

SCADE Ortamında C Programlama Dili Kullanılarak PLC'lere MODBUS Protokolü Üzerinden Bağlantı Sağlanması ve Temel Anklaşman Sisteminin Gerçekleştirilmesi

Establishing a Connection to PLCs via the MODBUS Protocol Using the C Programming Language in the SCADE Environment and Implementing the Basic Interlocking System

*Dora Demir¹, İbrahim Can Kolotoğlu², Muhammet Işık², Serhat Boynukalın³, Mehmet Turan
Söylemez²*

¹Elektronik ve Haberleşme Mühendisliği Bölümü
İstanbul Teknik Üniversitesi, İstanbul
demirdo20@itu.edu.tr

²Kontrol ve Otomasyon Mühendisliği Bölümü
İstanbul Teknik Üniversitesi, İstanbul
kolotoglu21@itu.edu.tr
isikmuhamm@gmail.com
soylemezm@itu.edu.tr

³Lisansüstü Eğitim Enstitüsü
İstanbul Teknik Üniversitesi, İstanbul
boynukalin@itu.edu.tr

Özetçe

Bu çalışmada, demiryolu sinyalizasyon sistemleri için emniyetli ve esnek bir kontrol altyapısı geliştirilmesi hedeflenmiştir. ANSYS SCADE ile oluşturulan model tabanlı tasarım, C programlama dili ile yazılan MODBUS entegrasyonu sayesinde PLC sistemlerine başarıyla entegre edilmiştir. Sinyalizasyon, makas ve ray kontrolü için tasarlanan İnsan-Makine Arayüzü (HMI), operatörlere sistemin durumunu gerçek zamanlı görme ve müdahale etme olanağı sağlamıştır. Sistem, veri trafiğini optimize eden değişim algılama mekanizması ve güçlü veri depolama altyapısıyla yüksek performans göstermiştir. Gerçek demiryolu maketi üzerinde yapılan testlerde gecikme, senkronizasyon ve kararlılık kriterleri sağlanmıştır. Sonuç olarak, bu proje; MODBUS destekli, esnek, emniyetli ve yeniden kullanılabilir bir sinyalizasyon çözümü sunarak, demiryolu otomasyon sistemlerinde uygulanabilir bir örnek teşkil etmektedir.

Abstract

This study aims to develop a safe and flexible control infrastructure for railway signaling systems. A model-based design created using ANSYS SCADE was successfully integrated with PLC systems through MODBUS communication implemented in the C programming language.

The designed Human-Machine Interface (HMI) enables real-time monitoring and control of signal, switch, and track conditions by operators. The system demonstrated high performance through an efficient change-detection mechanism and comprehensive logging infrastructure. Tests on a physical railway mock-up confirmed stability, low latency, and data synchronization. In conclusion, this project presents a reliable, reusable, and MODBUS-compatible signaling solution, offering a practical model for industrial railway automation applications.

1. Giriş

MODBUS, 1979 yılında Modicon tarafından programlanabilir lojik denetleyiciler (PLC'ler) arasında iletişim kurmak amacıyla geliştirilmiş bir protokoldür. Başlangıçta yalnızca seri iletişim için tasarlanmış olan bu protokol, teknolojik ilerlemelerle birlikte TCP/IP gibi modern ağlarla uyarlanmış ve endüstriyel otomasyon sistemlerinde yaygın bir şekilde kullanılmaya başlanmıştır (Thomas, 2015). MODBUS'ın açık kaynaklı yapısı ve kolay anlaşılabilirliği, farklı üreticiler tarafından geniş bir şekilde kullanılmasına olanak tanımış ve onu endüstride bir standart haline getirmiştir. MODBUS, farklı cihazların birbirleriyle uyumlu bir şekilde çalışmasını sağlamasıyla öne çıkar. Özellikle veri toplama, izleme ve kontrol sistemlerinde protokolün yapısı, sistem entegrasyonunu kolaylaştırmakta ve mühendisler arasında yaygın bir tercih olmasını sağlamaktadır.

(Galloway & Hancke, 2013). Yukarıda verilen nedenlere ek olarak, MODBUS protokolünü mühendisler için erişilebilir kılan sadece açık kaynak kodlu olmasının yanı sıra lisans gerektirmemesi de bu protokolün yaygınlaşmasına önemli ölçüde katkıda bulunmuştur.

