlenebilmektedir. Böylece, SCADA sistemi hem sistemin uzaktan izlenmesi ve denetiminde kullanılabilmekte hem de yazılım doğrulamalarının sonuçlarını görmek için kullanılabilmektedir.

Çizelge 5.1. Çalışma Anı Doğrulama İsterleri

	Alt Limit	Üst Limit
1nci Eklem Pozisyonu (J1)	39.82 deg	92.82 deg
2nci Eklem Pozisyonu (J2)	-22.34 deg	33.80 deg
3ncü Eklem Pozisyonu (J3)	-151.83 deg	-20.62 deg
4ncü Eklem Pozisyonu (J4)	-125.47 deg	-43.54 deg
5nci Eklem Pozisyonu (J5)	52.71 deg	92.81 deg
6ncı Eklem Pozisyonu (J6)	-234.33 deg	-109.43 deg
1nci Eklem Hızı (J1)	-2.86 deg/s	2.86 deg/s
2nci Eklem Hızı (J2)	-2.86 deg/s	1.72 deg/s
3ncü Eklem Hızı (J3)	-2.86 deg/s	2.86 deg/s
4ncü Eklem Hızı (J4)	-2.29 deg/s	2.29 deg/s
5nci Eklem Hızı (J5)	-2.29 deg/s	2.29 deg/s
6ncı Eklem Hızı (J6)	-3.44 deg/s	4.01 deg/s
1nci Eklem İvmesi (J1)	-0.24 deg/s ²	0.22 deg/s ²
2nci Eklem İvmesi (J2)	-0.22 deg/s ²	0.14 deg/s ²
3ncü Eklem İvmesi (J3)	-0.19 deg/s ²	0.19 deg/s ²
4ncü Eklem İvmesi (J4)	-0.22 deg/s ²	0.19 deg/s ²
5nci Eklem İvmesi (J5)	-0.21 deg/s ²	0.19 deg/s ²
6ncı Eklem İvmesi (J6)	-0.13 deg/s ²	0.27 deg/s ²
1nci Eklem Jerki (J1)	-0.38 deg/s ³	0.38 deg/s ³
2nci Eklem Jerki (J2)	-0.33 deg/s ³	0.26 deg/s ³
3ncü Eklem Jerki (J3)	-0.33 deg/s ³	0.37 deg/s ³
4ncü Eklem Jerki (J4)	-0.37 deg/s ³	0.34 deg/s ³
5nci Eklem Jerki (J5)	-0.34 deg/s ³	0.29 deg/s ³
6ncı Eklem Jerki (J6)	-0.19 deg/s ³	0.40 deg/s ³

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, SCADA sistemleri için yeni bir yaklaşım ortaya konulmuş ve gerçek sistem testleri yapılmıştır. Öncelikle, otomatik sistemin SCADA sisteminde izlenmesi ve kontrolü için sistem verilerinin belirlenmesi süreci anlatılmıştır. Ayrıca, sistemin doğrulama araçlarının SCADA sistemine aktarılacak verilerin belirlenmesi süreçleri de anlatılmıştır. Verilerin belirlenmesi sonrasında, bu verilerin SCADA sistemine aktarımına yönelik çalışmalar yapılmıştır. SCADA sistemlerinde yaygın olarak kullanılan protokollerden birisi OPC UA, bu çalışmada kullanılmıştır. Uzaktan izleme ve denetim yapılacak olan sistem ile SCADA sistemi arasında verilerin aktarımında bu protokol kullanılmıştır.

Bu çalışmada ele alınan robotik denetleme sistemi, robotik sistemler için yaygın kullanılan ROS ortamında çalışmaktadır. Verilerin aktarımı için ROS ortamındaki mesajları, OPC UA protokolü üzerinden iletecek bir yapı geliştirilmiştir. Aynı zamanda, yazılım doğrulaması için kullanılan MARVer yazılım doğrulama aracı da sonuçları aynı altyapı üzerinden SCADA sistemine ulaştırmaktadır. Böylece, SCADA sistemi hem robotik denetim sistemini izleme ve denetimde, hem de yazılım doğrulama aracını sonuçlarını izlemek için kullanılabilir.

Önerilen yaklaşımın çalışmasını göstermek adına, lab ortamında kurulan bir robotik denetleme sistemi çalıştırılmıştır. Sistem verileri ve doğrulama yazılımının çıktıları SCADA sisteminde gösterilmiştir. Böylece, literatüre yeni bir yaklaşım ve bu yaklaşımının gerçek sistem üzerinde gösterimi ile katkı sağlanmıştır.

