**PLC VE PYTHON TABANLI GÖRÜNTÜ İŞLEME İLE ÜRÜN AYRIŞTIRMA**

**2023**

**LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

**Ahmet Erdem KENET**

**Ayşeana KOCATÜRK**

**Halil Cem AYDIN**

**Muhammet Çağrı ÖZBEK**

**PLC VE PYTHON TABANLI GÖRÜNTÜ İŞLEME İLE ÜRÜN AYRIŞTIRMA**

**Ahmet Erdem KENET**

**Ayşeana KOCATÜRK**

**Halil Cem AYDIN**

**Muhammet Çağrı ÖZBEK**

**Karabük Üniversitesi**

**Mühendislik Fakültesi**

**Elektrik-Elektronik Mühendisliği Bölümünde**

**Lisans Tezi**

**Olarak Hazırlanmıştır**

**KARABÜK**

**Haziran 2023**

Ahmet Erdem KENET, Ayşeana KOCATÜRK, Halil Cem AYDIN, Muhammet Çağrı ÖZBEK, tarafından hazırlanan “PLC VE PYTHON TABANLI GÖRÜNTÜ İŞLEME İLE ÜRÜN AYRIŞTIRMA” başlıklı bu tezin Lisans Tezi olarak uygun olduğunu onaylarım.

16/06/2023

Dr. Öğretim Üyesi Hüseyin ALTINKAYA ..…….………...

Tez Danışmanı,

Karabük Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü

..…….………...

Karabük Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü

..…….………...

Karabük Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü

# ÖZET

**Lisans Tezi**

**PLC VE PYTHON TABANLI GÖRÜNTÜ İŞLEME İLE ÜRÜN AYRIŞTIRMA**

**Ahmet Erdem KENET**

**Ayşeana KOCATÜRK**

**Halil Cem AYDIN**

**Muhammet Çağrı ÖZBEK**

**Karabük Üniversitesi**

**Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü**

**Tez Danışmanı**

**Dr. Öğretim Üyesi Hüseyin ALTINKAYA**

**Haziran 2023, 71 sayfa**

Bu çalışmada, konveyör bant ile PLC (Programmable Logic Controller) kullanılarak içerisinde görüntü işleme dahilinde ürünler sensörler yardımıyla boyut, renk ve cins ayrımı olacak şekilde takip ve ayrıştırılması amaçlanmaktadır. Projenin altyapısında konveyör bant hattı bulunmaktadır. Konveyör bant sistemleri; genellikle katı ürünlerin emniyetli, istenilen hızda ve belirli bir mesafeye insan gücü gerektirmeden kolaylıkla taşınmasında önem taşıyan sistemlerdir. Bu işlemler sırasında PLC sistemi kullanılarak ürünlerin endüstriyel yazılım ile insan hatası olmadan, gerçek start ve stop butonları eklenerek sistemin kontrolü ve takibi yapılmaktadır. Bahsi geçen işlemlerin gerçekleşebilmesi için öncelikle sahada bulunan sensörler üretecekleri sinyalleri, PLC sisteminin dijital giriş modülüne göndererek sistemin çalışmasını sağlarlar. Konveyör bant üzerindeki ürünlerin renk ayrımı görüntü işleme kullanılarak yapılmaktadır. Görüntü işleme; görüntüyü dijital veri formuna çevirmek ve birtakım prosedürlerden geçerek bu işlemden istenilen bilgileri elde etmek için kullanılan metottur. Kullanılan görüntü işleme modeli YOLO v5’tir. Projede amaçlanan, konveyör üzerine koyulan cisimlerin hareketi 12V DC motor ile sağlanmaktadır. Cisimlerin, sensörlerin algıladığı mesafeden geçerek PLC kontrolü ile, belirlenen alanlara; rengine, cinsine ve boyutuna göre yönlendirilerek ayrıştırılmasıdır. İstenilen yerlere step motorun hareketi yardımıyla belirlenen derecelere göre yön verilerek taşınması amaçlanmıştır.

**Anahtar Kelimeler :** Konveyör, PLC, Görüntü İşleme, Sensör, Step Motor.

# ABSTRACT

**B. Sc. Thesis**

**PRODUCT SORTING WITH PLC AND PYTHON BASED IMAGE PROCESSING**

**Ahmet Erdem KENET**

**Ayşeana KOCATÜRK**

**Halil Cem AYDIN**

**Muhammet Çağrı ÖZBEK**

**Karabük University**

**Faculty of Engineering Department of Electrical-Electronics Engineering**

**Thesis Advisor**

**Assist. Prof. Dr. Hüseyin ALTINKAYA**

**June 2023, 71 pages**

In this study, it is aimed to track and separate the products using a Programmable Logic Controller (PLC) with a conveyor belt to distinguish size, colour, and type with the help of sensors within image processing. There is a conveyor belt line in the infrastructure of the project. Conveyor belt systems are generally systems that are important for transporting solid products safely, at the desired speed and easily to a certain distance without requiring human power. During these operations, the control and monitoring of the system is carried out by adding real start and stop buttons without human error with the industrial software of the products using the PLC system. For the operations to take place, first, the sensors located on the site send the signals they will generate to the digital input module of the PLC system to ensure the operation of the system. The colour separation of the products on the conveyor belt is carried out using image processing. Image processing: it is the method used to convert the image into a digital data form and to obtain the desired information from this process by going through several procedures. The image processing model used is YOLO v5. The movement of the objects placed on the conveyor, which is intended in the project, is provided by a 12V DC motor. It is the separation of objects by passing through the distance detected by sensors and directing them to the designated areas according to their colour, type, and size by PLC control. It is intended to be transported to the desired places by giving directions according to the degrees determined with the help of the movement of the stepper motor.

**Key Words :** Conveyor, PLC, Image Processing, Sensors, Stepper Motor.

# TEŞEKKÜR

Hazırlamış olduğumuz tez çalışmamızda bize yol gösteren ve bu yolda bize ışık tutan, emeklerini esirgemeyen öğrencisi olmaktan her zaman gurur duyduğumuz tez danışanımız sayın Dr. Öğretim Üyesi Hüseyin ALTINKAYA’ya teşekkürlerimizi sunarız.

Manevi destekleriyle bizi hiçbir zaman yalnız bırakmayan çok değerli ailelerimize minnet duyarız.

Gerçekleştirmiş olduğumuz projemizde değerli içerikleri ile sürekli faydalandığımız ve sorularımıza cevap bulduğumuz “S7-1200 PLC Programlama” kitabının yazarı Yavuz EMİNOĞLU’na teşekkür ederiz.

*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Ahmet Erdem KENET

Ayşeana KOCATÜRK

Halil Cem AYDIN

Muhammet Çağrı ÖZBEK

# İÇİNDEKİLER

**Sayfa**

[ÖZET iii](#_Toc137915952)

[ABSTRACT iv](#_Toc137915953)

[TEŞEKKÜR v](#_Toc137915954)

[İÇİNDEKİLER vi](#_Toc137915955)

[ŞEKİLLERİN LİSTESİ x](#_Toc137915956)

[BÖLÜM 1 1](#_Toc137915957)

[1. GİRİŞ 1](#_Toc137915958)

[BÖLÜM 2 2](#_Toc137915959)

[2. PLC’NİN YAPISI VE ÇALIŞMA İLKESİ 2](#_Toc137915960)

[2.1. PLC Tarihçesi 2](#_Toc137915961)

[2.2. PLC Nedir? 2](#_Toc137915962)

[2.2.1. PLC’nin Sağladığı Avantajlar 3](#_Toc137915963)

[2.2.2. PLC’nin Sağladığı Dezavantajlar 3](#_Toc137915964)

[2.3. PLC’nin Elemanları 4](#_Toc137915965)

[2.3.1. Merkezi İşletim Birimi (CPU) 4](#_Toc137915966)

[2.3.2. Sinyal Board 4](#_Toc137915967)

[2.3.3. Sinyal Modülleri 5](#_Toc137915968)

[2.3.4. Haberleşme Modülleri 5](#_Toc137915969)

[2.4. PLC Giriş Çıkış Birimleri 5](#_Toc137915970)

[2.4.1. Giriş Birimi 6](#_Toc137915971)

[2.4.2. Çıkış Birimi 6](#_Toc137915972)

[2.4.3. Giriş-Çıkış Birimlerine Bağlanabilecek Bazı Algılayıcılar; 7](#_Toc137915973)

[2.4.4. PLC’nin Çıkış Sinyallerine Göre Çalışan Bazı Dış Ortam İşlemcileri; 7](#_Toc137915974)

[2.5. PLC’nin Çalışma Fonksiyonu 7](#_Toc137915975)

[2.6. Program İşleme Şekilleri 8](#_Toc137915976)

[2.6.1. Lineer Program İşleme 8](#_Toc137915977)

[2.6.2. Yapısal Program İşleme 8](#_Toc137915978)

[2.7. PLC Modelleri 8](#_Toc137915979)

[BÖLÜM 3 9](#_Toc137915980)

[3. PROJE ELEMANLARI 9](#_Toc137915981)

[3.1. Güç Kaynağı Nedir? 9](#_Toc137915982)

[3.1.1. Güç Kaynağı Nasıl Çalışır? 9](#_Toc137915983)

[3.1.2. Valx VMA-1210 Güç Kaynağı 10](#_Toc137915984)

[3.1.3. Valx VMA-2410 Güç Kaynağı 10](#_Toc137915985)

[3.2. Konveyör Bant 11](#_Toc137915986)

[3.2.1. PLC ile Konveyör Çalışması 11](#_Toc137915987)

[3.3. Sensörler ve PLC Bağlantıları 12](#_Toc137915988)

[3.3.1. MZ80 Kızılötesi Mesafe Sensörü (E18-D80NK) Tanımı ve Bağlantısı 12](#_Toc137915989)

[3.3.2. Endüktif Metal Sensörü (LJ18A3-8-Z/BY) Tanımı ve Bağlantısı 13](#_Toc137915990)

[3.4. Sensörlerde PNP ve NPN Tiplerinin PLC Bağlantısı 15](#_Toc137915991)

[3.4.1. PNP Giriş Bağlantısı 15](#_Toc137915992)

[3.4.2. NPN Giriş Bağlantısı 15](#_Toc137915993)

[3.4.3. Endüktif Metal Sensörünün TIA Portal Bağlantısı 16](#_Toc137915994)

[3.4.4. MZ80 Kızılötesi Mesafe Sensörü TIA Portal Bağlantısı 16](#_Toc137915995)

[3.5. PLC ve 12 V DC Motor Haberleşmesi 17](#_Toc137915996)

[3.6. Röle Nedir? 18](#_Toc137915997)

[3.6.1. PNP ve NPN Türlerinin PLC de Birlikte Kullanılması 19](#_Toc137915998)

[3.7. Step Motor Nedir? 19](#_Toc137915999)

[3.7.1. Step Motorun Özellikleri 20](#_Toc137916000)

[3.7.2. Step Motor Çeşitleri 20](#_Toc137916001)

[3.7.3. Adım Sayısı 21](#_Toc137916002)

[3.7.4. Unipolar ve Bipolar Sürücüler 21](#_Toc137916003)

[3.7.5. Dijital Step Motor Sürücüsü (JKD556) 21](#_Toc137916004)

[3.7.6. Avantajları Nelerdir? 22](#_Toc137916005)

[3.7.7. Uygulama Alanları 22](#_Toc137916006)

[3.7.8. Step Motor-Driver Mekanik Bağlantısı 23](#_Toc137916007)

[3.7.9. Step Motor PLC Haberleşmesi 24](#_Toc137916008)

[3.7.10. Step Motor PLC Bağlantısı 24](#_Toc137916009)

[3.7.11. MC\_Power\_DB Ayarları 25](#_Toc137916010)

[3.7.12. MC\_Home\_DB Ayarları 26](#_Toc137916011)

[3.7.13. MC\_MoveAbsolute\_DB Ayarları 27](#_Toc137916012)

[3.8. Step Motor TIA Portal Bağlantısı 28](#_Toc137916013)

[3.8.1. Boyut Ayrıştırma 28](#_Toc137916014)

[3.8.2. Metal Ayrıştırma 29](#_Toc137916015)

[3.8.3. Görüntü İşleme ile Ayrıştırma 30](#_Toc137916016)

[3.9. Aparatlar 31](#_Toc137916017)

[3.9.1. Ürün Hizalama Aparatı 31](#_Toc137916018)

[3.9.2. Step Motora Bağlı Ürün Yönlendirme Aparatı 32](#_Toc137916019)

[3.9.3. Konveyöre LED Bağlanması 32](#_Toc137916020)

[BÖLÜM 4 33](#_Toc137916021)

[4. GÖRÜNTÜ İŞLEME 33](#_Toc137916022)

[4.1. Derin Öğrenme Nedir? 34](#_Toc137916023)

[4.2. Denetimli Öğrenme 35](#_Toc137916024)

[4.3. Denetimsiz Öğrenme 35](#_Toc137916025)

[4.4. Takviyeli Öğrenme 35](#_Toc137916026)

[4.5. YOLO v5 Nedir? 35](#_Toc137916027)

[4.6. Python ile Derin Öğrenme Uygulaması 36](#_Toc137916028)

[4.7. Görüntü İşleme TIA Portal Bağlantısı 41](#_Toc137916029)

[4.8. KepServerEX Kullanarak PLC Haberleşmesi 42](#_Toc137916030)

[BÖLÜM 5 46](#_Toc137916031)

[5. SİSTEMİN PANOLANMASI VE KABLO DÜZENİ 46](#_Toc137916032)

[5.1. Butonlarla Sistemin Kontrolü 47](#_Toc137916033)

[5.2. Ürünlerin Ayrıldığı Bölme 47](#_Toc137916034)

[BÖLÜM 6 48](#_Toc137916035)

[6. SONUÇ VE ÖNERİLER 48](#_Toc137916036)

[KAYNAKLAR 49](#_Toc137916037)

[7. Kaynakça 49](#_Toc137916038)

[EKLER 50](#_Toc137916039)

[models 50](#_Toc137916040)

[Config.py 50](#_Toc137916041)

[DetectionMain.py 50](#_Toc137916042)

[PLCUtils.py 52](#_Toc137916043)

[ProductDetector.py 53](#_Toc137916044)

[requirements.txt 55](#_Toc137916045)

[ÖZGEÇMİŞ 57](#_Toc137916046)

# ŞEKİLLERİN LİSTESİ

|  |  |
| --- | --- |
| **Şekil** | **Sayfa** |

[Şekil 2.1. S7-1200 Siemens Marka PLC 3](#_Toc137916047)

[Şekil 2.2. Haberleşme Modülleri [1] 5](#_Toc137916048)

[Şekil 2.3. PLC Yapısı [2] 6](#_Toc137916049)

[Şekil 2.4. PLC Basit Çalışma Döngüsü [2] 8](#_Toc137916050)

[Şekil 3.1. 12V-10A Güç Kaynağı [3] 10](#_Toc137916051)

[Şekil 3.2. 24V-10A Güç Kaynağı [4] 10](#_Toc137916052)

[Şekil 3.3. Konveyör Bant 11](#_Toc137916053)

[Şekil 3.4. Konveyör Bant Ladder Devre Şeması 12](#_Toc137916054)

[Şekil 3.5. MZ80 Kızılötesi Mesafe Sensörü 12](#_Toc137916055)

[Şekil 3.6. Sensör Muhafazasının Tasarımı 13](#_Toc137916056)

[Şekil 3.7. Büyük ve Küçük Ürünler 13](#_Toc137916057)

[Şekil 3.8. Endüktif Metal Sensörü 14](#_Toc137916058)

[Şekil 3.9. Algılanacak Materyaller (Metal-Metal Olmayan) 14](#_Toc137916059)

[Şekil 3.10. PLC Üzerinde PNP Bağlantı Şeması [5] 15](#_Toc137916060)

[Şekil 3.11. PLC Üzerinde NPN Bağlantı Şeması [5] 15](#_Toc137916061)

[Şekil 3.12. Endüktif Metal Sensörü TIA Portal Bağlantısı 16](#_Toc137916062)

[Şekil 3.13. MZ80 Kızılötesi Mesafe Sensörü TIA Portal Bağlantısı 17](#_Toc137916063)

[Şekil 3.14. Motor, Röle, PLC ve Güç Kaynakları Bağlantı Şeması 17](#_Toc137916064)

[Şekil 3.15. Motor, Röle, PLC ve Güç Kaynakları Bağlantısı 18](#_Toc137916065)

[Şekil 3.16. Röle Yandan Görünüm 18](#_Toc137916066)

[Şekil 3.17. Röle Üstten Görünüm ve Kontak Numaraları 19](#_Toc137916067)

[Şekil 3.18. PNP, NPN Dönüştürme Röle Bağlantısı 19](#_Toc137916068)

[Şekil 3.19. Nema 17 Step Motor 20](#_Toc137916069)

[Şekil 3.20. Dijital Step Motor Sürücüsü Pin Bağlantıları 22](#_Toc137916070)

[Şekil 3.21. Dijital Step Motor Sürücüsü ve Step Motor Pin Bağlantıları 23](#_Toc137916071)

[Şekil 3.22. Obje Seçimi 24](#_Toc137916072)

[Şekil 3.23. Genel Ayarlar 24](#_Toc137916073)

[Şekil 3.24. Mekanik Ayarlar 25](#_Toc137916074)

[Şekil 3.25. TIA Portal Step Motor Ayarları 25](#_Toc137916075)