Bu çalışmada kullanılmış bir program olan ANSYS SCADE (Safety Critical Application Development Environment); model tabanlı tasarımı, simülasyonu, doğrulanması ve otomatik kod üretimi için havacılık, otomotiv, demiryolu ve enerji gibi emniyet standartlarının yüksek olması gereken sektörlerde kullanılan bir mühendislik tasarım yazılımıdır. Havacılık endüstrisinde DO-178C, otomotivde ISO 26262, ve demiryolu sistemlerinde EN 50128 standartlarına uygun yazılımlar SCADE sayesinde oluşturulabilir. SCADE'in temel kullanım alanları, sistem mühendisliği süreçlerinde tasarım ve doğrulama süreçlerinin iyileştirilmesidir. Yazılım, karmaşık sistemler için görsel bir modelleme yaklaşımı sunar. Bu yaklaşım, tasarım hatalarının erken aşamalarda tespit edilmesini sağlamasından dolayı maliyetli revizyonların da önüne geçilmiş olunur. Buna ek olarak, SCADE ile oluşturulan modeller, simülasyon aracılığıyla test edilebilir ve ardından otomatik olarak yüksek kaliteli, sertifikasyona hazır kaynak koda dönüştürülebilir. Bu özellik sayesinde, manuel kodlamadan kaynaklanabilecek insan hatalarını önemli ölçüde azaltılarak yazılım geliştirme sürecinin hızlandırılması sağlanır (Pereira et al., 2021).

SCADE'in ürettiği C kodları, yüksek doğrulukta ve optimize edilmiş olup, insan müdahalesine gerek kalmadan derlenebilir ve çalıştırılabilir. Bu özellik sayesinde, manuel kodlamadan kaynaklanabilecek hataları azaltır ve yazılım geliştirme sürecinde zamandan da tasarruf sağlar. İlave olarak, üretilen C kodu, emniyet standartları doğrultusunda bağımsız denetim ve sertifikasyon süreçlerinden geçmeye hazır bir yapıdadır (Smith & Johnson, 2021).

2. Yöntem

2.1. Anlaşman Sistemi

Anlaşman sistemi, demiryolu sinyalizasyonunun en önemli unsurlarından biridir ve trenlerin emniyetli bir şekilde hareket etmesini sağlamak amacıyla hareket yetkisinin belirlenmesi ve makas kontrolü gibi kritik süreçleri yönetir. Sistem, sahadaki makasların konumlarını, sinyallerin durumlarını ve ray bloklarının meşgullük bilgilerini merkeze ileterek bu unsurların koordinasyonunu sağlar. Sahadan gelen bu veriler ışığında, güzergâh planlaması yapılır ve elektronik kilitleme mekanizmaları devreye girerek trenlerin emniyeti sağlanmış rotalar üzerinde hareket etmelerini sağlar. Ayrıca, trenlerin rotalarında kalmasını sağlamak ve olası hataları önlemek için otomatik tren kontrol (ATC) ve otomatik tren koruma (ATP) sistemleriyle entegre çalışır. Bu sistemler, demiryolu trafiğinin hem emniyetli hem de verimli bir şekilde yönetilmesinde kritik bir rol oynar. Anlaşman sistemi, makas ve sinyal konumlarının doğru şekilde kontrol edilmesini sağlayarak trenlerin emniyetli güzergâhlar üzerinde hareket etmesine yardımcı olur. Aynı zamanda, ray bloklarının doluluk durumunu izleyerek çarpışma riskini en aza indirir ve trafik akışının düzenli bir şekilde devam etmesini destekler. Sonuç olarak, anlaşman sistemi, demiryolu sinyalizasyonunun temel taşıdır ve emniyet ile etkinliği bir araya getirerek tren operasyonlarının sorunsuz bir şekilde yürütülmesini sağlar. Emniyeti ve doğruluğu sayesinde, demiryolu sistemlerinin

sürdürülebilir ve emniyetli bir altyapı oluşturmasında vazgeçilmez bir araçtır (Kaya & Yılmaz, 2018).