Gelecekte, önerilen yaklaşımın farklı sistemler ve altyapılarda kullanımına yönelik metodolojiler geliştirilebilir. Farklı sistemlerde uygulamalarını görmek mümkün olabilecektir. Ayrıca, doğrulama aracının entegrasyonu sonrasında, otomatik sistem yazılımında değişiklikler yapılarak doğrulama aracı çalıştırılabilir. Böylece, SCADA sistemine entegre bir doğrulama aracının, sistem kurulumu sonrası bakım süreçlerinde kullanımı gösterilmiş olacaktır.

KAYNAKLAR DİZİNİ

- Aghenta, L.O., Iqbal, M.T., (2019). Development of an IoT Based Open Source SCADA System for PV System Monitoring, 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE); May 5-8; Edmonton, AB, Canada.
- Antao, L., Pinto, R., Reis, J., Gonçalves, G., (2018). Requirements for Testing and Validating the Industrial Internet of Things, 2018 IEEE International Conference on Software Testing, Verification, and Validation Workshops, Västerås, Sweden, 110-115
- Babay, A., Tantillo, T., Aron, T., Platania, M., Amir, Y., (2018). Network-Attack-Resilient Intrusion-Tolerant SCADA for the Power Grid, 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 255-266.
- Bere, M., Muyingi, H., (2015). *Initial investigation of Industrial Control System (ICS) security using Artificial Immune System (AIS)*, 2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), , 79-84, Windhoek, Namibia.
- Behrmann, G., David, A., & Larsen, K. G. (2004). A Tutorial on Uppaal. In Formal Methods for the Design of Real-Time Systems (pp. 200–23
- Chan, A.C.F., Zhou, J., (2022). Home Applied Cryptography and Network Security Workshops Conference paper Toward Safe Integration of Legacy SCADA Systems in the Smart Grid, Applied Cryptography and Network Security Workshops.
- Demirci, D., Özkan, M., Ergün, H.C., Şahin, M.T., Demirci, Z., Ozkan, M., Ergun, H.C., Sahin, M.T., Yazıcı, A., (2022). MARVer: A Tool for Verification of Robotic System's Safety, Fırat Üniversitesi Uzay ve Savunma Teknolojileri Dergisi, 1(1), 300-305.

- Demirci, Z., (2023). Verification and Validation of Robotic System Safety [Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü].
- Erez, N., Wool, A., (2015). Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems, International Journal of Critical Infrastructure Protection, 10, 59-70.
- Ferrando, A., Cardoso, R.C., Fisher, M., Ancona, D., Franceschini, L., Mascardi, V., (2020). ROSMonitoring: A Runtime Verification Framework for ROS," in Towards Autonomous Robotic Systems, 387-399.
- Fernhill, (2023). Fernhill SCADA Web Sayfası, Erişim: <https://www.fernhillsoftware.com/index.php>, Erişim tarihi: 20.07.2023.
- FreeSCADA, (2023). FreeSCADA Web Sayfası, Erişim: <https://www.freescada.co.uk/>, Erişim tarihi: 20.07.2023.
- freeopcua, (2023). freeopcua kod deposu linki, Erişim:<https://github.com/FreeOpcUa/freeopcua>, Erişim tarihi: 20.07.2023.
- Gao, J., Liu, J., Rajan, B., Nori, R., Fu, B., Xiao, Y., Liang, W., Philip Chen, C.L., (2014). *SCADA communication and security issues*, Security and Communication Networks, 7175-194.
- Giraldo, J., Sarkar, E., Cardenas, A. A., Maniatakos, M., Kantarcioglu, M., (2017). *Security and Privacy in Cyber-Physical Systems: A Survey of Surveys*, IEEE Design Testing, 34(4)7-17.
- Ghosh, S., Sampalli, S., (2019). A Survey of Security in SCADA Networks: Current Issues and

Future Challenges, IEEE Access, 7, 135812-135831.

Ikhsan, A.F., Rukmana, A., Suarsa, A.C., Masykur, M., Wiharso, T.A. (2019). The prototype of SCADA application in the school building, 4th Annual Applied Science and Engineering Conference.

IFARLAB, (2023). Eskişehir Osmangazi Üniversitesi Akıllı Fabrika ve Robotik Laboratuvarı web sayfası. Erişim tarihi: 20.07.2023.

Ingrand, F., (2019). Recent Trends in Formal Validation and Verification of Autonomous Robots Software, 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 321-328.