[Şekil 3.26. MC\_Power Ladder Devresi 26](#_Toc137916076)

[Şekil 3.27. MC\_Home Ladder Devresi 26](#_Toc137916077)

[Şekil 3.28. MC\_MoveAbsolute Ladder Devresi 27](#_Toc137916078)

[Şekil 3.29. Boyut Ayrıştırma Step Motor Blok Diyagramları 28](#_Toc137916079)

[Şekil 3.30. Küçük Ürün 28](#_Toc137916080)

[Şekil 3.31. Büyük Ürün 28](#_Toc137916081)

[Şekil 3.32. Metal Ayrıştırma Step Motor Blok Diyagramları 29](#_Toc137916082)

[Şekil 3.33. Metal Ürün 29](#_Toc137916083)

[Şekil 3.34. Görüntü İşleme ile Ayrıştırma Step Motor Blok Diyagramları 30](#_Toc137916084)

[Şekil 3.35. Kırmızı Ürün 31](#_Toc137916085)

[Şekil 3.36. Mavi Ürün 31](#_Toc137916086)

[Şekil 3.37. Yeşil Ürün 31](#_Toc137916087)

[Şekil 3.38. Ürün Hizalama Aparatı 31](#_Toc137916088)

[Şekil 3.39. Step Motora Bağlı Ürün Yönlendirme Aparatı 32](#_Toc137916089)

[Şekil 3.40. Şerit LED Bağlantısı 32](#_Toc137916090)

[Şekil 4.1. Roboflow [6] Fotoğraf Yüklenmesi 36](#_Toc137916091)

[Şekil 4.2. Fotoğraf Etiketleme 37](#_Toc137916092)

[Şekil 4.3. Fotoğrafların Eğitimi 37](#_Toc137916093)

[Şekil 4.4. Çözünürlük Ayarlama 37](#_Toc137916094)

[Şekil 4.5. Resim Farklılaştırma İşlemi 38](#_Toc137916095)

[Şekil 4.6. Resim Çoğaltma İşlemi 38](#_Toc137916096)

[Şekil 4.7. Python İndirme Kodu Oluşturma 39](#_Toc137916097)

[Şekil 4.8. YOLOv5 Dosyaları Çekilmesi 39](#_Toc137916098)

[Şekil 4.9. API key Çalıştırılarak Roboflow [6] Veriseti Çekilmesi 40](#_Toc137916099)

[Şekil 4.10. Eğitim 40](#_Toc137916100)

[Şekil 4.11. Görüntü İşleme TIA Portal Blokları 41](#_Toc137916101)

[Şekil 4.12. 3 Farklı Renkte Ürünler 41](#_Toc137916102)

[Şekil 4.13. PLC OPC UA Server Kurulması 42](#_Toc137916103)

[Şekil 4.14. KepServerEX Yeni Kanal Oluşturulması 42](#_Toc137916104)

[Şekil 4.15. Endpoint URL Ekleme 43](#_Toc137916105)

[Şekil 4.16. PLC den Gelen Verilerin Import Edilmesi 43](#_Toc137916106)

[Şekil 4.17. OPC UA Server Interface 44](#_Toc137916107)

[Şekil 4.18. Server’dan Import Edilen Veriler 44](#_Toc137916108)

[Şekil 4.19. Server Interface Üzerindeki Verilerin Görüntülenmesi 45](#_Toc137916109)

[Şekil 4.20. OPC Quick Client 45](#_Toc137916110)

[Şekil 5.1. Sistem Panosu 46](#_Toc137916111)

[Şekil 5.2. Genel Sistem Görünümü 46](#_Toc137916112)

[Şekil 5.3. Fiziksel Butonlar 47](#_Toc137916113)

[Şekil 5.4. Ürünlerin Ayrıldığı Bölme 47](#_Toc137916114)

BÖLÜM 1

# GİRİŞ

Konveyör bant üzerinden ürün ayrıştırma projeleri farklı özellikteki cisimleri insan gücü olmadan daha pratik ve seri bir şekilde ayrıştıran önemli otomasyon sistemlerinden biridir. Projenin tasarım aşamasında farklı üç grubun ayrıştırılması beklenmektedir. Bunlar fabrika sistemlerinde en çok ihtiyaç duyulan ayrıştırma şekillerinden seçilmiştir. Boyut ayrıştırma sistemi ile birçok kargo şirketine kolaylık sağlanacağı düşünülmüştür. Metal ayrıştırma ile geri dönüşüm bantlarında metal cisimlerin ayrıştırılıp geri dönüştürülmesinde kolaylık sağlanacağı ve renk ayrıştırma ile görüntü işleme kullanılarak çok daha detaylı şekilde istenen nesneyi tanıyan algılayan ve ona göre PLC ye sinyal gönderen bir sistem ile gelişen teknolojiyle uyumlu insanların hayatını kolaylaştıracak bir sistem düşünülmüştür. Hedeflenen ayrıştırmaların belirlenen bölmelere düşmesi ve bunun için bir motor kullanılması düşünülmüştür. Bunların yanı sıra çok karmaşık olan görüntü işleme ve sistemin PLC ile haberleşmesi için birçok yöntem araştırılmış ve proje için en uygun olanının deneyerek tespit edilmeye çalışılacaktır. Konveyör bant’ın hareketi için ise sistemin stabil şekilde ilerlemesini sağlayacak bir DC motor düşünülmüştür. Uzun bir bant kullanılarak PLC den gelen sinyal ile motorun cisimlere vereceği yön için yeterli mesafe olmasına dikkat edilmiştir. Bilindiği üzere bu işlemler birkaç saniye sürmektedir ve motorun hareketi ve aynı zamanda yeni bir cisim gönderilme ihtimaline karşı tekrar eski konuma gelmesi önemlidir. Ayrıştırmayı düşünülen cisimler için uygun sensörler araştırılmıştır. Sensörlerin hangi cismi neye göre algılayacağı ve konumları tasarlanırken sistemi komplike olmaktan kurtarıp en sade ve anlaşılır bir senaryo ile tasarlanması düşünülmektedir. Sensörlerin kaç volt aralığında çalıştığı projede kullanılacak güç kaynakları için önemlidir ve bu sebeple PLC için kullanılacak 24V DC güç kaynağına ya da 12V DC güç kaynağıyla enerjilendirilecek sensörler araştırılmıştır. Sensörlerin uzak mesafeden cismi algılaması istenen bir özelliktir araştırmalarda bu da dikkate alınmıştır. Kullanılacak motorun ise gelen cisme yön verebilecek bir güce sahip olması beklenmektedir. Bant üzerinde giden cisimlerin en hızlı şekilde ayrıştırılması bunun içinde sensör, PLC ve motor haberleşmesinin en sağlıklı şekilde olması çok önemlidir. Günümüz ihtiyaçları için tasarlanan bu projenin birçok farklı sektörde fabrika sistemlerinde geliştirilerek kullanılması beklenmektedir. Bu beklentiyi karşılamak için özellikle farklı sektörlere hitap edecek şekilde farklı gruplardaki ürünleri boyuna, yapıldığı malzemeye ve rengine göre ayrıştırılması hedeflenmiştir.

BÖLÜM 2

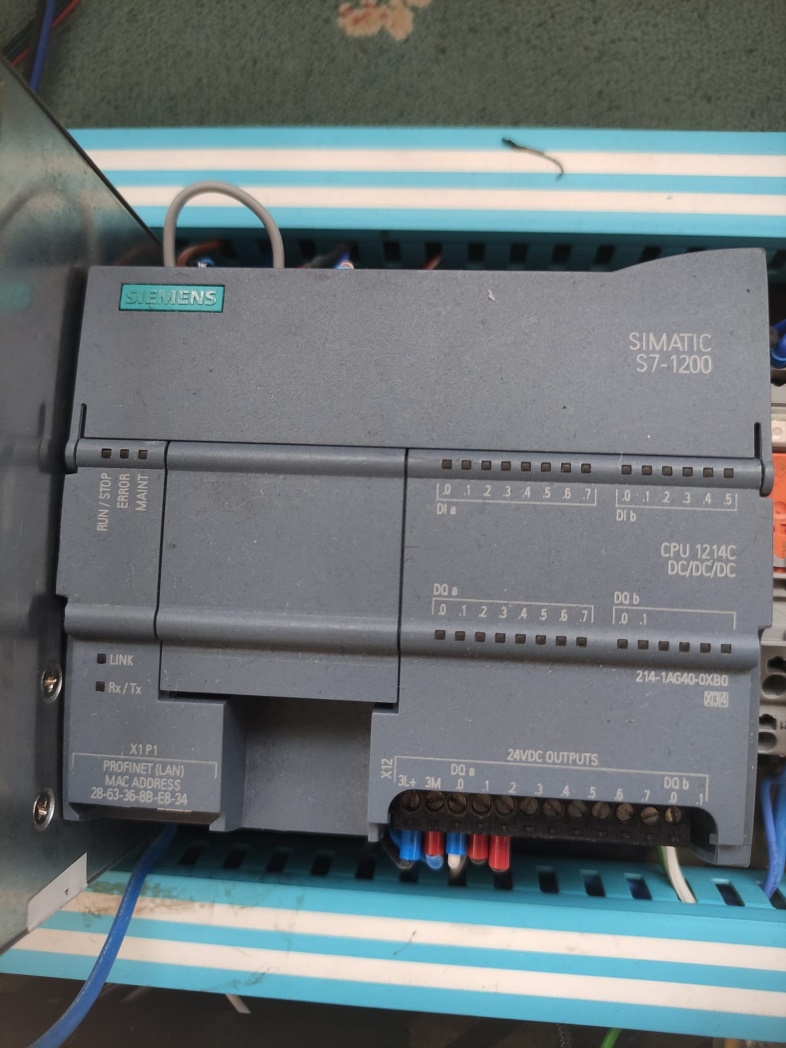
# PLC’NİN YAPISI VE ÇALIŞMA İLKESİ

## PLC Tarihçesi

İlk ticari PLC, 1969 yılında MODICON tarafından geliştirilmiştir. O zamanlar röle kontrol devrelerinin yerini alması için geliştirilen bu cihaza sadece temel mantık işlemlerini gerçekleştirebildiği için PLC adı verildi. İlk PLC'lerin endüstride başarıyla kullanılmaya başlanmasının ardından Allen-Bradley, General Electric, GEC, Siemens, Westinghouse gibi firmalar ekonomik ve yüksek performanslı PLC'ler üretmiş, ardından Mitsubishi, Omron, Toshiba ve diğer firmalar düşük maliyetli, yüksek performanslı PLC'ler geliştirmeye başlamıştır. Bu cihazlar o zamandan beri endüstriyel otomasyon devrelerinde yaygın olarak kullanılmaktadır. Endüstriyel sektörde otomasyon ve endüstriyel kontrolün evrimi, PLC'lerin gerçek konumunu belirlemiştir. Başlangıçta daha basit analog kontrollerle başlayan elektronik kontrol sistemleri zamanla yetersiz kalınca çözüm, analog bilgisayar adı verilen sistemler kullanılarak daha sofistike dijital altyapı sistemlerini devreye sokmak olmuştur. Bu tür denetleyicilerin sadece kontrol döngülerinde değil, geri beslemeli kontrol döngülerinde de kullanılması PLC adının tartışılmasına yol açmıştır. Bu nedenle birçok üretici PLC yerine PC (programlanabilir kontrolör) adını kullanmayı uygun bulmuş, özel şirketler ise kişisel bilgisayarla karıştırılmaması ve başlangıçta PLC adı altında üretildiği için bunu devam ettirmişlerdir.

## PLC Nedir?

PLC adı, Programmable Logic Controller'ın kısaltmasından gelir. Bunlar, sistem veya sorundan bağımsız olarak evrensel olarak oluşturulmuş kontrollerdir. PLC, analog ve dijital I/O kartları aracılığıyla birçok makineyi ve sistemi aynı anda kontrol edebilen bir cihazdır. Merkezi işlem birimi sayesinde sayısal işlemler yapabilmektedir. Aynı zamanda sayaçlar, zamanlayıcılar, karşılaştırmalar, veri işleme ve sıralama gibi işlemler gerçekleştirebilir. PLC’ler, yazılımlar ve programlama desteği yardımıyla giriş bilgilerini kullanarak, çıkış birimlerine veri atayabilen giriş/çıkış birimleri, bellek, merkezi işlem birimi ve programlayıcı biriminden oluşur. Denetleyici sorunlarını gidermek için kullanılan tüm öğeler, PLC üreticileri tarafından sağlanan kullanıma hazır sistemlerdir.



Şekil 2.1. S7-1200 Siemens Marka PLC

### PLC’nin Sağladığı Avantajlar

* Devre tasarımında kolaylık sağlar ve güvenilirdir.
* Kontrol kabininde daha az dağınıklık ve daha az yer kaplar.
* Yeni uygulamalara daha hızlı uyum sağlar.
* Olumsuz çevre koşullarından kolay kolay zarar görmez.
* Kablolama için daha az alan gerekir.
* Programları hazır şekilde kullanma imkânı vardır.
* Giriş/çıkış durumları kontrol edilebilir.

### PLC’nin Sağladığı Dezavantajlar

* Girişteki değişimin en kısa sürede çıkışa gönderilmemesi (seri sinyal işleme) PLC için bir dezavantajdır.

## PLC’nin Elemanları

### Merkezi İşletim Birimi (CPU)

CPU (Merkezi İşlem Birimi) giriş değerini okur. Belleğin yüklenen bilgi ve verilere göre program yürütür ve çıkan değerleri çıktıya aktarır. S7-1200CPU, kompakt bir tasarıma sahiptir. Yani belirli sayıda dijital giriş, çıkış, analog giriş ve yüksek hızlı sayıcı modülleri CPU ile aynı yuva içerisinde üretilmektedir.

CPU, MRES, STOP, RUN, RUN-P, SF, BATF, DC5V ve FRCE çalışma modlarına sahiptir.

MRES; hafızanın sıfırlama işlemini yapar.

STOP; durdurma işlemini yapar.

RUN; çalışma modu bilgisayar okuma işlemidir.

RUN-P; çalışma modu bilgisayar tarafından hem okunabilir hem de yazılabilir.

SF; grup hatasını, yani bir CPU veya modül hatasını gösterir.

BATF; pil hatası, pil voltajın düşük olduğu veya pil olmadığı anlamına gelir.

FRCE; bir veya daha fazla giriş veya çıkışta kullanıcı tarafından zorla işlem yapılıyor demektir.

Daha eski CPU’larda 3.6 voltluk lityum piller bulunur. Ömrü 5 yıldır. Ancak iki yılda bir değişmesi gerektiği söylenir. Programı korur. Donanım ayarlarını korur ve gerçek zaman saatini işletir. PLC ile haberleşmesini mümkün kılar.

Yani CPU türleri, piller yerine süper kapasitörlere sahiptir. Bilgileri ve verileri 10 haftaya kadar saklayabilir.

### Sinyal Board

CPU'nun sağına ve soluna takılan sinyal ve haberleşme modüllerinin yanı sıra CPU'nun kendisine takılan sinyal kartları da kullanılmaktadır. Bu kartlar özellikle az duyulan ihtiyaçlar için dijital, analog giriş, çıkış veya haberleşme kartlarıdır. Sinyal kartı 24 volt DC ile çalışabileceği gibi 5 volt DC ile de çalışabilir. Sayısal sinyal kartı üzerinde 4 giriş 4 çıkış veya 2 giriş 2 çıkış sinyalleri kullanabilir. Analog sinyal bağlantısına bir giriş ve bir çıkış analog sinyali bağlanabilir. Sinyal panosuna piller (akü modülleri) de eklenebilir.

### Sinyal Modülleri

CPU’nun sağ tarafına CPU tipine göre farklı sayılarda giriş çıkış sinyal modülleri eklenebilir. Bu kartlar DI, DO, DIO, AI, AO, AIO, şeklinde olabilir.

Dijital sinyal modüllerinde 8-16 giriş 8/16 çıkış veya 8 giriş / 8 çıkış sinyalleri kullanılabilir.

Analog sinyal modüllerinde ise 4-8 giriş 2-4 çıkış sinyali veya 4 giriş / 2 çıkış bağlanabilir.

### Haberleşme Modülleri

CPU'nun sol tarafına takılır. PROFIBUS kablosu veya PROFINET kablosu bağlanır. Sinyal ve haberleşme modülleri montaj yerlerine takıldıktan sonra doğru tip switch’ler seçilerek CPU'ya bağlanır. Besleme gibi herhangi bir işlem gerekmez.



Şekil 2.2. Haberleşme Modülleri [1]

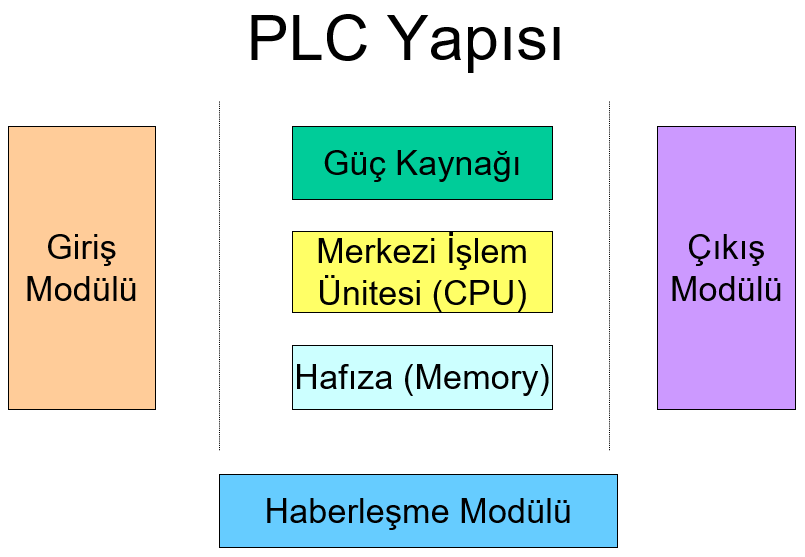
## PLC Giriş Çıkış Birimleri

PLC giriş/çıkış ünitesi, sistemler arasındaki haberleşmenin kontrol edilmesini sağlar. PLC'nin giriş birimleri duyu organları, çıkış birimleri ise el ve ayakları denebilir.