2.2. C libmodbus Kütüphanesi

C libmodbus, açık kaynaklı bir MODBUS iletişim protokolü kütüphanesidir ve C programlama dili kullanılarak yazılmıştır. Bu kütüphane, MODBUS protokolünün RTU (Remote Terminal Unit), ASCII ve TCP/IP gibi farklı varyasyonlarını destekler. Endüstriyel cihazlar arasında veri alışverişini kolaylaştırmak için tasarlanmıştır ve kullanıcıların MODBUS protokolüne uygun bir şekilde cihazlarla haberleşmesini sağlar (Clarke et al., 2019). C programlama dilinde MODBUS protokolünü kullanarak cihazlarla iletişim kurmayı kolaylaştıran bir kütüphane olması sebebiyle bu projede de kullanılmıştır.

2.3. ANSYS SCADE Display ile Ekran Tasarımı

ANSYS SCADE Display, özellikle emniyet kritik gömülü sistemler için İnsan-Makine Arayüzleri (HMI) geliştirmede kullanılan bir yazılımdır. Bu araç, kullanıcıların grafiksel ekran tasarımlarını kolaylıkla oluşturmasına ve bu tasarımları gerçek sistemlerde kullanılabilecek sertifikalandırılabilir koda dönüştürmesine olanak tanır (Smith & Brown, 2020).

SCADE Display'in en büyük avantajlarından biri, tasarımın ilk aşamalarından itibaren hata tespiti ve doğrulama süreçlerini entegre etmesidir. Bu özellik, hem geliştirme sürecini hızlandırır hem de kullanıcıların emniyetli HMI çözümleri üretmesine yardımcı olur. Örneğin, tren kontrol panelleri bu yazılım sayesinde emniyet standartlarına uygun bir şekilde tasarlanabilir ve doğrulanabilir (Pereira et al., 2021).

2.4. Kod Üretimi ve Derlenmesi

SCADE gibi araçlarda oluşturulan grafiksel modeller, otomatik olarak bir programlama dilinde kod haline getirilir. Bu çalışmada C programlama dili esas alınmıştır. Bu kodun MODBUS ile uyumlu çalışabilmesi için gerekli kütüphaneler (örneğin, MODBUS TCP/IP veya RTU desteği sağlayan lib dosyaları) eklenir.

2.5. Anlaşman Tasarımı

Anlaşman tasarımı, saha ekipmanlarının emniyetli bir şekilde kontrol edilmesi ve doğru pozisyonda kilitlenmesi işlevlerini içerir. Bu süreçte, makas motorları, sinyal lambaları ve ray devreleri gibi bileşenler sistemin merkezi tarafından izlenir ve kontrol edilir (Çimen et al., 2020). Anlaşman tasarımı, demiryolu trafiğinin emniyetli ve verimli bir halde yönetilmesi için gereklidir.

3. Model

3.1. Tren Hattı Kontrol Sistemi

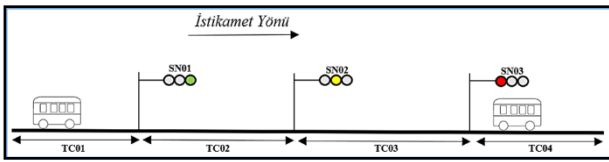
Proje, bir anlaşman sistemi üzerinden raylar, makaslar, sinyaller ve tren hareketlerini kontrol ederek, tren trafiğinin emniyetli bir şekilde yönetilmesini amaçlar. İnsan-Makine Arayüzü (HMI) tasarımı, tren operatörlerine sistemin durumu hakkında gerçek zamanlı bilgi sağlarken, operasyonel kontrolü kolaylaştırır. Projenin nihai hedefi, bir HMI üzerinden basit bir arayüzle tüm süreçleri izlemek ve kontrol etmektir. Örnek olarak Şekil 1 baz alınarak gerekli açıklamalar, ilgili şemanın altında takdim edilmiştir.

- TC01/02/03/04: Ray devresini temsil eder.
- SN01/02/03: Sinyalleri temsil eder.

Şekil 1 incelendiğinde, sinyal sistemi sistemin emniyetinin bozulmaması için soldan gelen ilk aracın öncelikle yavaşlamasını (SN02), daha sonrasında ise durması sinyalini vermiştir (SN03).