IndigoSCADA, (2023). IndigoSCADA Web Sayfası, Erişim: <http://www.enscada.com/>, Erişim tarihi: 20.07.2023.

Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M., (2019). Formal Specification and Verification of Autonomous Robotic Systems: A Survey, ACM Computing Surveys, 52(5)

McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A., Maniatakos, M., Karri, R., (2016). The Cybersecurity Landscape in Industrial Control Systems, in Proceedings of the IEEE, 104(5), 1039-1057, May 2016.

Nazir, S., Patel, S., Patel, D., (2017). Assessing and augmenting SCADA cyber security: A survey of techniques, Computer & Security, 70, 436-454.

Ndukwe, C., Iqbal, M.T., Khan, J., (2022). An Open Source LoRa Based, Low-Cost IoT Platform for Renewable Energy Generation Unit Monitoring and Supervisory Control, Journal of Energy and Power Technology, 4(1).

OPC UA, (2023). OPC UA Web Sayfası, Erişim: <https://opcfoundation.org/about/opc-technologies/opc-ua/>, Erişim tarihi: 20.07.2023.

OpenSCADA, (2023). Open SCADA Projesi Web Sayfası, Erişim: <http://oscada.org/>, Erişim tarihi: 20.07.2023.

Qays, M.O., Ahmed, M.M., Mahmud, M.A.P., Abu-Siada, A., Muyeen, S.M., Hossain, M.L., Yasmin, F., Rahman, M.M., (2022). Monitoring of renewable energy systems by IoT-aided SCADA system, Energy Science & Engineering.

python-opcua, (2023). python-opcua kod deposu linki, Erişim: <https://github.com/FreeOpcUa/python-opcua>, Erişim tarihi: 20.07.2023.

ROS, (2023). ROS Web Sayfası, Erişim: <https://www.ros.org/>, Erişim tarihi: 20.07.2023.

ROS-OPC UA, (2023). ROS – OPC UA Haberleşme paketi web sayfası, Erişim: http://wiki.ros.org/ros_opcua_communication, Erişim tarihi: 20.07.2023.

RapidSCADA, (2023). RapidSCADA Web Sayfası, Erişim: <https://rapidscada.org/>, Erişim tarihi: 20.07.2023.

Sinha, R., Patil, S., Gomes, L., Vyatkin, V., (2019). A Survey of Static Formal Methods for Building Dependable Industrial Automation Systems, IEEE Transactions on Industrial Informatics, 15(7), 3772-3783,

SZARP, (2023). SZARP Proje Web Sayfası, Erişim: <https://gitlab.newterm.pl/Newterm/szarp>, Erişim tarihi: 20.07.2023.

ScadaBR, (2023). ScadaBR Proje Web Sayfası, Erişim: <https://github.com/ScadaBR/ScadaBR>, Erişim tarihi: 20.07.2023.

- S.E.E.R 2, (2023). S.E.E.R 2 Proje Web Sayfası, Erişim: <https://sourceforge.net/projects/seer2/>, Erişim tarihi: 20.07.2023.
- Shitharth, S., Winston, P., (2017). An enhanced optimization based algorithm for intrusion detection in SCADA network, *Computers & Security*, 70, 16-26.
- Webster, M., Dixon, C., Fisher, M., Salem, M., Saunders, J., Koay, K.L., Dautenhahn, K., Saez-Pons, J., (2016). Toward Reliable Autonomous Robotic Assistants Through Formal Verification: A Case Study, *IEEE Transactions on Human-Machine Systems*, 46(2), 186-196.
- Wang, Y., Xu, Z., Zhang, J., Xu, L., Wang, H., Gu, G., (2014). SRID: State relation-based intrusion detection for false data injection attacks in SCADA, *European Symposium on Research in Computer Security*, 401-418.
- Wong, K., Dillabaugh, C., Seddigh, N., Nandy, B., (2017.)Enhancing Suricata intrusion detection system for cyber security in SCADA networks, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE),
- VALU3S, (2023). VALU3S Proje Web Sayfası. Erişim: <https://valu3s.eu/>, Erişim tarihi: 20.07.2023
- Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Im, E. G., Pranggono, B., Wang, H. F., (2014). Multiattribute SCADA-Specific Intrusion Detection System for Power Networks, *IEEE Transactions on Power Delivery*, 29(3), 1092-1102.
- Zheng, X., Julien, C., (2015.)Verification and Validation in Cyber-Physical Systems: Research Challenges and a Way Forward, 2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems, Florence, Italy, 2015, 15-1.