### Giriş Birimi

Sahadaki ekipmanlar tarafından kontrol edilen basınç, seviye, sıcaklık sensörleri, butonlar ve anahtarlar gibi sensörlerin sahadaki elektriksel değerlerini mantıksal değerlere (0'dan 1'e) çevirerek Merkezi İşlem Birimi (CPU)'ya ileten ünitedir.  Giriş devresine gelen sinyalin mantıksal ”1” kabul görünen alt sınır ve mantıksal “0” kabul edilen üst sınırı bulunur.

### Çıkış Birimi

CPU tarafından çıkış komut belleğine (PIQ) yazılan mantıksal değeri (1/0 VEYA analog çıkış sinyali) elektrik sinyaline çeviren birimdir. Kontrollü sistemlerde kontaktör, röle, selenoid gibi kontrol elemanlarının sürmeye uygun donanım üzerinde olan birimdir.  
PLC çıkışından büyük bir akım çekilemez. Çıkış akımı kapasitesi PLC kataloğunda açıklanmıştır. Her karta, kartın takıldığı yuvaya bağlı olarak, yazılımda bir "bayt" ve "bit" adresi atanır. Tüm S7-1200 CPU'lar kompakt olduğundan üzerlerindeki G/Ç kartlarına "0"dan başlayan adresler atanır. Eklenen sinyal modüllerinin adresleri de eklendikleri sırayla verilir. Bununla birlikte, modülün kompakt veya gelişmiş olmasına bakılmaksızın, yazılımda ilk "byte" adresi değiştirilebilir. Çıkış kanalından alınan sinyal kontrol elemanından geçerek güç kaynağının "-" ucuna bağlanır. Kanal adresleri, giriş kartlarıyla aynı mantık kullanılarak oluşturulur. Adın 'I' girişi yerine 'Q' çıkışıyla başlaması yeterlidir. Giriş/çıkış elemanları ile giriş/çıkış üniteleri arasındaki bağlantı çok basit ve pratiktir. Herhangi bir problemde mevcut modülün yenisi ile değiştirilmesi çok kolaydır. Giriş/çıkış devresinin ON\OFF durumu, yerleşik LED lamba ile gösterilir.

Şekil 2.3. PLC Yapısı [2]

### Giriş-Çıkış Birimlerine Bağlanabilecek Bazı Algılayıcılar;

* Sınırlama anahtarları (limit switches)
* Basma düğmeler (push buttons)
* Basınç anahtarları (pressure switches)
* Seviye anahtarları
* Basınç uyarıcıları
* Dönüştürücüler (transducers)
* Seçici anahtarlar (selector switches)

### PLC’nin Çıkış Sinyallerine Göre Çalışan Bazı Dış Ortam İşlemcileri;

* Selenoid valfler
* Lambalar
* Motor sürücüleri
* Röleler
* Kontaktörler
* Göstergeler

## PLC’nin Çalışma Fonksiyonu

Bir PLC temel olarak üç bölümden oluşur. Birinci kısım merkezi işlem yani CPU, ikinci kısım tüm sinyal elemanlarının bağlı olduğu giriş kısmı (giriş kartı), üçüncü kısım ise kontrol edilecek cihazın bağlı olduğu çıkış kısmıdır (çıkış kartı). PLC, kendi güç kaynağı voltajını CPU için DC 24V veya AC 220V'den 24V'a düşüren bir modele sahiptir. Giriş ve çıkış kartları 24 ile 220 V arasında değişmektedir. Sahadan gelen sinyallerin durumuna göre (0/1) şeklinde Proses giriş resmi=PII tablosuna yazar. Aynı şekilde çıkışlar proses çıkış resmi = PIQ tablosuna yazar. PII'den okunan bilgileri, PII üzerindeki mantık işlemlerine göre elde edilen çıktılara yazılır. Tüm komutlar işlendikten sonra PIQ'da sonuç çıkış kartına ok içeren bir resim

Açıklama otomatik olarak oluşturulduyazılır ve PII'deki verilerle yeni bir döngü başlar.

Şekil 2.4. PLC Basit Çalışma Döngüsü [2]

## Program İşleme Şekilleri

### Lineer Program İşleme

Bu projede PLC geliş programı, tek bir blok üzerinden program alt programı olarak kullanılmamaktadır. Çünkü tek bir kod parçası ile oluşturulmuştur. Komutlar, program belleğinde sırayla işlenir. Son satırdaki komut işlendikten sonra program ilk satırdan devam eder. Kısacası sürekli bir döngü vardır. Programdaki bütün komutların bir kez işlenmesi için gerekli zamana çevrim süresi yani cycle adı verilir. Doğrusal programlama, tüm programın tek blokta yazıldığı basit, ayrıntılı olmayan programlar için kullanılır.

### Yapısal Program İşleme

Programlar, işlevlerine göre daha küçük mantıksal alt bloklara ayrılır. Alt programları belirli bir sırayla çağırarak bir program oluşturur. Program tarafından kullanılan veriler de veri bloklarında saklanır.

## PLC Modelleri

S7-300, S7-400, S7-1200, S7-1500 SIEMENS PLC modelleri bulunur. Bunların S7-1200 ve S7-1500 modelleri kompakt yapılıdır. Bu yapılar, modeli maliyet, performans ve yerden tasarruf açısından daha da avantajlı ve kolay kullanımlı hale getirmektedir.

BÖLÜM 3

# PROJE ELEMANLARI

## Güç Kaynağı Nedir?

DC güç kaynakları çoğu elektronik işleriyle uğraşanların en önemli araçlarındandır. Hazırlanan devreleri, yapılan projeleri enerjilendirmek için güç kaynaklarından yardım alınır. Yapılan bu devrelerde ihtiyaç duyulan gerilim ve akım değerleri farklılık gösterebilir. Bu nedenle sabit güç adaptörleri kullanmak yerine potansiyometre ile ayarlanabilir bir DC güç kaynağı kullanımı daha pratik olur. Ayarlanabilir güç kaynağı kullanıldığı takdirde istenilen gerilim değerlerinde çıkışlar elde edilebilir.

### Güç Kaynağı Nasıl Çalışır?

Şehir şebeke gerilimi genel olarak 220 Vrms/50 Hz’dir. Bu gerilimi istenilen bir DC gerilim değerine dönüştürmek için transformatörler kullanılır. Transformatör kısaca akımı ve frekansı sabit kalmak şartıyla gerilimi yükseltip alçaltmaya yarayan bir devre elemanıdır. DC güç kaynaklarının temelde basit bir mantığı vardır. Şebekeden alınan AC gerilim transformatörler yardımıyla belli oranda düşürülür. Bu düşürme oranı trafoların üretimi ile alakalı bir durumdur. DC güç kaynaklarında şebeke geriliminin istenilen değerlerde olması için trafo kullanılmıştır. Şebeke gerilimi düşürüldükten sonra trafo çıkışından elde edilen gerilim yine AC’dir. Dolayısıyla AC gerilimi DC ye çevirmek için bir yapıya ihtiyaç vardır. Bu ihtiyaç diyotlar sayesinde giderilebilir. Piyasada, iç yapısındaki diyotlar sayesinde doğrultma sağlayan elektronik elemanlar bulunur. Bu elemanların 2 AC girişi, 2 de doğrultulmuş olan DC çıkışı bulunur. Buradan elde edilen DC çıkışına kapasitör kullanarak daha düz bir çıkış elde edilir. Bu sayede 220 V AC giriş 5/12/24 V DC gibi istenen değerlere dönüştürülebilir.

### **Valx VMA-1210 Güç Kaynağı**

Projede kullanılan 12 V güç kaynağı Valx VMA-1210 için modelin teknik özellikleri, şebekeden gelen kablo için 1 adet topraklama ucu ve 1 adet + uç ve 1 adet – uç bulunmaktadır. Çalışma sıcaklığı -10 ile +70 derece arasında olmakla birlikte çalışma nem oranı %20 ile %80 arasındadır. Güç kaynağı kısa devre, aşırı yük, aşırı voltaj, aşırı sıcaklık gibi etkenlere karşı koruma içerir. Voltaj ayarı mevcuttur ancak projede gerek duyulmamaktadır. Çıkışından 12 V ve maksimum 10 A elde edilebilir. Projede bu model güç kaynağının kullanılmasının sebepleri düşük maliyetli, yüksek verimli ve uzun ömürlü olmasıdır.

Şekil 3.1. 12V-10A Güç Kaynağı [3]

### Valx VMA-2410 Güç Kaynağı

Projede kullanılan 24 V güç kaynağı Valx VMA-2410 için modelin teknik özellikleri, şebekeden gelen kablo için 1 adet topraklama ucu ve 2 adet + uç ve 2 adet – uç bulunmaktadır. Çalışma sıcaklığı -10 ile +70 derece arasında olmakla birlikte çalışma nem oranı %20 ile %80 arasındadır. Güç kaynağı kısa devre, aşırı yük, aşırı voltaj, aşırı sıcaklık gibi etkenlere karşı koruma içerir. Voltaj ayarı mevcuttur ancak projede gerek duyulmamaktadır. Çıkışından 24 V ve maksimum 10 A elde edilebilir.



Şekil 3.2. 24V-10A Güç Kaynağı [4]

Proje için gerekli olan güç kaynakları tespit edildiğinde, sensörler için gerekli olan değerin 24 V olduğu görülmektedir. Konveyör sistemi için kullanılan motorun 12 V besleme sayesinde çalıştığı, PLC’nin beslemesi ise 24 V ile çalıştığı bilinmektedir. Projede kullanılan motor, PLC ve sensörler için güç kaynaklarında değişiklik yapılmasına gerek olmadan bağlantıları yapılabilir.

## Konveyör Bant

Projede kullanılan konveyör bant için merdiven korkuluklarından oluşan bir kasa oluşturuldu. Bu metal kasanın iki ucunda konveyör bandın dönme hareketini yapabilmesi için silindirler yerleştirildi. Daha sonra konveyör üzerinde cisimlerin taşınabilmesi için 30 cm genişliğinde bant kullanıldı. Bandın bir silindirine 12 V besleme ile çalışan bir motor bağlandı. Sensörler için ve kameranın sabit durabilmesi için metal korkuluklara üç boyutlu

yazıcıdan çıkarılan model yerleştirildi. Konveyör bandın boyu 140 cm eni ise 43,5 cm’dir.

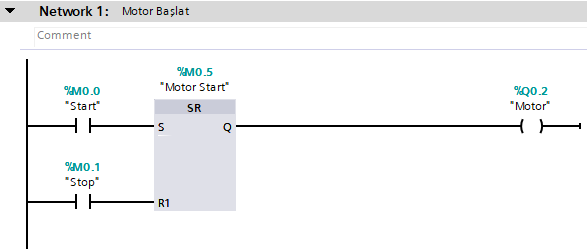
duvar, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3.3. Konveyör Bant

### PLC ile Konveyör Çalışması

Network 1 motor başlatma kısmını oluşturur. Start butonu 1 ve stop butonu 0 olduğunda Q0.2 çıkışı SR tarafından aktif kalacağından stop butonuna basılana kadar motor çalışmaya devam edecektir. Bu sisteme yazılımsal kilitleme yani mühürleme adı verilir. Projelerde mühürleme devresi motorun çalışma devamlılığını sağlaması amaçlı uygulanmaktadır. Stop butonunun bir kere basılması durumunda hat kesilecek, dolayısıyla motor çalışmayı durduracaktır.



Şekil 3.4. Konveyör Bant Ladder Devre Şeması

## Sensörler ve PLC Bağlantıları

Ürün ayrıştırma sırasında 2 adet MZ80 kızılötesi mesafe sensörü (E18-D80NK) ve 1 adet endüktif metal sensörü (LJ18A3-8-Z/BY) kullanılmıştır. Sensörler dijital olarak çıkış verir. PLC’de dijital input modülüne bağlanarak TIA Portal ’da normal kontak şeklinde yapılır.

### MZ80 Kızılötesi Mesafe Sensörü (E18-D80NK) Tanımı ve Bağlantısı

Kızılötesi mesafe sensörleri, mesafeyi, aydınlık veya karanlıkta algılamada kullanılan sensörlerdir. LED yapısıyla kızılötesi ışık yayan ve bu ışığın yansımasını kontrol eden fotodiyot bulunur. NPN NC (Normally Closed/Normalde Kapalı) tipindedir. MZ80’in menzili 3-80 cm arasında ayarlanabilir. 24V ile çalışır, tepki süresi 2ms’dir. +24V GND ve Data çıkışı bulunur. +24V ve GND uçlarını ayarlanabilir güç kaynağının soketine, sinyal ucu ise PLC’nin belirlenmiş dijital input girişine bağlanır.



Şekil 3.5. MZ80 Kızılötesi Mesafe Sensörü

Projede boyut ayrımı sırasında küçük ve büyük materyalleri ayırt edebilmek için üst üste iki adet MZ80 koyulması tasarlanmıştır. Konveyör banttan küçük olan cisim geçerken sadece alttaki sensör tarafından algılanır ve PLC’ye sinyal iletilir. PLC bu sinyale göre step motorun gerekli açıyı almasını sağlar. Aynı şekilde büyük bir cisim geçtiğinde ise üst üste bulunan iki sensör de cismi algılayıp PLC’ye sinyal gönderir. PLC de buna göre step motorun istenilen açıyı almasını sağlar.

mutfak aletleri, oklava, merdane, ahşap, tahtadan, iç mekan içeren bir resim

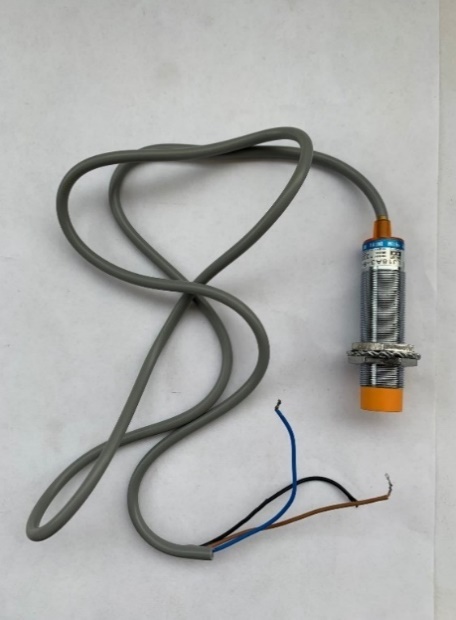
Açıklama otomatik olarak oluşturuldu

Şekil 3.6. Sensör Muhafazasının Tasarımı

Şekil 3.7. Büyük ve Küçük Ürünler

### Endüktif Metal Sensörü (LJ18A3-8-Z/BY) Tanımı ve Bağlantısı

Konveyör bant üzerinden gelerek menziline giren metal cisimleri algılayan sensördür. Algılama alanı içinde manyetik alan vardır. Bu alanın içine giren metal materyaller manyetik alanı etkiler. Etkilenme, sensörde bulunan elektronik devrelerde kayıt altına alınır ve sensörden çıkış sinyali verir. Böylece sensörün algıladığı cismin metal olup olmadığı tespit edilir. Besleme gerilimi 6 V-36 V DC’dir. Bu sensör projede bulunan 24 V DC güç kaynağı ile enerjilendirilecektir. Algılama mesafesi 8mm’dir. Bu yüzden materyallerin sensör menziline girmesi için çok yakınından geçmesi gerekmektedir. Sensörün tipi PNP NO (Normally Open/Normalde Açık)’dır. Sensörde bulunan + ve – uçları 24 V güç kaynağına bağlanırken sinyal ucu ise PLC’nin dijital input modülüne bağlanır.



Şekil 3.8. Endüktif Metal Sensörü

Konveyörden gelen metal bir nesne sensörün menziline girdiğinde, sensör PLC'ye bir sinyal gönderir ve PLC sinyali tanır. PLC daha sonra step motora metal nesneleri ayırması için bir sinyal gönderir. MZ80 kızılötesi mesafe sensörü, metalik olmayan nesneleri algılarken aynı zamanda boyut ayrımı için kullanılmaktadır. Konveyörün üzerinden metalik olmayan bir nesne geçtiğinde, MZ80 bunu algılar ancak endüktif sensör algılamaz. PLC'ye yalnızca mesafe sensörünün sinyali gönderilir. Metal nesne geçtiğinde endüktif sensör algılar ve PLC'ye gönderildikten sonra kilitleme oluşturularak endüktif sensörün çalışmasından sonra MZ80 mesafe sensöründen gelen sinyalin herhangi bir işlevi olmayacağından doğru bir şekilde algılama gerçekleşir ve uygun şekilde step motora bilgi gönderilir.