Bir trenin hızını emniyetli bir şekilde ayarlayabilmesi için farklı sinyaller aracılığıyla ilerideki koşullar hakkında önceden bilgi alması sağlanır ve birden fazla bloğun durumu önceden görüntülenebilir, bu da tren trafiğinin daha düzenli ve hızlı bir şekilde akmasını sağlar.

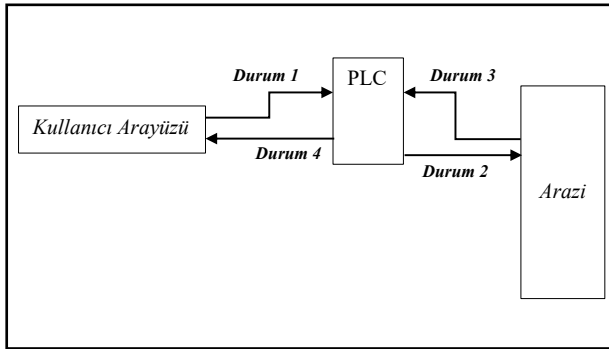
Makinistin tren hızını ayarlayabilmesi için, sinyaller aracılığıyla hat üzerindeki blokların meşguliyet veya müsaitlik durumu bloğa girmeden önceden gösterilir. Böylece, bloklar önceden izlenebilir ve tren trafiğinin daha güvenli, düzenli ve akıcı şekilde yönetilmesi sağlanır.



Şekil 1: İki trenli bir ray yolunun sinyalli gösterimi.

3.2. Sistemin Yapısı

ANSYS SCADE yardımıyla tasarlanmış olan kullanıcı arayüzü; operatörlere trenlerin konumunu, makasların durumunu, sinyalizasyon bilgilerini ve ray bloklarının (ray devreleri) doluluğu hakkında bilgi sağlamasıyla beraber müdahale etmesine de olanak sağlar.



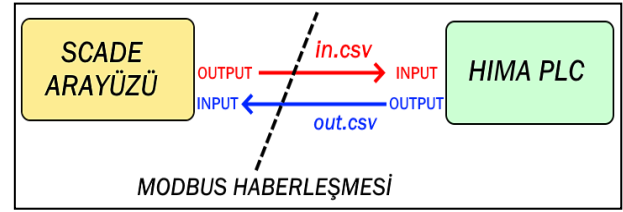
Şekil 2: Haberleşme şematigi.

Durum 1: Operatör, İnsan-Makine Arayüzü (HMI) üzerinden bir işlem talebi başlattığında, bu komut MODBUS protokolü ile PLC'ye iletilir.

Durum 2: PLC, bu komutu ilgili saha cihazına MODBUS komutları ile (örneğin, bir sinyal lambasına) gönderir. Bu şematikte (Şekil 2) komut, "Arazi" kısmına gönderilir.

Durum 3: Cihaz, aldığı komutu uygulayarak durumunu günceller. Sinyal cihazları (örneğin, kırmızı ışık, yeşil ışık gibi sinyal durumları) veya ray devreleri (rayın dolu/boş durumu) gibi bilgileri MODBUS üzerinden PLC'ye geri gönderir.

Durum 4: PLC bu verileri toplar ve HMI'ye iletir. Operatör, durumu gerçek zamanlı olarak görür.



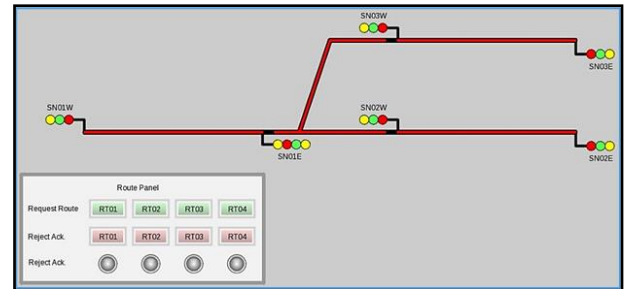
Şekil 3: Veri tabanlı haberleşme şeması örneği.

HMI Entegrasyonu: HMI, operatörlerin sistemi görsel bir arayüz üzerinden gerçek zamanlı olarak izlemelerine ve kontrol etmelerine olanak tanır. HMI, MODBUS üzerinden gelen verileri işleyerek trenlerin konumları, sinyal durumu ve makas pozisyonları gibi bilgileri görselleştirir. Bu, operatörlere daha bilinçli ve hızlı kararlar alma imkânı tanır (Cena et al., 2011).

3.3. Ekran Tasarımı

Projenin gelişim aşamasında ANSYS SCADE Display ile yapılan arayüz (Şekil 4), bir demiryolu sinyalizasyon ve yönlendirme sisteminin emniyetli ve etkin bir şekilde kontrol edilmesini sağlar.

Arayüzün işlevleri incelenmek istenirse: Kullanıcılar, arayüzdeki Route Paneli üzerinden belirli bir tren için rota talep edebilir. Örneğin, RT01 düğmesine basıldığında, trenin SN01W'den SN01E'ye ilerlemesi için bir rota ayarlanabilir. Rota talebi yapıldığında, ilgili sinyal noktaları otomatik olarak güncellenir ve trenin emniyetli bir şekilde hareket etmesi sağlanır.



Şekil 4: SCADE Display ile Oluşturulmuş Örnek Bir Arayüz Tasarımı.

3.4. Hat Kontrol Sistemi Verileri

HMI ve PLC arasındaki veri alışverişinde kullanılan değişkenlerin isimlendirilmesi ve işlenmesi için kullanılan giriş (input) ve çıkış (output) verilerini temsil etmek üzere kullanılan CSV formatındaki veriler aşağıda verilmiştir. Aşağıda verilmiş olan tablolardan Tablo 1, giriş kod verileri için olup; Tablo 2 ise çıkış kod verileri için bir nevi bilgilendirme amacıyla oluşturulmuştur. Bazı veriler için tekrarlayan kısımlar yazılmamış olup, değişen kısımlar yazılarak açıklama yapılmıştır.

değil, kodun içinde hesaplanan yerel değişkenleri de test için erişilebilir hale getirir. (Schlich et al., 2020, s. 62-63).

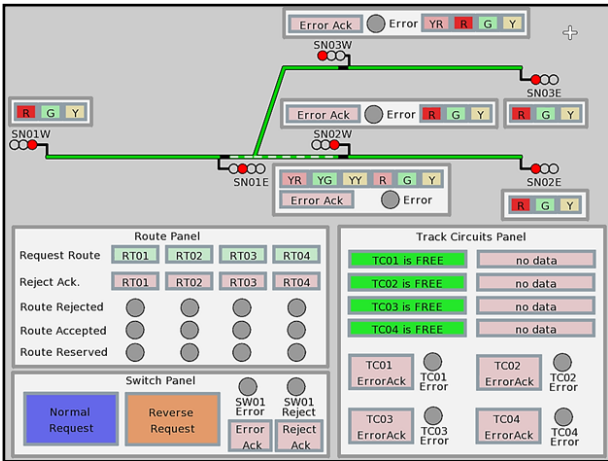
KCG, ürettiği C kodunu kanal işlemleri için makrolar tanımlayarak kolaylaştırır. Kanal, iki süreç arasındaki veri iletimini sağlayan ve giriş ile çıkış verilerini bünyesinde içeren bir yapıdır. KCG'nin makroları, SCADE modellerinden üretilen C kodunun kanallar aracılığıyla nasıl iletişim kurulması gerektiğini tanımlar (Schwarz et al., 2018).

4. Bulgular ve Sonuçlar

4.1. Projenin C Programlama Dili ile Entegrasyonu

Projede, C programlama dili kullanılarak arayüz oluşturulmuş ve PLC sistemlerinin entegrasyonu başarıyla gerçekleştirilmiştir. C dili, düşük seviye donanım kontrolü ve yüksek seviye uygulama geliştirme özellikleri sayesinde sistemin işlem hızını artırmış, üretim maliyetlerini düşürmüş ve programlama kolaylığı sağlamıştır (Bayrakçı & Büyükpatpat, 2021). C programlama dilinde ulaşılabilir bilgi kaynaklarının bol olması, geliştirme sürecinde karşılaşılan sorunların hızlıca çözülmesine yardımcı olmuştur. Dolayısıyla, arayüz ve PLC arasındaki veri alışverişi optimize edilmiş, operatörlerin sistemle etkileşimi daha verimli hale getirilmiştir.