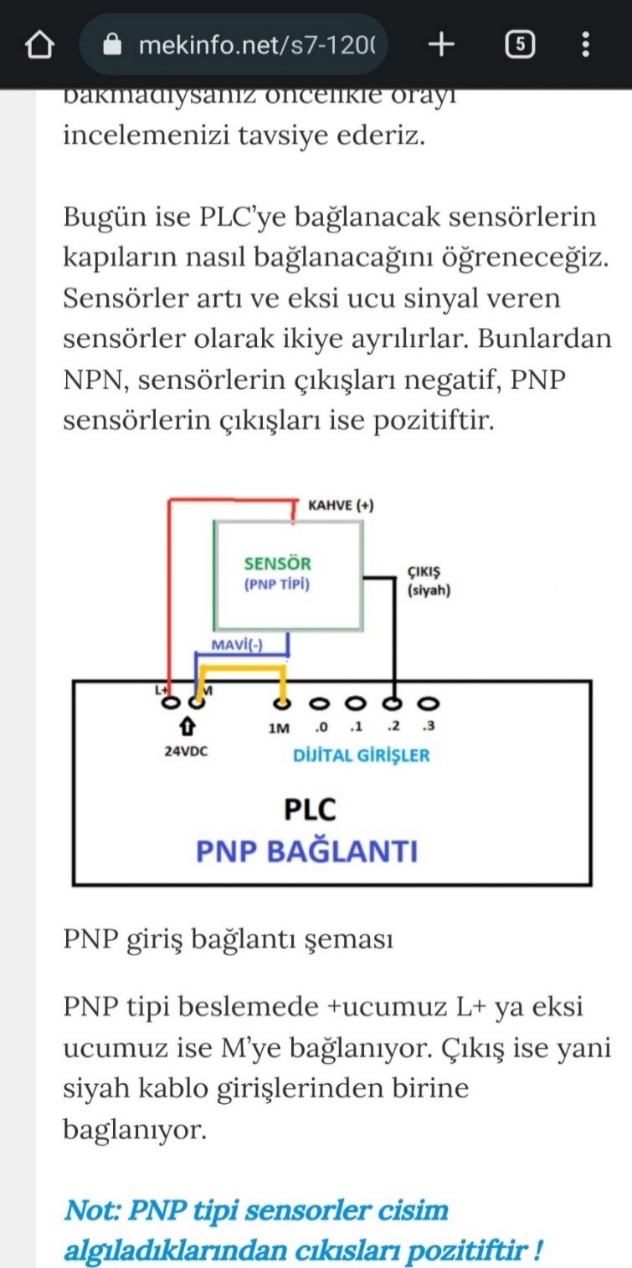


Şekil 3.9. Algılanacak Materyaller (Metal-Metal Olmayan)

## Sensörlerde PNP ve NPN Tiplerinin PLC Bağlantısı

### PNP Giriş Bağlantısı

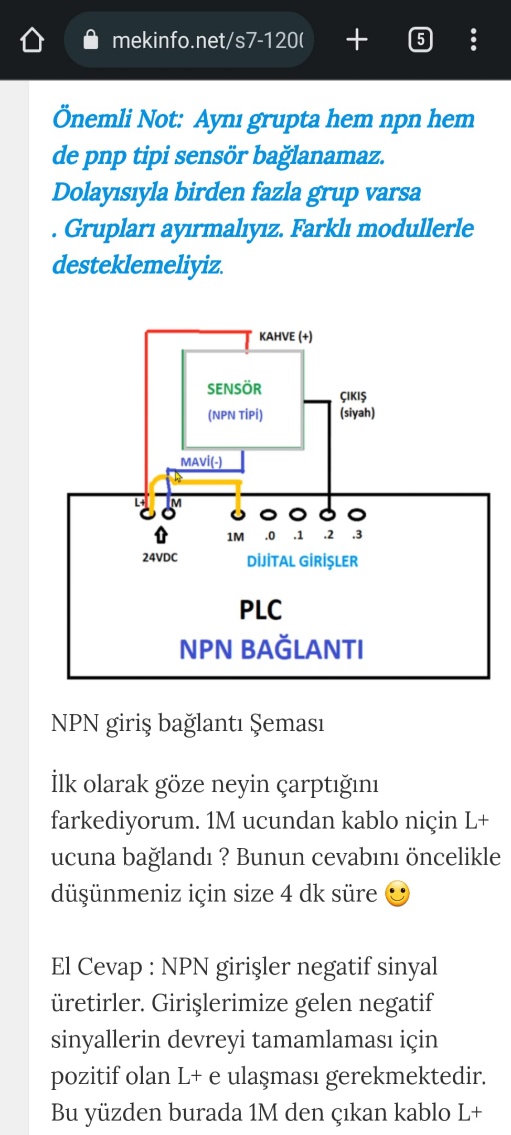
PNP tipi beslemede + uç PLC üzerinde bulunan L+ bölümüne, – uç ise M portuna bağlanır. Sinyal ucu PLC’nin belirlenen dijital girişine bağlanır. Bunun sonucunda sensörden gelen veriler PLC’ye iletilir ve 1M aracılığıyla M’ye bağlanır ve devre tamamlanmış olur. Bağlantı olmaması durumunda algılama yapılamayacaktır. PNP tipi algılayıcılar pozitif çıkış verir.



Şekil 3.10. PLC Üzerinde PNP Bağlantı Şeması [5]

### NPN Giriş Bağlantısı

NPN tipi beslemede + uç PLC üzerinde bulunan L+ bölümüne, – uç ise M portuna bağlanır. Sinyal ucu PLC’nin belirlenen dijital girişine bağlanır. NPN tipi bağlantılarda PNP’den farklı olarak 1M’den çıkan kablo L+ ucuna bağlanmalıdır. Böylece sensörler tarafından iletilen negatif sinyaller pozitif L+ bölümüne ulaşarak devreyi tamamlar.



Şekil 3.11. PLC Üzerinde NPN Bağlantı Şeması [5]

### Endüktif Metal Sensörünün TIA Portal Bağlantısı

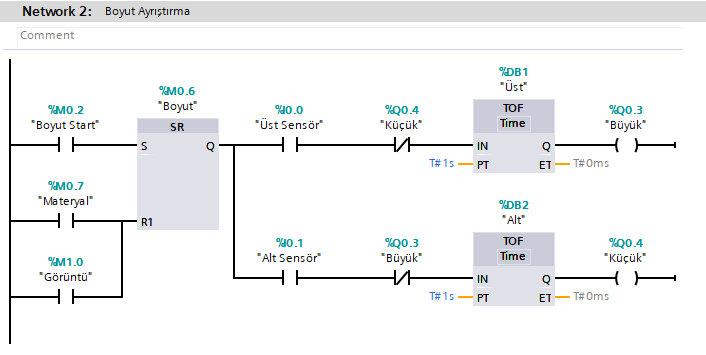
Materyal start aktif olduğunda metal veya metal olmayan algılama sistemi çalışacaktır. Global memory şeklinde tanımlanan materyal start butonu aktif olduğunda set öncelikli çalıştığı için çıkış verecektir. Endüktif sensör PLC’de I0.2 adresine tanımlanmıştır. Sensör ürünü gördüğü zaman 5 saniye boyunca metal çıkışını verecektir. Burada endüktif sensörü algılamadığında alt tarafta bulunan kızılötesi sensör I0.1 adresinden algıladığında metal olmadığı anlaşılacaktır. Her ikisinin çıkışları birbirinin önüne koyulup kilitleme oluşturulmuştur. Sistem set öncelikli çalıştığı için boyut ve görüntü start aktif olduğunda materyal ayrıştırma çalışmayacaktır.



Şekil 3.12. Endüktif Metal Sensörü TIA Portal Bağlantısı

### MZ80 Kızılötesi Mesafe Sensörü TIA Portal Bağlantısı

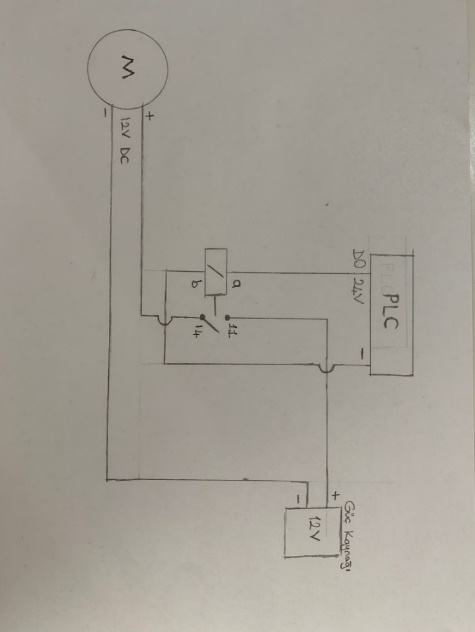
Network 2 boyut ayrıştırma şeklindedir. Boyut start butonu global memory yapılmıştır. Bu buton SR den geçerek aktif hale gelir. Kızılötesi sensörlerden üstte bulunan sensör I0.0 adresine altta bulunan sensör I0.1 adresine tanımlanmıştır. Büyük ürün geçtiği zaman iki sensörü de göreceği için alt ve üst sensörler arasında kitleme olacak şekilde biri geldiğinde diğeri görmemesi adına yapılmıştır. Sensörler ürünü tanıdıktan 1 saniye boyunca PLC ye veri akışı sağlayarak ürünün büyük veya küçük olduğunu belirtmiş olur. Reset öncelikli çalıştığı için materyal ve görüntü geldiğinde boyut aşamaları çalışmayacaktır.



Şekil 3.13. MZ80 Kızılötesi Mesafe Sensörü TIA Portal Bağlantısı

## PLC ve 12 V DC Motor Haberleşmesi

12 V DC motorun çalışabilmesi için 12 V’luk güç kaynağına ve röleye ihtiyaç vardır. Aynı zamanda motoru uzaktan kontrol etmek ve izlemek için PLC kullanılarak röle ile bağlantısı yapılır. PLC kullanımında PLC’yi beslemek için 24 V güç kaynağı kullanılır. PLC’nin 24 V dijital output ucu ile rölenin A1 bobin giriş ucu bağlanır. Rölenin A2 bobin çıkış ucuna ise -24 V güç kaynağı ucuna bağlanır. 24 V’luk güç kaynağının – ucundan PLC’deki M’ye (-24 V DC) bağlanır. 24 V güç kaynağının – ucu PLC’de bulunan 24 V DC output kısmından 3M’ye bağlanır. +24 V güç kaynağından PLC’de bulunan 24 V DC output un 3L+ kısmına bağlanır. 24 V güç kaynağının + ucundan PLC’nin 24 V DC L+ ucuna bağlanır.

Motorun çalışabilmesi için 14 numaralı kontak (NO (Normally Open)), motorun +12 V DC ucuna bağlanır. Rölenin 11 numaralı kontağı yani COM ucu +12 V DC güç kaynağına bağlanır. Motorun -12 V DC ucu 12 V güç kaynağının – ucuna bağlanır.

Şekil 3.14. Motor, Röle, PLC ve Güç Kaynakları Bağlantı Şeması

elektronik eşyalar, bağlantı parçası içeren bir resim

Açıklama otomatik olarak oluşturuldu

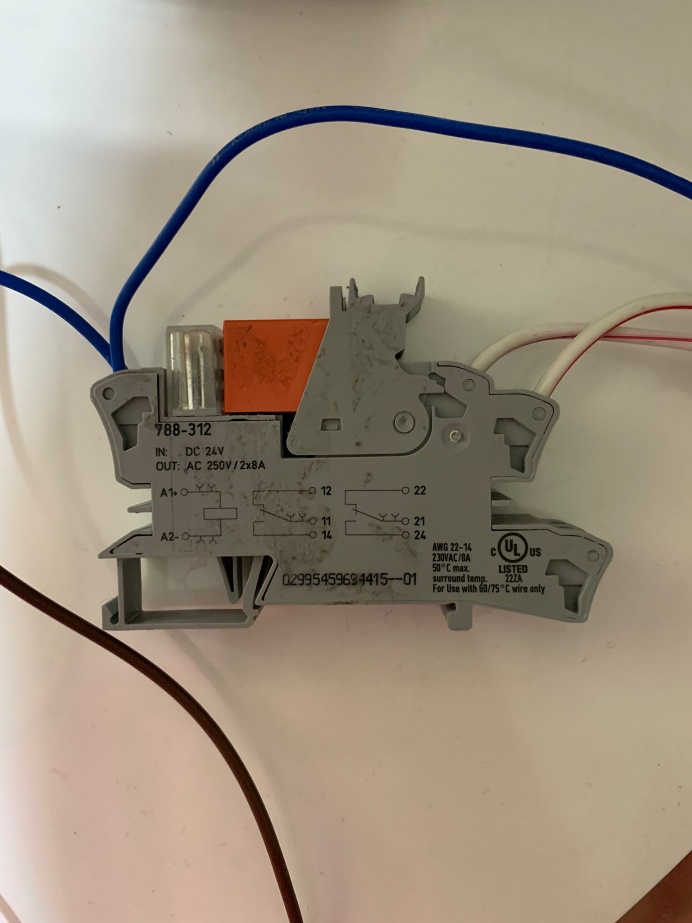
Şekil 3.15. Motor, Röle, PLC ve Güç Kaynakları Bağlantısı

## Röle Nedir?

iç mekan, bağlantı parçası içeren bir resim

Açıklama otomatik olarak oluşturulduProjede RT424024 kodlu girişi 220 V çıkışı 24 V DC röle kullanıldı. Röle üzerinden akım geçtiğinde tetiklenen elektriksel bir anahtardır. Röleler birden fazla anahtar kontağı içerebilir. Böylece birçok yükü aynı anda anahtarlayabilirler. Röleler; bobin, palet ve kontak olmak üzere 3 bölümden meydana gelirler. Rölenin giriş kısmında bobin bulunur. Diğer kısımlarının bobin ile herhangi bir elektriksel bağlantısı yoktur. PLC’den çıkan 24 V dijital output rölede bulunan bobinin A1 yani bobinin giriş ucuna bağlanır. A2 yani bobin çıkış ucundan 24 V DC güç kaynağının – ucuna bağlanır. Rölede bulunan 11 kontağı yani COM ucu 12 V güç kaynağının + ucuna bağlanır. Rölede bulunan 14 kontağı yani NO, motorun 12 V DC + ucuna bağlanır. Böylelikle güç kaynakları motor ve PLC arasında kullanılır.

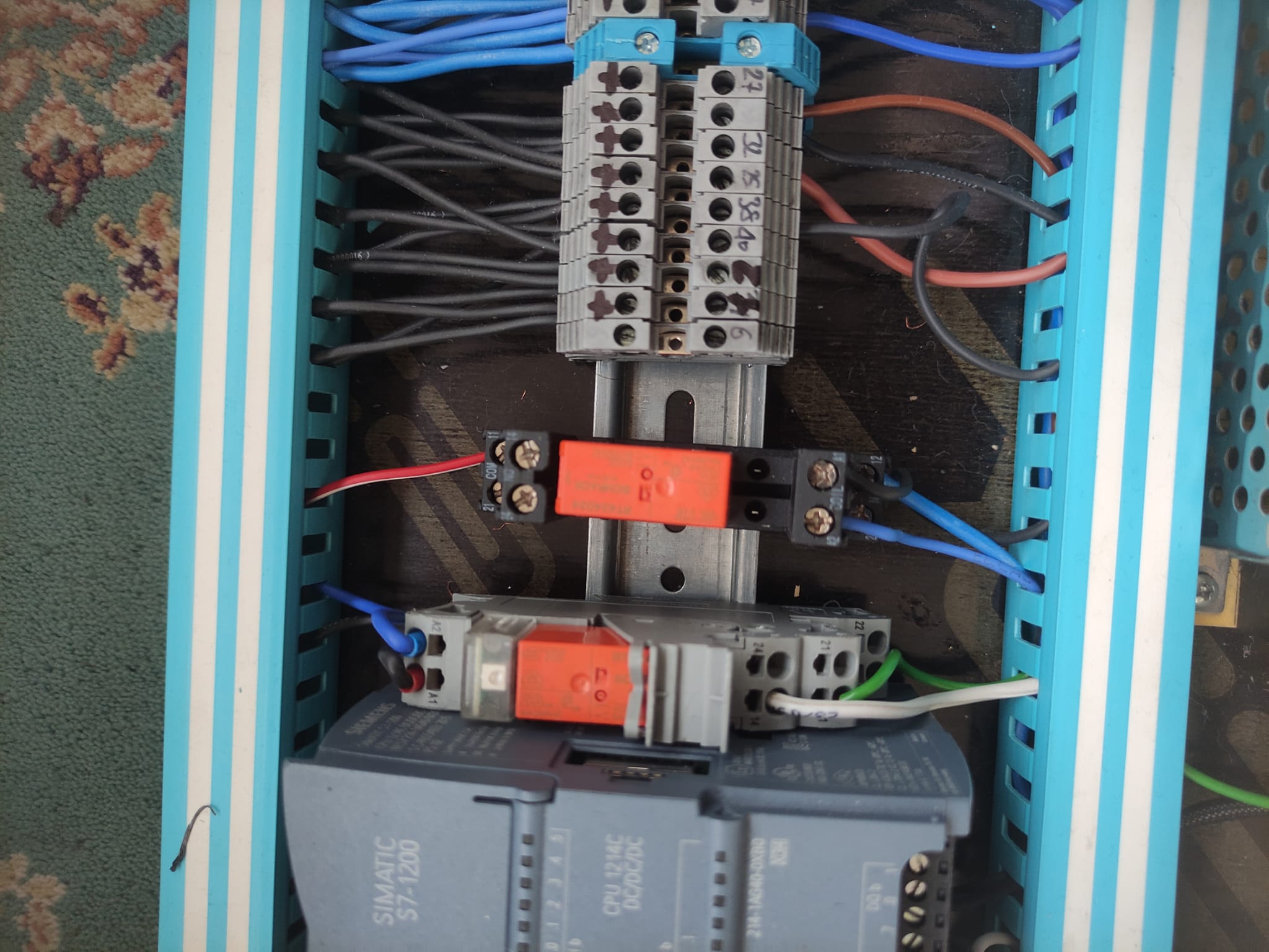
Şekil 3.16. Röle Yandan Görünüm



Şekil 3.17. Röle Üstten Görünüm ve Kontak Numaraları

### PNP ve NPN Türlerinin PLC de Birlikte Kullanılması

NPN kızılötesi mesafe sensörü gördüğünde çıkışa 0 V gönderdiği için 1M bağlantısının L+ ucuna bağlanması gerekmektedir. Ancak PNP Endüktif sensörü gördüğünde 24 V gönderdiği için 1M bağlantısının L- ucuna bağlanması gerektiğinden iki sensör de aynı PLC de kullanılamama durumu oluşmaktadır. Sensörün GND ucu – baraya, 24 V ucu + baraya, sinyal ucu rölenin A1 giriş ucuna bağlanır. Rölenin A2 bobin çıkış ucundan ise PLC nin dijital girişine bağlanılmasıyla PNP türü NPN e dönüştürülmüş olur. Bu yüzden röle ile PNP tip sensörün çıkışı terslendiğinde, sensör gördüğünde çıkışı 0 V yapmaktadır.