4.2. Arayüzün Kullanım Kolaylığı



Şekil 6: Tasarlanan uygulama arayüzü.

Tablo 1 ve Tablo 2 incelendiğinde, arayüzdeki ifadelerin ne anlama geldiği anlaşılabilir. Arayüz, operatörün zorlanmayacağı şekilde tasarlanmıştır. Ekranın ortasından itibaren üst kısımda ray devresinin simülasyonu varken, alt kısmında ise kontrol panelleri bulunmaktadır. Operatör, gerçekleştirmek istediği işlemleri paneller sayesinde kolaylıkla yapabilir.

Arayüz ilk açıldığında (Şekil 6), tüm sinyallerin varsayılan olarak kırmızı renkte olduğu ve makasların en son bırakıldığı konumda (bu örnekte normal konumda) bulunduğu görülmektedir.

4.3. MODBUS Entegrasyonu

Proje kapsamında gerçekleştirilen MODBUS entegrasyonu, sistemin farklı endüstriyel cihazlar arasında emniyetli ve standartlara uygun bir iletişim ağı kurmasını sağlamıştır. MODBUS protokolü ile yapılan veri alışverişi, gecikme sürelerinin düşük seviyelerde tutulduğunu ve veri kaybının minimize edildiğini ortaya koymuştur. Bu, endüstriyel otomasyon süreçlerinde emniyetli bir kontrol mekanizması oluşturulması açısından önemli bir kazanım olmuştur. Ayrıca, protokolün esnek yapısı, farklı cihazlarla entegrasyonun kolaylıkla gerçekleştirilmesine olanak tanımıştır.

4.4. Konfigürasyon

Yapılandırma sistemi, INI dosya formatını kullanır. Bu sistem, ModbusConfig, InputMappings ve OutputMappings olmak üzere üç temel bölüme ayrılmıştır. ModbusConfig bölümü, sistemin bağlantı parametrelerini içerirken; InputMappings ve OutputMappings bölümü, sırasıyla giriş ve çıkış değişkenlerini MODBUS adreslerine eşler.

InputMappings ve OutputMappings bölümleri, SCADE değişkenleri ile MODBUS kayıt adresleri arasındaki eşleştirmeleri tanımlar. Her bir eşleştirme, değişken adı ve ilgili MODBUS adresini içerir. Örneğin, "out_RT01_Accept=1" tanımı, ray durumu kabul sinyalinin birinci MODBUS adresine atandığını gösterir. Bu yapılandırma sistemi çeşitli özellikler sayesinde sistemin emniyetini artırır. Eksik veya bozuk yapılandırma dosyalarını algılar ve varsayılan ayarları otomatik olarak oluşturur. Bu, sistemin minimum kesintiyle çalışmasını sağlar.

4.5. Test ve doğrulama süreci

Sistem, SILWORKS ile programlanan HIMA PLC'ler kullanılarak gerçek bir demiryolu maketi üzerinde test edilmiştir. Test sürecinde bağlantı kararlılığı, veri senkronizasyon doğruluğu ve sistem yanıt süreleri detaylı şekilde incelenmiştir. Performans açısından sistem, standart MODBUS TCP/IP protokolünün sunduğu iletişim hızlarında çalışmaktadır. Veri güncelleme döngüsü 1 saniye olarak ayarlanmış olup, bu süre demiryolu sinyalizasyon sistemlerinin gereksinimleri için yeterli bulunmuştur.

4.6. Optimizasyon

Sistemdeki performans optimizasyonu, birden fazla seviyede gerçekleştirilir. İletişim seviyesinde, sistem, ağ yükünü azaltan toplu işlemleri (batch operations) uygular. Değişim algılama mekanizması, yalnızca değiştirilmiş değerleri ileterek gereksiz ağ trafiğini önler. Bağlantı havuzu (connection pooling), sistemin duyarlılığını korurken kaynakların verimli bir şekilde kullanılmasını sağlar. Bellek optimizasyonu, kaynakların verimli kullanımına odaklanırken sistem emniyetini de korur. Sistem, bellek kullanımını performans gereksinimleriyle dengeleyen dikkatlice tasarlanmış tampon boyutlandırması uygular. Ayrıca, kaynak temizleme prosedürleri, sistemin uzun süreli çalışma dönemlerinde bellek kullanımının stabil kalmasını sağlar.