Şekil 3.18. PNP, NPN Dönüştürme Röle Bağlantısı

## Step Motor Nedir?

Step Motorlar, enerji olarak elektrik enerjisini fiziksel enerjiye dönüştüren elektromekanik makinelerdir. Step motorların çok çeşidi vardır. Bunlar planlanan sistemlerin gerekliliklerine göre hizmet ederler. Step motorlardan yüksek verim elde edilir. Step motorlar yapısı itibariyle içerisinde mıknatıs olan rotordan oluşur. Stator sargıları ile çevrilmiş haldedir. Step motor harekete geçtikten sonra içinden akım geçmesine izin verildiğinde statorda bulunan sargılar mıknatıslanarak sırayla motoru itmeye sebep olur. Bu elektromanyetik itmeye step yani adım denir ve step motorların çalışma ilkesini oluşturur. Bu projede gerekli olan güç faktörü göz önüne alınarak tercih edilen 17HS4401 Nema 17 Step Motor kullanılmıştır.

duvar, kirli içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3.19. Nema 17 Step Motor

### Step Motorun Özellikleri

Kullanılan step motorun NEMA 17, modeli 17HS4401 olup tipi ise bipolar step motordur. Motorun ağırlığı 280 gramdır. Adım açısı 1.8° olan step motorun uzunluğu 40 mm’dir. Faz başına direnci 2.2 Ω olan motorun faz indüktansı 2.8 mH’dir.Tutunma torku 40 N.cm’dir, yani motor, belirli bir yük altında dururken hedeflenen torku üretebilir. Tetik torku 2.2 N.cm’dir, yani motorun hareketini başlatmak için uygulanan en düşük tork değeridir. Rotor ataleti 54 g.cm karedir, yani rorotun dönme hareketi için gereken enerji miktarını ifade eder. Motorun nominal gerilimi 3.4 V DC'dir ve nominal akımı 1.5 A/fazdır. Yani motor 3.4 V DC gerilim ve 1.5 A/faz akım kullanılarak çalıştırılabilir. Bu özellikler motorun elektriksel davranışını anlamak için önemlidir.

### Step Motor Çeşitleri

Step motorların çok fazla çeşidi olmakla beraber, bunların bir kısmı spesifik sürücüler gerektirir. En yaygın şekilde kullanılan çeşitleri; Kalıcı Mıknatıs ya da Hibrit Step Motorlar, 2-faz bipolar step motorlar ve 4-fazlı unipolar step motorlardır. Dikkat edilmesi gereken en önemli konulardan biri motorun hangi işi yapması beklendiğidir. Step motorlar çok büyük ve çok küçük boylarda olabileceği gibi, güçleri de boyutuna göre değişmektedir. Motorun yapması istenilen işe uygun olduğunu yahut olmadığını anlamak için torkuna bakmak gerekmektedir. Örneğin NEMA 17 tipi, üç boyutlu yazıcılarda ve küçük CNC makinelerinde kullanılabilmektedirler. Bundan daha küçük motorlar ise robotik gibi daha hassas işlerde, daha büyük motorlar ise endüstriyel uygulamalarda kullanılmaktadır. NEMA sayıları, motorun montajı için gereken levhanın boyutunu gösterir, step motorun başka niteliklerini anlatmamaktadır. Mesela 2 ayrı NEMA 17 tipi motor, baştan aşağı birbirinden başka elektriksel ya da mekanik nitelikleri barındırabilirler.

### Adım Sayısı

Bu konuda dikkat edilmesi gereken bir diğer husus, istenilen konumlandırma çözünürlüğüdür. Çözünürlük genel olarak, adım başına derece şeklinde tanımlanır. Step motorlarda adım sayısı, bir devir için 4 ila 400 arasında değişim göstermektedir. En genel kullanılan adım sayıları 24, 48 ve 200 olarak tespit edilmiştir.

### Unipolar ve Bipolar Sürücüler

Unipolar sürücülerin, enerji verme şekli değişmez, her durumda fazlara enerji verir. Ortak uç daima negatif, öteki uç ise daima pozitif olur. Bu sürücü tipi, basit transistör devreleri ile kullanılabilir. Bobinlerin yalnızca %50’sini aynı anda enerjilendirebildiği için daha az torkunun olması bu tipin bir dezavantajıdır.

Bipolar sürücüler, fazlarda gerçekleşen akım akış durumunu tersine döndürmek amacıyla H köprü devresi kullanarak motoru sürer. Polariteyi değiştirip fazlara enerji verir, bu sayede unipolar tipten farklı olarak bobinlerin tamamı enerjilendirilir.

2 kutuplu 2 fazlı motorlarda iki bobin grubu vardır. Bu tipteki motorlarda her faz adına ikişerden dört tel vardır. Bazıları ise, tek veya iki kutuplu olarak çalıştırmaya uygun biçimde esnek kablolar ile sunulur.

Çeşitlerinden bahsetmek gerekirse 4 kablolu, 5 kablolu, 6 kablolu, 8 kablolu, 4 fazlı unipolar, 2 fazlı seri bipolar ve 2 fazlı paralel bipolar sayılabilir.

### Dijital Step Motor Sürücüsü (JKD556)

32-bit DSP işlemci tabanlı yeni nesil dijital 2 fazlı step motor sürücüsüdür. 32 bit DSP işlemcisi, karmaşık algoritmaların uygulanmasını sağlar. 20 V DC'den 50 V DC'ye kadar besleme gerilimi aralığı bulunmaktadır. Faz akımı 5,6 A'nın altında olan çeşitli 2 fazlı hibrit kademeli motorları sürmek için uygundur.



Şekil 3.20. Dijital Step Motor Sürücüsü Pin Bağlantıları

### Avantajları Nelerdir?

-Yüksek performans, düşük fiyat olması,

-Mikro adım fonksiyonunun olması,

-Otomatik boşta akım azaltma özelliği bulunması,

-Optik izolasyon sinyalleri I/O bulunması,

-200Kpps'ye kadar maksimum yanıt frekans aralığı bulunması,

-Düşük sıcaklık artışı, yumuşak hareket özelliklerinin mevcut olması,

-Çevrimiçi uyarlanabilir PID teknolojisinin bulunmasıdır.

### Uygulama Alanları

Step motor sürücüsü; eşit küçük ölçekli otomasyon ekipman ve cihazlarında, etiketleme makinesi, kesme makinesi, paketleme makinesi, çizim makinesi, gravür makinesi, otomotiv endüstrisi, güneş enerjisi izleme sistemleri ve CNC makinesi gibi alanlar uygulama alanlarıdır. Genel bir tanım yapılırsa düşük titreşim, düşük gürültü, yüksek hassasiyet ve yüksek hız gerektiren ekipmanlar da kullanım alanları arasındadır.

### Step Motor-Driver Mekanik Bağlantısı

PLC’nin 1214C DC/DC/DC 4.4 versiyonunda bulunan çıkışlarda Q0.0 ve Q0.1 için frekans üretimi olarak 100 kHz çıkış sağlayabilir. Ancak geri kalan çıkışlarda ise 20 kHz’e kadar sağlayabilir. Bu sebepten dolayı TIA Portal üzerinden seçilen Q0.0 çıkışı step motor sürücüsünün PULSE+ girişine bağlanır. Bağlanmasının sebebi step motorun PLC ile haberleşmesini sağlamaktır. PLC’nin Q0.1 çıkışı step motorun DIR+ girişine bağlanır. Yani TIA Portal üzerinden Direction Output Q0.1’e bağlanmış olacaktır. Sürücü üzerinde bulunan ENABLE girişinin amacı step motor üzerindeki yükün kontrol edilmesine yarar sağlar. Step motor çalışmazken sürücü step motoru enerjilendirir. Bu yüzden step motorda aşırı ısınmaya sebep olur. ENABLE aktif olduğunda step motor harekete geçecek kapalı olduğunda ise çalışmayarak step motor üzerindeki yükü ortadan kaldırmış olacaktır. ENABLE girişi PLC’nin Q0.2 çıkışı üzerinden kontrol edilecektir. PULSE– ile DIR–birbiriyle köprülenecektir. Sürücüye enerji verilirken sürücüde bulunan GND girişi 24 Voltluk güç kaynağının – girişine bağlanır. Sürücünün VDC girişi ise 24 Voltluk güç kaynağının + girişine bağlanır. Step motorun datasheet’de bulunan akım değeri 1.7 A olduğundan sürücü üzerinde bulunan PA SETTING kısmında bulunan SW1-SW2-SW3 kullanılarak SW1-on, SW2-off, SW3-off konumuna getirilerek RMS 1.5 A seçilmiştir. Step motor PULSE-REV üzerinde bulunan 3200 değeri için SW5, SW6, SW7, SW8 kullanılarak sırasıyla off, off, on, on konumuna getirerek seçilmektedir.

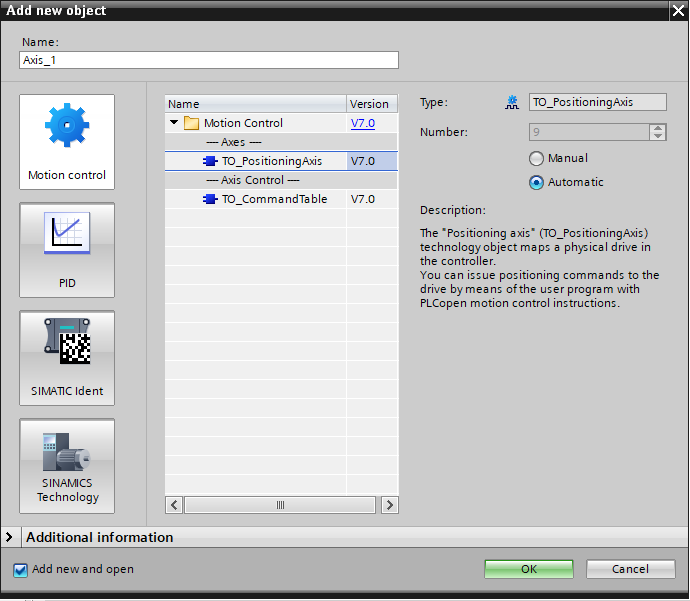
metin, elektronik eşyalar içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3.21. Dijital Step Motor Sürücüsü ve Step Motor Pin Bağlantıları

### Step Motor PLC Haberleşmesi

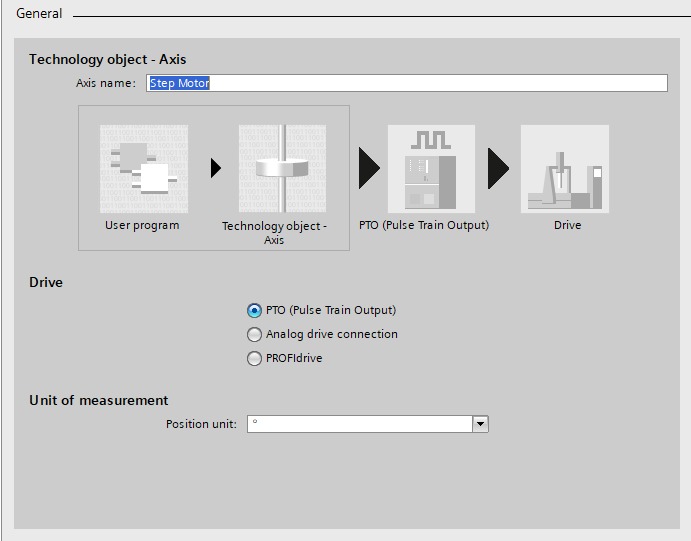
Oluşturulan TIA portal projesi üzerinden technology objects kısmından add new object seçilerek TO\_PositioningAxis yardımıyla step motor, TIA Portal üzerinden ayarlama yapılmıştır. Drive kısmı için şekilde gözüktüğü üzere ayarlamalar yapılmıştır. Hız kontrolü, pozisyon kontrolü, ivme kontrolü, home ayarları, pozisyon limitleri vb. ayarlamalar konveyör bant üzerinde çeşitli denemeler ile istenilen sonuçlar ile parametreler kaydedilmiştir.



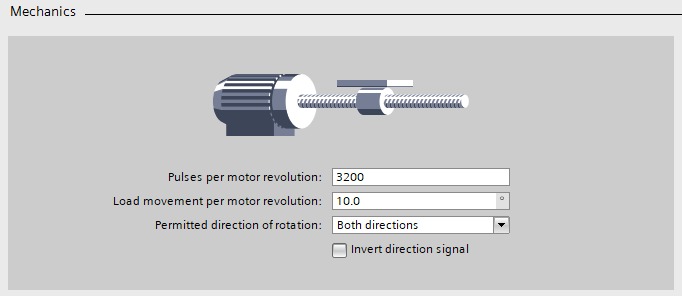
Şekil 3.22. Obje Seçimi

### Step Motor PLC Bağlantısı

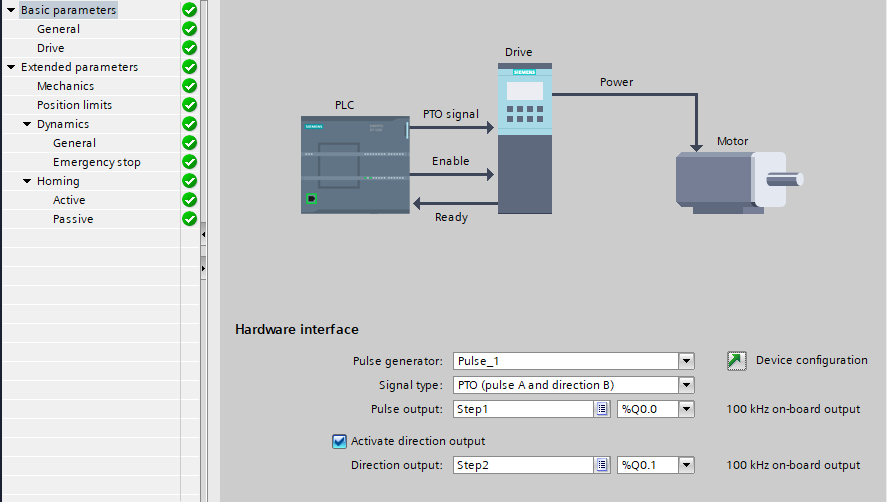
TIA Portal üzerinden Step motorun çekeceği akım, çözünürlü, step motorun dönüşü açısı belirlenmiştir. Frekans ile hız bilgisi kullanımı açısından sürücü PTO seçilmiştir.



Şekil 3.23. Genel Ayarlar



Şekil 3.24. Mekanik Ayarlar



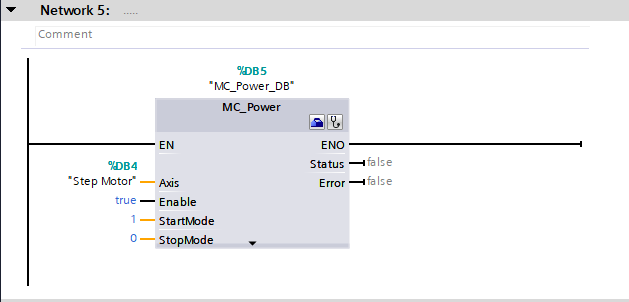
Şekil 3.25. TIA Portal Step Motor Ayarları

Step motor için TIA Portal üzerinden MC\_POWER\_DB, MC\_HOME\_DB ve MC\_Move\_Absolute\_DB kullanılmıştır.

### MC\_Power\_DB Ayarları

Hareket kontrol talimatı MC\_POWER ile gerçekleştirilmiştir. Bir ekseni etkinleştirmeyi ve o ekseni devre dışı bırakmak için kullanılmıştır. Kontrolör ve sürücü arasındaki döngüsel BUS iletişimi sağlamıştır. Axis, step motorun veri bloğudur. Teknoloji objesi ayarlarından step motorun etiketi seçilmiştir. StartMode 1 olacaktır. Çünkü parametre içerisinde başlangıçta bulunan konumlandırma ekseni etkinleştirildiğinde ve eksenin devre dışı bırakılmasına neden olan bir kesintinin başarılı bir şekilde onaylanmasından sonra eksen etkinleştirilmesine izin vermesi için yapılmıştır. Start butonu gibi yapılmıştır

StopMode 0 olacaktır. Çünkü acil stop butonu gibi düşünülmüştür. Herhangi bir acil, arıza gibi durumlarda kapanması için yapılmıştır.

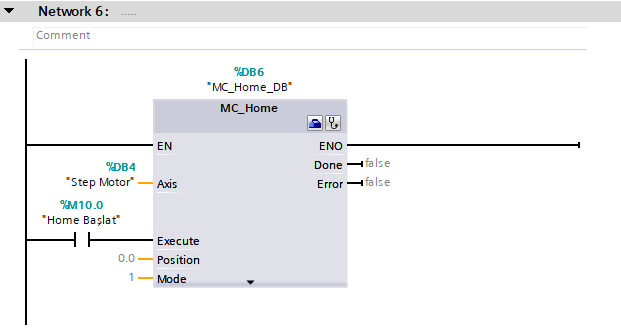


Şekil 3.26. MC\_Power Ladder Devresi

### MC\_Home\_DB Ayarları

Hareket Kontrol komutu "MC\_Home", eksen koordinatlarını gerçek, fiziksel sürüş konumuyla eşleştirmek için kullanılmıştır. Step motorun konumu ne olursa olsun aktifleştirilince pozisyonu 0 olmuştur. Step motorun başlangıç pozisyonuna dönmesini gerçekleştirmiştir.

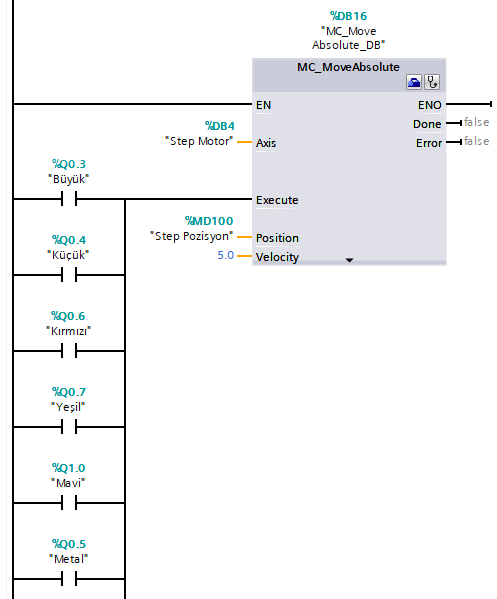
Axis etiketlemede step motor olarak seçilmiştir. MC\_HOME tek başına çalışamaz muhakkak buton eklenmesi gerekmektedir. Execute üzerinden “Home Başlat” butonu çalıştırıldığında ayarlanan pozisyonlar belirlenmiştir. Bu pozisyonlar ise ürünün rengi; mavi, kırmızı, yeşil olacak şekilde her birinin farklı derecelendirilmesi, ürünün metal olup olmaması ve ürünün boyutu küçük ve büyük olması şeklinde çıkış geldiğinde ayarlanan değerlere göre step motorun pozisyonu belirlenmiştir. Home üzerinde bulunan Mode 1 olarak alınmıştır. Çünkü geçerli eksen konumu, ayarlanan pozisyon parametresinin değeri ile kaydedilmiştir.



Şekil 3.27. MC\_Home Ladder Devresi

### MC\_MoveAbsolute\_DB Ayarları

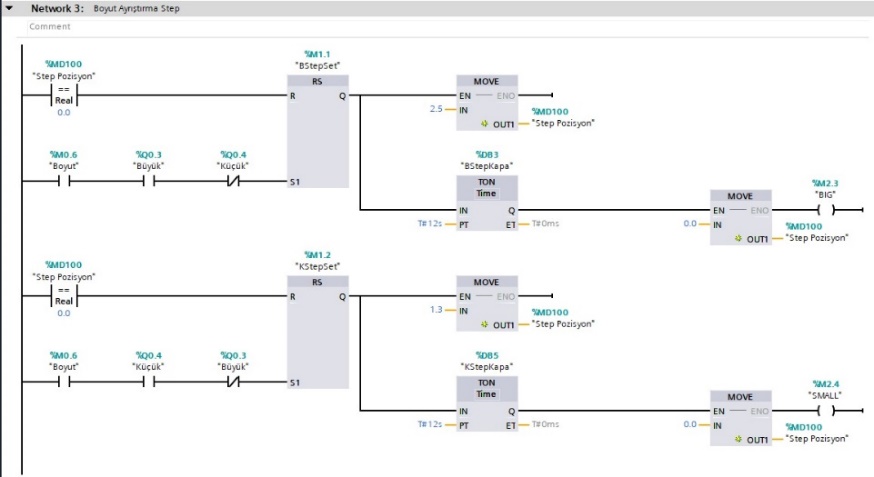
Axis etiketlemede step motor olarak seçilmiştir. Execute üzerinden “Home Başlat” butonu çalıştırıldığında ayarlanan pozisyonlar belirlenmiştir. Bu pozisyonlar ise ürünün rengi; mavi, kırmızı, yeşil olacak şekilde her birinin farklı derecelendirilmesi, ürünün metal olup olmaması ve ürünün boyutu küçük ve büyük olması şeklinde çıkış geldiğinde ayarlanan değerlere göre step motorun pozisyonunu belirlemiştir. Belirlenen pozisyona göre step motora bağlı ürün yönlendirme aparatının hızı 5 olarak seçilmiştir.



Şekil 3.28. MC\_MoveAbsolute Ladder Devresi

## Step Motor TIA Portal Bağlantısı

### Boyut Ayrıştırma



Şekil 3.29. Boyut Ayrıştırma Step Motor Blok Diyagramları

Network 3 boyut ayrıştırma step aşaması gösterilmiştir. Step pozisyon 0 yani home olacak şekilde ayarlanmıştır. Reset öncelikli bir blok yapılmıştır. Global memory’deki boyut start çalıştığında üstte bulunan sensör büyük ürünü algıladığı zaman çıkışa göre büyük çıkışı aktif sağlanmıştır. Küçük çıkışıyla kitleme yapılmıştır. Eğer aynı anda küçük ve büyük sinyali gelirse step motorun hareketi öngörülemeyeceğinden hasar söz konusu olmuştur. Büyük çıkış verildiğinde step motor pozisyonunu 90 derece açması için 2.5 değeri girilmiştir. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra BIG çıkışı verir.

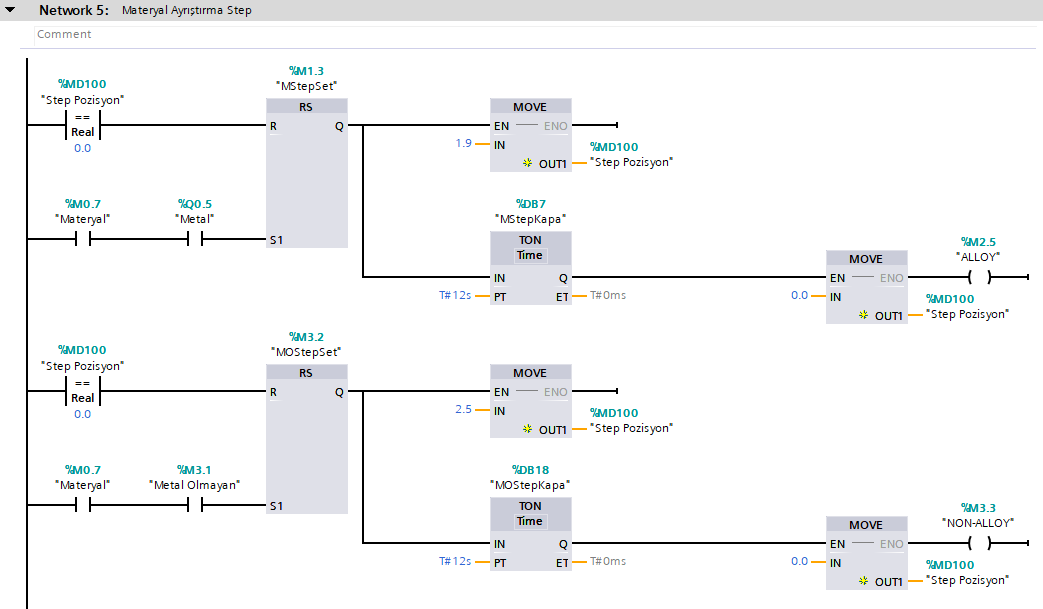
Global memory’deki boyut start çalıştığında alttaki sensör küçük ürünü algıladığı zaman çıkışa göre küçük çıkışını aktif yapmıştır. Büyük çıkışıyla kitleme yapılmıştır. Küçük çıkış verildiğinde step motor pozisyonunu açması için 1.3 değeri girilmiştir. Bu da yaklaşık olarak 45 derece olmuştur. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra SMALL çıkışı vermiştir.



Şekil 3.30. Küçük Ürün

Şekil 3.31. Büyük Ürün

### M**etal Ayrıştırma**



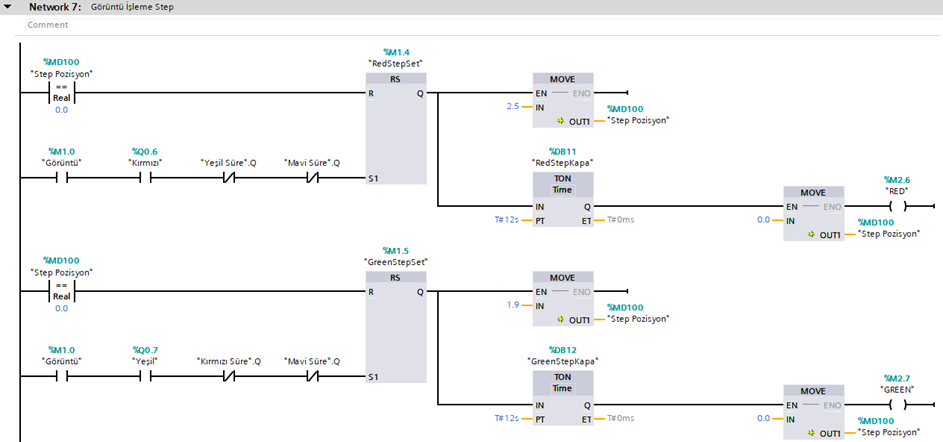
Şekil 3.32. Metal Ayrıştırma Step Motor Blok Diyagramları

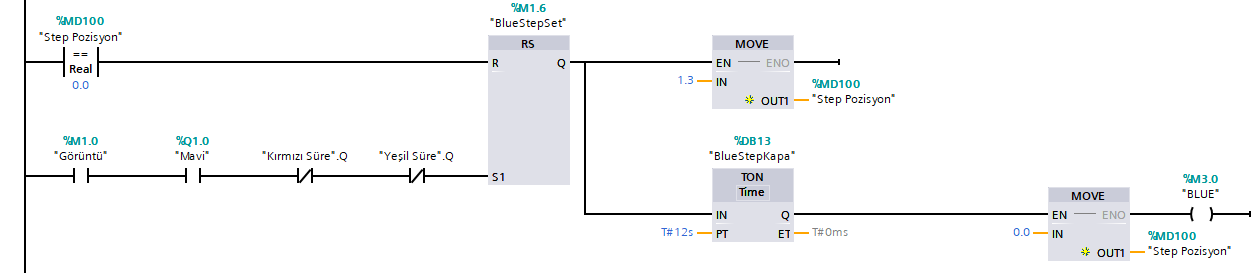
Global memory’deki materyal start çalıştığında metal sensör metal ürünü algıladığı zaman çıkışa göre metal çıkışı aktif çalışmıştır. Metal çıkış verildiğinde step motor pozisyonunu 1.90 değeri atayarak açması sağlanmıştır. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra ALLOY çıkışı vermiştir. Eğer ürün geçerken metal algılanmaz ise alttaki kızılötesi sensör devreye girerek step motor pozisyonuna 2.5 değeri atarak NON-ALLOY çıkışı elde edilmesi sağlanmış ve Move Absolute bloğuna gönderilmiştir.



Şekil 3.33. Metal Ürün

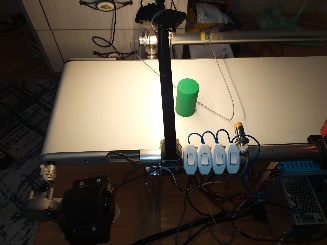
### Görüntü İşleme ile Ayrıştırma





Şekil 3.34. Görüntü İşleme ile Ayrıştırma Step Motor Blok Diyagramları

Global memory’de tanımlanan görüntü start butonu aktif olduğunda görüntü işlemede kullanılan kodlar çalıştırmıştır. Kırmızı ürün geldiğinde yeşil ve mavi ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Kırmızı algılandığında step motor pozisyonunu 90 derece açması için 2.5 değeri girilmiştir. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra RED çıkışı vermiştir. Global memory’de tanımlanan görüntü start butonu aktif olduğunda görüntü işlemede kullanılan kodlar çalıştırmıştır. Yeşil ürün geldiğinde kırmızı ve mavi ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Yeşil algılandığında step motor pozisyonunu açması için 1.90 değeri girilmiştir. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra GREEN çıkışı vermiştir. Global memory’de tanımlanan görüntü start butonu aktif olduğunda görüntü işlemede kullanılan kodlar çalıştırmıştır. Mavi ürün geldiğinde yeşil ve kırmızı ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Mavi algılandığında step motor pozisyonunu tamamen açması için 1.3 değeri girilmiştir. Ürünün konveyör banttan düşme süresi boyunca yani Step motor kapanma süresi 12 saniye belirlenmiştir. 12 saniyeden sonra BLUE çıkışı vermiştir.



Şekil 3.35. Kırmızı Ürün

Şekil 3.36. Mavi Ürün

Şekil 3.37. Yeşil Ürün

## Aparatlar

### Ürün Hizalama Aparatı

Ayrıştırılacak ürünlerin konveyör banttan hizalı bir şekilde ilerlemesi ve endüktif sensör tarafından rahatça algılanabilmesi için 1m ince demir çubuk ile tasarlanmış sağa odaklı bir düzenek yapılmıştır. Tasarımın amaçlarından diğeri ise ürünün konveyör bandın sağından gelecek şekilde iletilmesi düşünülerek konveyörün sonunda bulunan step motor ucuna demir çubuk yerleştirilmiştir.

Ürünler sensörler tarafından algılandığında belli bir açı ile açılacağı için ürünlerin sağ taraftan gelme tasarımı step motorun malzemeleri uygun kutulara düşürmesi için yapılan müdahaledir. Ürünün konveyör bantta gelebileceği en uzun çap hesaplanıp bırakılacak boşluk ve aparatın boyutu ayarlanmıştır. Oluşturulacak demir çubuk konveyör banda yerden 2 cm yüksekte kalacak şekilde ayarlanmıştır. Çünkü konveyör banda gelen ürünün denge noktası bulunmak istenmiştir. Bu sayede gelen silindir ürün dönerek hareketi tamamlamıştır.

Ayarlanan 1 metrelik demir çubuk istenilen şekle sokularak çubuğun uçları profil demirin uygun yerlerine sabitlenmiştir. Demir çubuğu profile sabitleme işi demir kelepçeler kullanılarak gerçekleştirilmiştir.



Şekil 3.38. Ürün Hizalama Aparatı

### Step Motora Bağlı Ürün Yönlendirme Aparatı

Konveyör banttan gelen ürünler, sensörler tarafından renk, boyut ve metal olup olmamasına göre her bir durumda gelen ürünlerin ayarlanan bir açıyla gerekli kutulara düşmesi amaçlanmaktadır. Kullanılan demir çubuğun step motorun ucuna sabitlenmesi için demir bir aparat kullanılmıştır. Konveyör banttaki step motor konveyör bantla 90° açı yapacak şekilde ve konveyör bandın bitimine 20cm kalacak şekilde konumlandırılmıştır.



Şekil 3.39. Step Motora Bağlı Ürün Yönlendirme Aparatı

### Konveyöre LED Bağlanması

Günün belirli zamanlarında yapılan denemeler sonucu karşılaşılan sorunun yetersiz ışık olduğu kanaatine varılmıştır. Sabah saatlerinde elde edilen veriler doğru iken, akşam saatlerine gelindiğinde kameranın lensine kötü görüntü kalitesinde veri gelmekteydi. Bu sorunu kameranın bağlandığı köprünün altına şerit LED eklenmesi ile çözülmesi amaçlanmıştır. Yapılan denemeler sonucu LED in kameranın gözlemlediği alanı çok iyi bir şekilde ışıklandırması sonucu günün herhangi bir saatinde görüntü kalitesinin bozulmadığı gözlemlenmiştir.