4.7. Kodun çalışma akışı

Kod, yapılandırma dosyasını okuyarak sistemin parametrelerini yükler. Eğer yapılandırma dosyası bulunamazsa, sistem

varsayılan olarak bilgisayarın yerel adresine bağlantı sağlar ve bu bağlantının yapılandırmasını bir dosya halinde kaydeder. Daha sonra programda tanımlı değişkenlerin adreslerinin tanımlanıp tanımlanmadığını iteratif olarak kontrol eder. Tanımlanan adresler, ilgili yapı içerisinde MODBUS adresleriyle eşleştirilir.

Bağlantı, yapılandırma dosyasından okunan parametrelerle MODBUS bağlantı başlatma fonksiyonu aracılığıyla başlatılır. Ardından, giriş (input) ve çıkış (output) listeleri PLC'den toplu olarak okunur. Bu okuma işlemi, sistem performansını artırmak amacıyla belirli zaman aralıklarında tekrarlanır. Bu süreçte, değişkenlerin değer değişiklikleri sürekli olarak izlenir ve yalnızca değişen parametreler tespit edildiğinde, tek tek karşı tarafa yazılır.

Kullanıcı arayüzünde birden fazla tuşa aynı anda basılması mümkün olmadığı için, performansı optimize etmek amacıyla parametreler sırayla işlenir. Arayüz programı, her bir çizim sırasında (işlem gücüne bağlı olarak saniyede birkaç bin defa) tüm değişkenleri içeren bir struct verisini MODBUS yazma fonksiyonuna gönderir. Bu yazma fonksiyonu, sırasıyla okuma fonksiyonunu, bağlantı kurma fonksiyonunu ve adres ile değişken eşleştirme fonksiyonunu çağırarak mevcut durumu sürekli kontrol eder. Herhangi bir bağlantı kopması durumunda, iteratif yeniden deneme mekanizmalarını devreye sokarak iletişimin kesintisiz bir şekilde sürdürülmesini sağlar.

Sistem, hata ve çalışma durumunun sürekli izlenebilmesi için tüm işlemleri ayrıntılı bir şekilde kayıt altına alır. Bu işlemler, veri depolama fonksiyonu aracılığıyla programın çalışma dizininde bir log dosyasına kaydedilir. Bu mekanizma, sistemin performansını ve kararlılığını artırırken, olası sorunların tespit edilmesi ve çözülmesi için gerekli verileri sağlar.

4.8. Sonuçlar

SCADE modeli ile oluşturulan yazılımın, elle yazılan MODBUS bağlantı kodlarıyla entegre edilmesi, sistemin esnekliğini ve yeniden kullanılabilirliğini artırmıştır. Bu entegrasyon sırasında herhangi bir veri kaybı yaşanmamış ve bağlantı stabil bir şekilde çalışmıştır. Giriş ve çıkış sinyallerinin değişim algılama mekanizması sayesinde gereksiz ağ trafiği önlenmiş, yalnızca değişen verilerin iletimi sağlanmıştır. Bu yaklaşım, ağ yükünü azaltarak iletişim hızını artırmış ve işlem sürelerini optimize etmiştir. Sistem, operatörlerin hata tespiti ve düzeltme işlemlerini kolaylaştıran kapsamlı veri depolama özellikleri ile desteklenmiştir. Ayrıca, Sistemde kullanılan adresler, IP'ler ve portlar dinamik olarak değiştirilip yapılandırılabilir hale getirilmiştir. Ancak, C dilinin yapısal özellikleri ve programın ana mantığı gereği, değişkenlerin dinamik olarak aktarılması hem gereksiz bir işlem olarak değerlendirilmiş hem de dilin doğal olarak desteklemediği bir durum olduğundan bu yaklaşım tercih edilmemiştir.