Şekil 3.40. Şerit LED Bağlantısı

BÖLÜM 4

# GÖRÜNTÜ İŞLEME

Son yıllarda makine öğrenimi araştırmacıları, yalnızca insan hayatını kolaylaştırmak için değil, aynı zamanda ekonominin belirli sektörlerinin üretkenliğini ve etkinliğini artırmak için çok çeşitli araştırmalar geliştirdi. Yapay zekâ ve nesne tanıma özel ilgi alanlarıdır. Taahhütleri, birçok şirketin ve faaliyetin yapay zekayı kendi iç süreçlerine entegre etmesini sağladı. Tıp uzmanları, görüntüleri kaydeden fiziksel muayeneler (X-ışınları ve ultrason taramaları gibi) sırasında tümörleri ve anormallikleri tespit etmek için kullanmaktadır. Havaalanı güvenlik personeli, yolcuların ve gözetimsiz bagajların şüpheli davranışlarını tespit etmek için kullanmaktadır. Sürücüsüz arabalar, bisikletler, diğer araçlar ve hatta yayalar gibi engellerin varlığını tespit etmek için de kullanılmaktadır. Bu projede ise konveyör bant üzerinde 3 farklı renkte gelecek olan silindirlerin hangi renkte olduğunu algılamak ve ona göre sinyal üretip PLC tarafına bilgi göndermek amaçlanmıştır.

Nesne tanıma olarak da bilinen görüntü işleme, yaşama ve çalışma şeklimizi değiştirmektedir. Ancak görüntü işleme uygulaması piyasaya çıkmadan önce, tamamen karmaşık bir süreçten geçmesi gerekiyor.

Öncelik olarak Google Colab’ın bu konuda sağlamış olduğu ekran kartları sayesinde silindirlerin etiketlenmiş fotoğrafları burada eğitilmek üzere çeşitli süreçlere girmektedir. Buna daha sonra da değinilecektir.

Görüntü tanıma (veya nesne tanıma), insan görsel korteksinin nasıl çalıştığını yeniden oluşturmak için çeşitli teknikler kullanmaktadır. Bu çeşitli yöntemler, sinir ağına girdi olan görüntüleri yakalar. Ardından, dikdörtgenin sınırladığı alanı işler ve görüntüdeki nesnenin konumunu ve kategorisini tanımlamak için etiketlemektedir.

Verileri etiketledikten sonra belirli bir süreçten geçmektedir. Burada YOLO V5 devreye girmektedir. Görüntülerin kalitesi ve boyutu büyük ölçüde farklılık gösteriyorsa, sistemin hepsini işlemesi meşakkatli olmaktadır. Ancak Google colab sayesinde bu sorun ortadan kalkmaktadır.

Modele hangi nesne sınıflarını tanıyacağını ve sınıflandıracağını söylemek gerekmektedir. Etiketleme işlemi görüntüde istenilen nesnenin dörtgen bir çevre içine alınıp kaydedilmesidir. Etkili bir eğitim için gereken minimum görüntü sayısı 200 civarı olduğundan mevcut bulunan kamera ile 10 FPS ve 60 saniye boyunca video çekimi yapılmıştır. Ardından Shotcut programı ile videoyu görüntülere bölerek her bir silindir için 750 ye yakın fotoğraf elde edilmiştir. Bu fotoğraflar Roboflow sitesinde etiketleme işlemine tabi tutulduktan sonra çeşitli eklemeler yapılarak fotoğraf sayısını çoğaltılabilir. Bu çoğaltma işlemi fotoğrafı ters çevirme, genişletme ve boyut değiştirme şeklinde olduğundan her bir parça için 1000 in üzerinde fotoğraf elde edilerek doğruluk oranının yüksek seviyelerde olması hedeflenmektedir. Elde edilen yüzlerce fotoğraf Google colab de görüntü işlemeye sokulur. Burada ilk olarak Roboflow sitesindeki fotoğrafların çekilmesi sağlanmıştır. Ardından environment kurularak modelin eğitilmesi Python da train.py kullanılarak başlandı. Epoch diye adlandırılan test sayısı ne kadar artarsa elde edilen model de bir o kadar doğru olacaktır. Dosyalar sekmesinde bulunan best.pt dosyası proje için önemli olan eğitim bilgilerini tutar. Bu sistem sayesinde kameranın hangi açısına gelirse gelsin doğru bir şekilde ilerleyen cisim her türlü algılanacaktır. Eğitime sokulmayan herhangi bir durumdaki resmi de kendi içerisinde belirlediği doğruluk payıyla kırmızı, mavi veya yeşil mi olduğuna karar verip bir çıktı üretmesi ve bu çıktının KepServerEX yoluyla PLC ye gönderilmesi amaçlandı. KepServerEX uygulamasına daha sonra değinilecektir.

## Derin Öğrenme Nedir?

Derin öğrenme, büyük ölçüde insan beyninin nasıl çalıştığını modelleyen algoritmalar olan sinir ağlarının katmanları tarafından desteklenir. Büyük miktarda veri için eğitim, bir sinir ağındaki nöronlardan oluşur. Sonuç, eğitimden sonra yeni verileri işleyen derin bir öğrenme modelidir. Derin öğrenme modelleri, birden çok veri kaynağından bilgi alır ve bu verileri gerçek zamanlı olarak analiz eder. Hiçbir insan müdahalesi gerekli değildir. Derin öğrenmede, grafik işleme birimleri (GPU'lar), aynı anda birden çok hesaplamayı işleyebildikleri için model eğitimi için optimize edilmiştir. Derin öğrenmede, öğrenme denetimli, denetimsiz veya yarı denetimli olabilir.

## Denetimli Öğrenme

Bu, öngörülemeyen verileri tahmin etmeye yardımcı olmak için eğitim verilerinden öğrenen bir algoritmadır. Denetimli makine öğrenimi, istenen sonuçları elde etmek için eğitim veri kümelerini kullanır.

## Denetimsiz Öğrenme

Kullanıcının modeli kontrol etmesini gerektirmeyen bir makine öğrenimi tekniğidir. Bunun yerine, modeller daha önce keşfedilmemiş kalıpları ve bilgileri keşfetmek için kendi başlarına çalışabilir.

## Takviyeli Öğrenme

Yazılım aracılarının bir ortamda nasıl davranması gerektiği ile ilgilenen bir makine öğrenme tekniği olarak tanımlanır. Bu, kümülatif ödülün bir kısmını en üst düzeye çıkarmaya yardımcı olan derin bir öğrenme tekniğinin parçasıdır. Sinir ağlarında, girdi katmanı ile çıktı katmanı arasında kalan katmanlar gizli katmanlardır. Sinir ağları ve derin öğrenme arasındaki farklar şunlardır: Basit bir sinir ağı bir veya iki gizli katmana sahip olabilirken, derin öğrenme ağı onlarca veya yüzlerce katmana sahip olabilir. Farklı katmanların ve düğümlerin sayısını artırmak, ağın doğruluğunu artıracaktır. Bununla birlikte, daha fazla katman, modelin daha fazla parametre ve hesaplama kaynağı gerektirdiği anlamına da gelir. Bunu da daha önce bahsedilen YOLO V5 yapmaktadır.

## YOLO v5 Nedir?

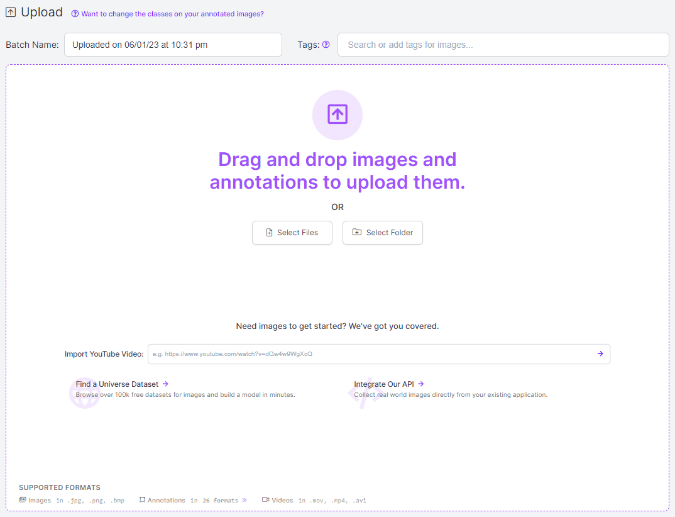
Son yıllarda nesne tanıma alanında çığır açan YOLO (You Only Look Once) algoritması hızlı olmasından kaynaklı sıklıkla tercih ediliyor. YOLO'dan daha iyi tahminciler oldu ve olmaya devam ediyor, ancak asıl sorun yavaş olmalarıdır. YOLO'yu diğer algoritmalardan ayıran en önemli özellik gerçek zamanlı nesne algılamadır. Bu da projedeki en çarpıcı detay olarak ön plana çıkıyor. YOLO, zaman ve doğruluk açısından rakiplerinin çok ilerisindedir. Ayrıca, gerektiği gibi doğruluk ve hız arasında kolayca geçiş yapılabilir. Öncelikle diğer algoritmaların neden yavaş olduğuna değinmek gerekir. Örneğin, R-CNN gibi alan tabanlı nesne algılama algoritmaları, önce nesnelerin bulunabileceği alanları belirler ve burada ayrı evrişimli sinir ağı (CNN) sınıflandırıcıları çalıştırır. Bu yöntem iyi sonuçlar verir, ancak görüntü iki ayrı işlemden geçerek görüntü üzerinde daha fazla işlem yapılmasına ve daha düşük FPS'ye (saniyedeki kare sayısı) neden olur. Bu da istenilen bir sonuç değildir. Kameradan gelen verinin hızlı bir şekilde işlenmesi ve KepServerEX ile PLC ye aktarılması arasında kısa bir süre olması bu sayede step motora en kısa sürede bilgi gönderilmesi beklenmektedir. Her ızgara bir cismin bir bölgede olup olmadığını, merkezinin o bölgede olup olmadığını, boyunu, yüksekliğini ve hangi sınıfta olduğunu gösterir. Kameradan alınan resim farklı karelere bölündüğünde karelerin içinde eğer istenilen renkte cisim varsa ona ait bir vektör oluşturur. Bu nedenle YOLO, her ızgara için ayrı tahmin vektörleri oluşturur. Çıktı vektörüne göre, her ızgara yalnızca bir nesne tanımlayabilir. Örneğin, yalnızca 3x3'lük bir ızgara kullanırsa, 9 nesneyi tahmin edilebilir.

## Python ile Derin Öğrenme Uygulaması

Her üç renk için 74 saniyelik videolar çekilmelidir. Çekilen videolardaki her bir frame için Shotcut programı kullanılarak video, fotoğraflara dönüştürülür. Bu sayede tek tek fotoğraf çekilmemiş olmaktadır. Her renk için 700 adet fotoğraf elde edildikten sonra boş olan fotoğraflar silinir. Kalan 1050 adet fotoğraf Roboflow [6] adlı siteye yüklenmektedir.

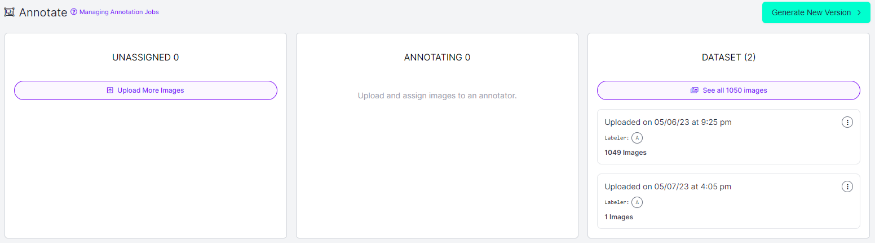
Roboflow [6] sitesinin çalışma mantığı şu şekildedir;

* Fotoğraflar siteye yüklenmiştir.



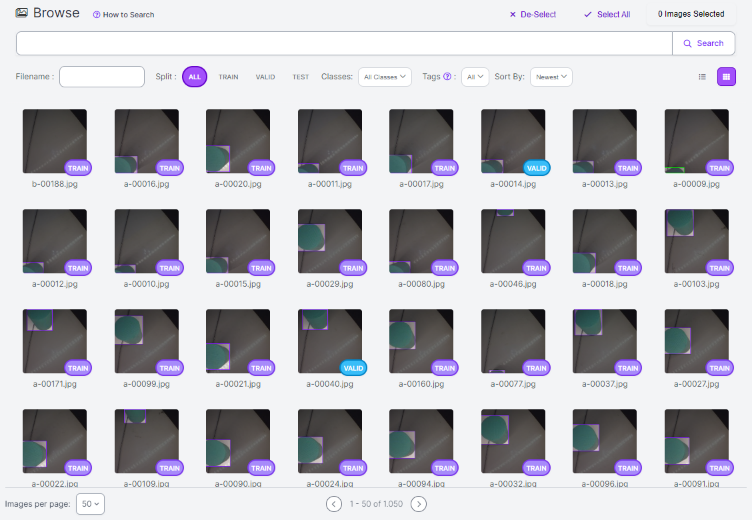
Şekil 4.1. Roboflow [6] Fotoğraf Yüklenmesi

* Fotoğraflardaki renkli ürünlerin etiketleme işlemi tek tek yapılmıştır.



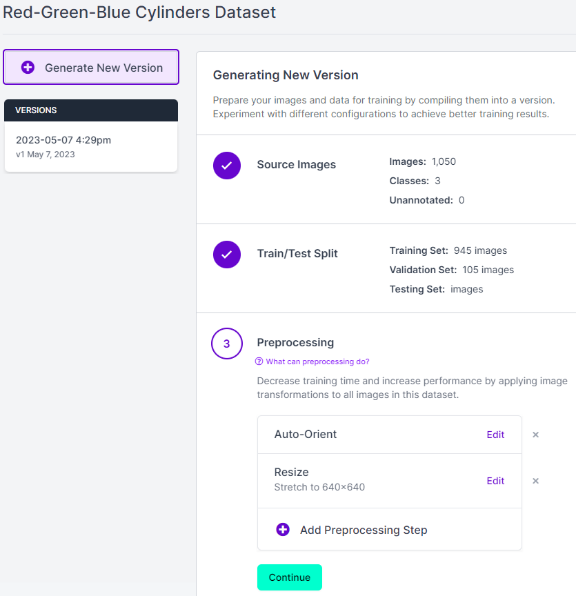
Şekil 4.2. Fotoğraf Etiketleme

* Etiketlenen fotoğrafların %10 u geçerli %90 ı eğitilecek veri olarak seçilmiştir.



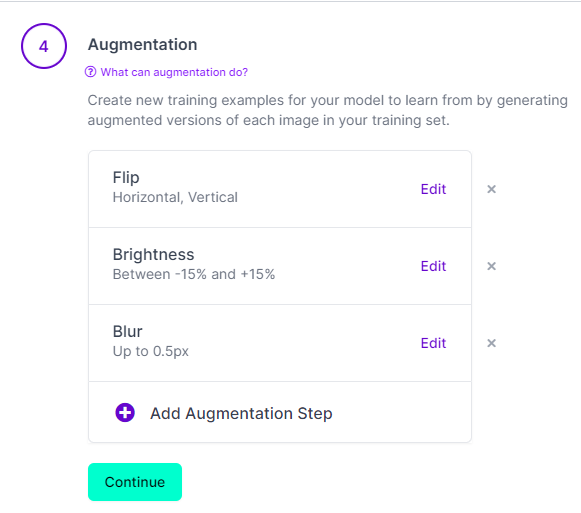
Şekil 4.3. Fotoğrafların Eğitimi

* Fotoğrafların düzgün işlenebilmesi için kare formatına çevrilmiştir.

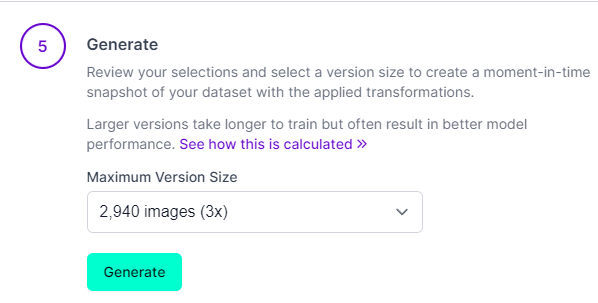


Şekil 4.4. Çözünürlük Ayarlama

* Augmentation sekmesinde eğitilecek 950 adet fotoğraf çeşitlendirme işlemine tabi tutulmuştur. Burada örneğin parlaklık, bulanıklık, ters-düz olacak şekilde ekleme yapılmıştır.

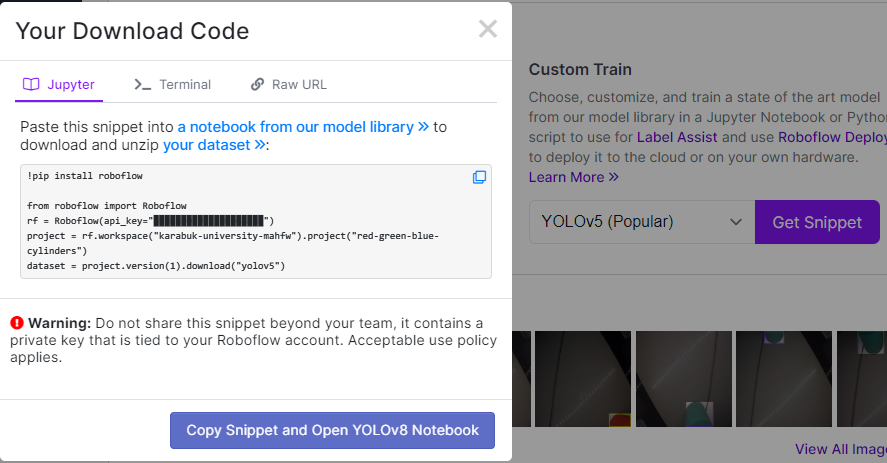


Şekil 4.5. Resim Farklılaştırma İşlemi

* Elde edilen görüntüler sitenin ücretsiz kısmında 3 katına kadar çoğaltılarak kesinlik sağlanmıştır.