SCADE Display ile tasarlanan kullanıcı arayüzü (HMI), operatörlerin sistemi daha kolay kontrol etmesine ve izleme yapmasına olanak tanımıştır. Bu arayüz, giriş ve çıkış sinyallerinin görselleştirilmesini sağlayarak kullanıcı deneyimini iyileştirmiştir. Özellikle sinyal durumlarının gerçek zamanlı gösterimi ve kontrol mekanizmalarının anlık geri bildirimleri, sistemin kullanıcı dostu yapısını güçlendirmiştir. Kodda, HMI üzerinden kullanıcıların sistem durumlarını gerçek zamanlı olarak görmelerini ve kontrol etmelerini sağlayan bir mekanizma bulunur. Uygulama arayüzü, sinyal

durumlarını, hata bildirimlerini ve sistem parametrelerini kullanıcıya görsel olarak sunar.

Kaynakça

- [1] Schlich, B., Sinha, S., & Peleska, J. (2020). Asserting functional equivalence between C code and SCADE models in code-to-model transformations. *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 1-10. <https://doi.org/10.1145/3425174.3425213>
- [2] Schwarz, S., Farail, P., & Fayolle, S. (2018). Developing portable reusable apps with SCADE, the FACE Technical Standard, and ARINC 661. *ERTS 2018 Proceedings*, 1-8.
- [3] Bayrakçı, H. C., & Büyükpapat, H. (2021). PLC ve SCADA kontrol yöntemleri ile sıvı dolum otomasyonu. *Avrupa Bilim ve Teknoloji Dergisi*, (27), 283-291. <https://doi.org/10.31590/ejosat.877004>
- [4] Cena, G., Bertolotti, I. C., Scanzio, S., Valenzano, A., & Zunino, C. (2011). Evaluation of EtherCAT distributed clock performance. *IEEE Transactions on Industrial Informatics*, 8(1), 20-29. <https://doi.org/10.1109/TII.2011.2176750>
- [5] Galloway, B., & Hancke, G. P. (2013). Introduction to industrial control networks. *IEEE Communications Surveys & Tutorials*, 15(2), 860-880. <https://doi.org/10.1109/SURV.2012.071312.00124>
- [6] Pereira, F., Smith, J., & Brown, A. (2021). Model-based development of safety-critical systems: A comparative analysis of tools. *Journal of Systems Engineering*, 28(3), 15-30. <https://doi.org/10.1016/j.jse.2021.04.002>
- [7] Smith, R., & Johnson, L. (2021). Advancing safety-critical software development with model-based design. *Journal of Embedded Systems Engineering*, 9(4), 45-56. <https://doi.org/10.1016/j.jese.2021.07.004>
- [8] Thomas, G. (2015). *Introduction to the Modbus Protocol*. Industrial Automation Press.
- [9] Kaya, M., & Yılmaz, H. (2018). Demiryolu Sinyalizasyon Sistemlerinin Güvenilirlik Analizi. *Elektrik, Elektronik ve Bilgisayar Mühendisliği Dergisi*, 27(3), 45-56. <https://dergipark.org.tr/tr/download/article-file/516711>
- [10] Smith, R., & Brown, L. (2020). Leveraging model-based design in critical systems development. *Embedded Systems Journal*, 12(4), 45-60. <https://doi.org/10.1016/j.esj.2020.07.003>
- [11] Çimen, A., Akbaş, A., & Karagöz, T. (2020). Petri ağları ile demiryolu anlaşıman ve sinyalizasyon tasarımı. *Journal of Railway Engineering*, 15(2), 45-60. <https://doi.org/10.1016/j.jre.2020.07.003>
- [12] Clarke, J., Smith, P., & Brown, T. (2019). Industrial communication with open-source libraries: A focus on Modbus and libmodbus. *Journal of Industrial Informatics*, 34(2), 78-90. <https://doi.org/10.1016/j.jii.2019.04.003>
- [13] Kurian, E., Briola, D., Braione, P., & Denaro, G. (2022). Automatically Generating Test Cases for Safety-Critical Software via Symbolic Execution (Version 1). *arXiv*. <https://doi.org/10.48550/ARXIV.2209.11138>