Şekil 4.6. Resim Çoğaltma İşlemi

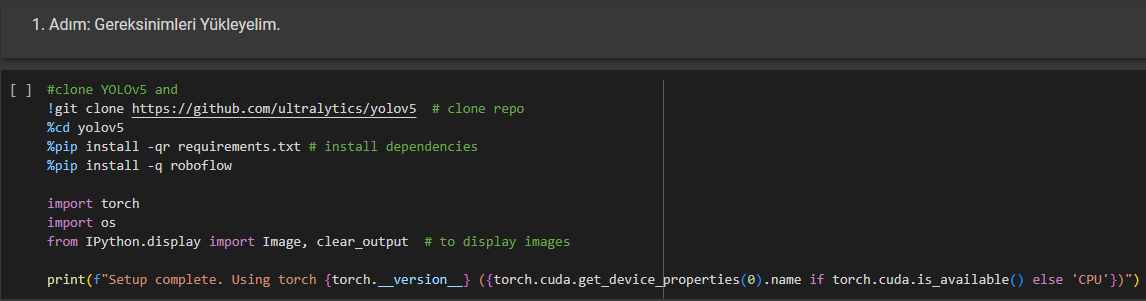
* Hazırlanan veri seti YOLOv5 seçilerek get snippet tuşuna basılmış ve indirmek için gerekli olan Python kodunu oluşturmuştur.



Şekil 4.7. Python İndirme Kodu Oluşturma

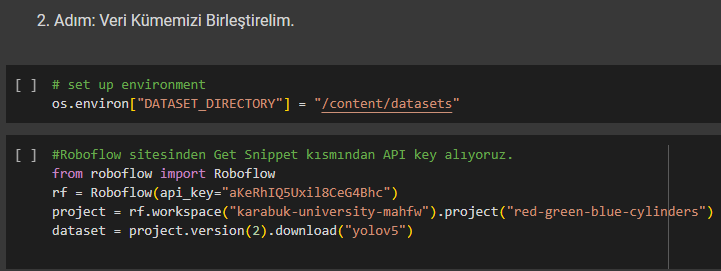
Google Colab aracılığıyla Google’ın bize sağlamış olduğu güçlü ekran kartları sayesinde işlem süresi azaltılmıştır.

* İlk olarak ultralytics [7] in yolov5 dosyaları github’dan çekilmiştir.
* Gerekli kütüphaneler eklendikten sonra CUDA sürücü olan ekran kartından işlem yapılmıştır.



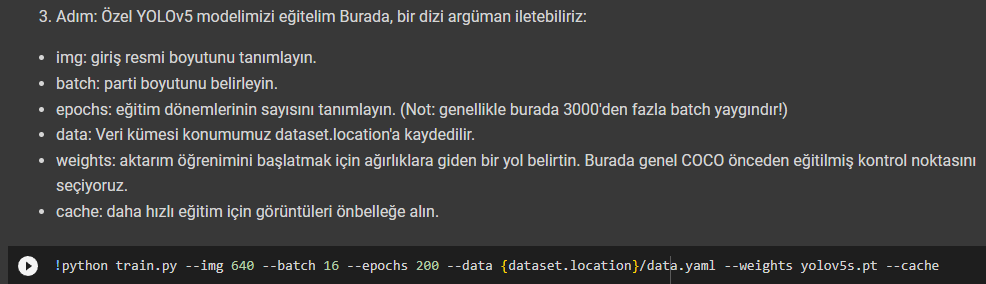
Şekil 4.8. YOLOv5 Dosyaları Çekilmesi

* Ortam oluşturulduktan sonra Roboflow [6] sitesinden Get Snippet kısmından API key çalıştırılmıştır.



Şekil 4.9. API key Çalıştırılarak Roboflow [6] Veriseti Çekilmesi

* YOLOv5 kullanarak Python kütüphanesinde bulunan train.py kodu çalıştırılmıştır.



Şekil 4.10. Eğitim

Kişisel bilgisayarlarda 24 saatte yapılacak işi 4 saatte yapılabildiği görüldükten sonra best.pt model dosyası oluşturulmuştur.

## Görüntü İşleme TIA Portal Bağlantısı

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 4.11. Görüntü İşleme TIA Portal Blokları

Global memory’de tanımlanan görüntü start butonu aktif olduğunda görüntü işlemede kullanılan kodlar kullanılmıştır. Kırmızı ürün geldiğinde yeşil ve mavi ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Kırmızı ürün geldiğinde 5 saniye boyunca PLC’ye kırmızı çıkışını vermiştir. Yeşil ürün geldiğinde kırmızı ve mavi ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Yeşil ürün geldiğinde 5 saniye boyunca PLC’ye yeşil çıkışını vermiştir. Mavi ürün geldiğinde kırmızı ve yeşil ürünlerin sürelerinin çıkışlarıyla kitleme yapılmıştır. Mavi ürün geldiğinde 5 saniye boyunca PLC’ye mavi çıkışını vermiştir. Boyut veya materyal start basıldığında reset çalışacağından görüntü işleme çalışması sağlanmamıştır.

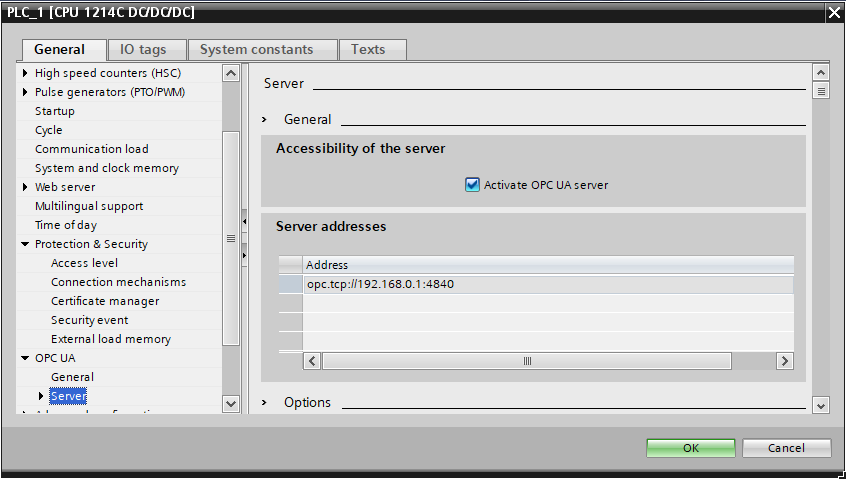
silindir, iç mekan, vazo, duvar içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 4.12. 3 Farklı Renkte Ürünler

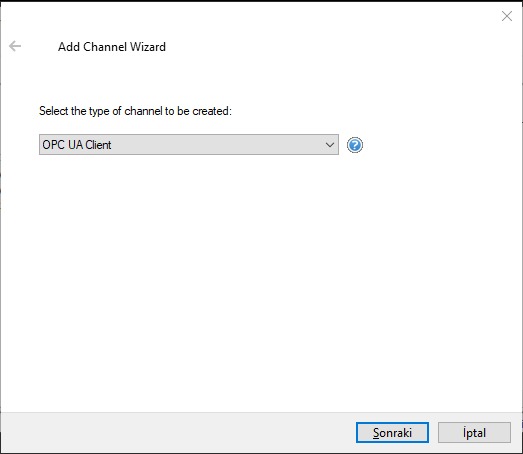
## KepServerEX Kullanarak PLC Haberleşmesi

KepServerEX uygulaması Python’dan gelecek olan görüntü işleme esnasında elde edilen ürünün bilgisi gönderildi ve PLC’ye yazmak için arada bir köprü niteliği taşımaktadır. Bu köprü PLC içinde bulunan OPC UA Server aracılığıyla elde edilmiştir.



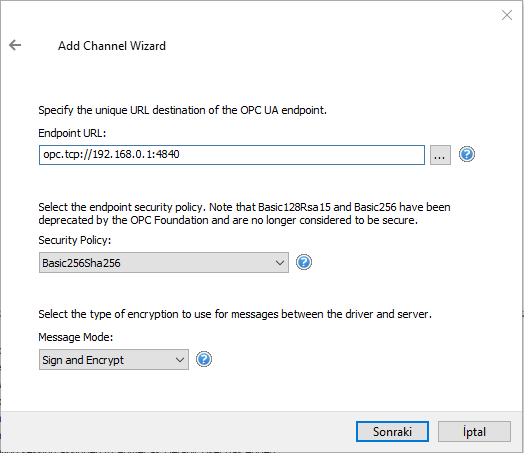
Şekil 4.13. PLC OPC UA Server Kurulması

Bu sekmede bulunan “opc.tcp://192.168.0.1:4840” adresi oluşturulmuştur. KepServerEX üzerinde bulunan Connectivity kısmından New Channel seçilerek sihirbazdan OPC UA Client seçilmiştir.



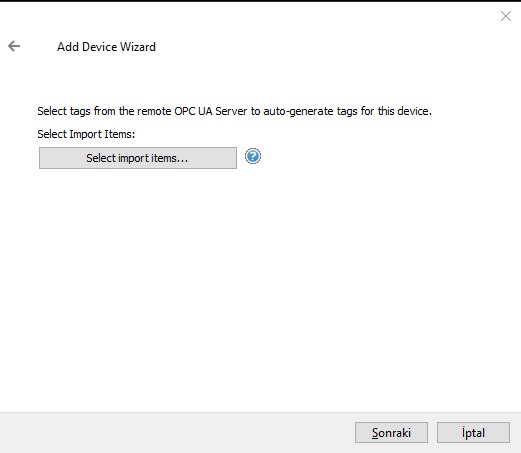
Şekil 4.14. KepServerEX Yeni Kanal Oluşturulması

PLC Üzerinde ayarlanan OPC UA Client adresi tam olarak girilerek kanal oluşturulmuştur.



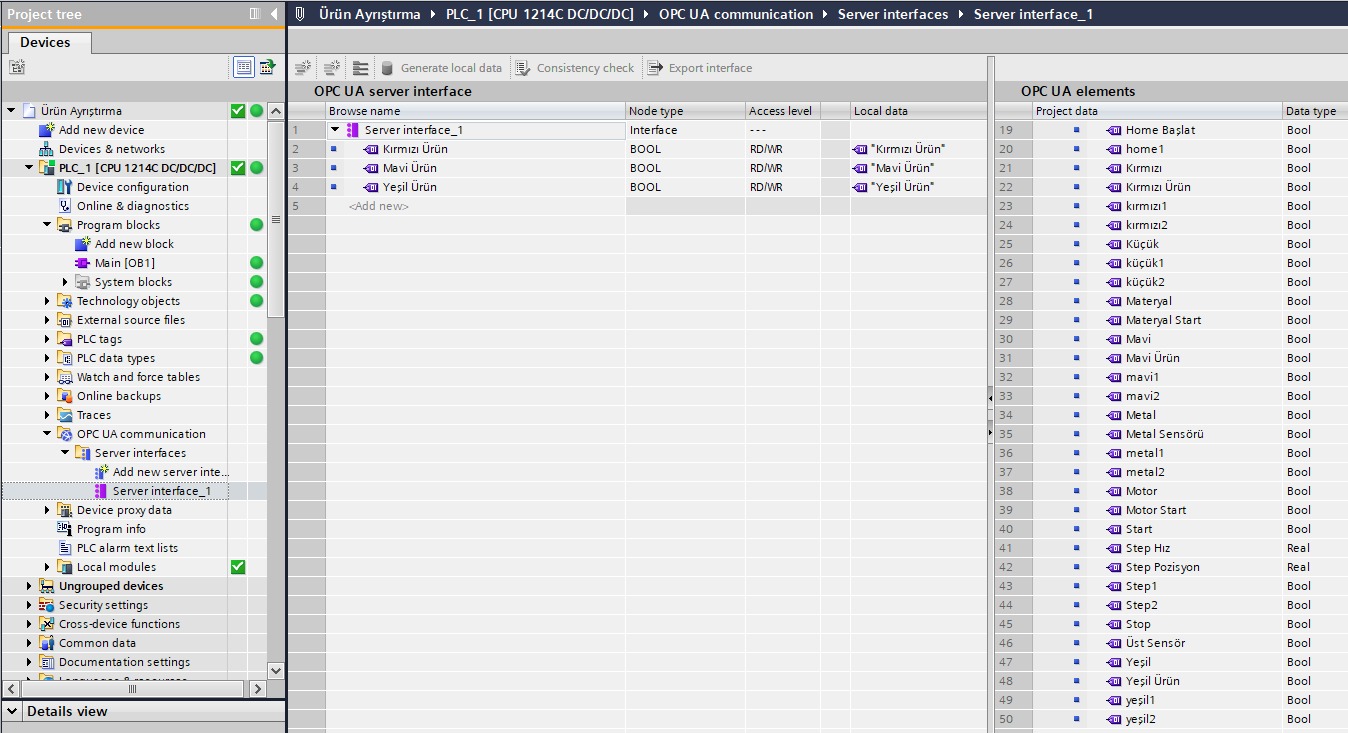
Şekil 4.15. Endpoint URL Ekleme

Ardından oluşturduğumuz kanala PLC’den elde edilen Server Interface in bulunduğu kısım KepServerEX’e yüklenmiştir.



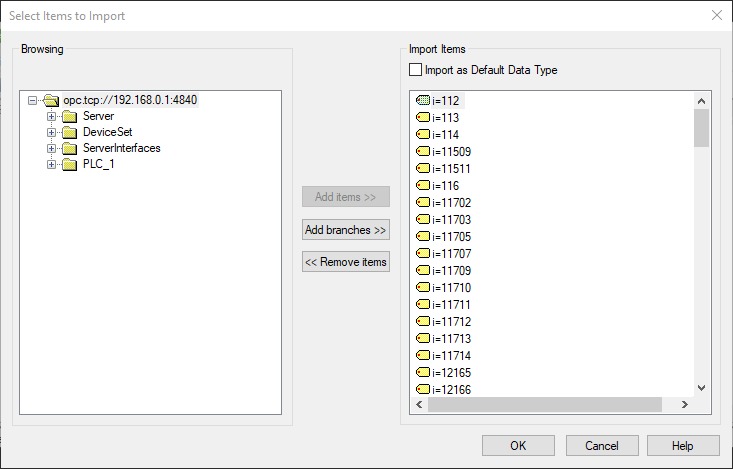
Şekil 4.16. PLC den Gelen Verilerin Import Edilmesi

Import edilecek verilerin PLC den gelen kısmındaki asıl önemli olan Server Interface içinde bulunan kırmızı, yeşil ve mavi tagları kontrol edilmesi amaçlanmıştır. OPC UA elements kısmından ilgili taglar OPC UA server interface içerisine çekilerek KepServerEX üzerinde görüntülenmesi sağlanmıştır. Bu işlemden sonra Device Wizard kısmına dönülmüştür.



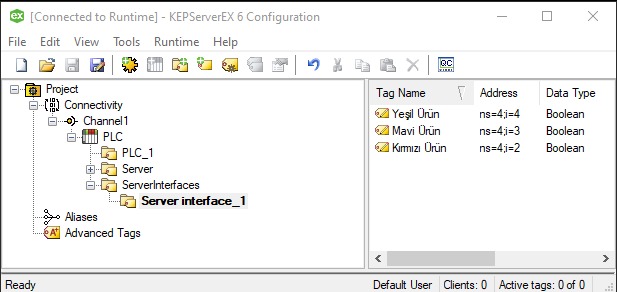
Şekil 4.17. OPC UA Server Interface

Sonraki sekmenin açılabilmesi için PLC’de online olmak gerektiğinden bilgisayarla bağlanarak online olunmuştur. Bu sayede import edilecek server verileri KepServerEX içine eklenmesi hedeflenmiştir. “opc.tcp://192.168.0.1:4840” seçilerek add branches butonuna basılarak veriler aktarılmıştır.



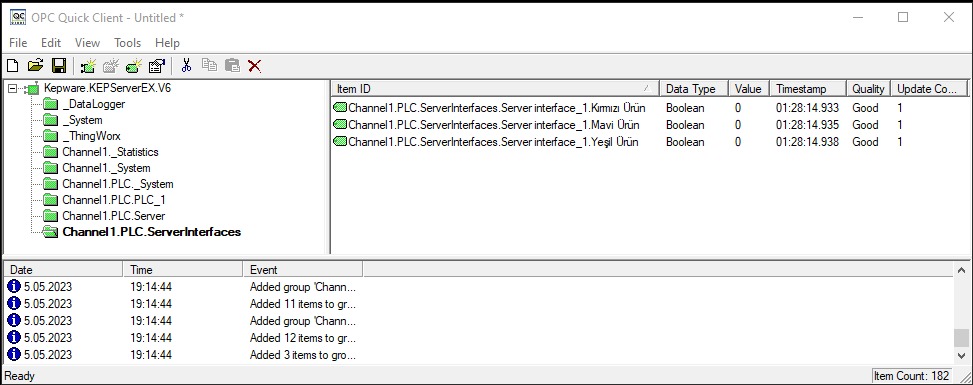
Şekil 4.18. Server’dan Import Edilen Veriler

PLC üzerinde ayarladığımız taglar KepServerEX üzerinde gözükmektedir. Burada bizi ilgilendiren tag’lara Server Interfaces sekmesinden ulaşabilir durumda olmuştur. Yeşil, mavi ve kırmızı ürünün kendilerine ait özel adresler oluşmuştur. Bu adresler Python da yazılacak olan kod için önemlidir. Veri tipinin Bool olması belirtilen şartlarda olmuştur. Belirtilen durumda PLC üzerinde ayarlanan ve Server Interface üzerine eklenen tag’ların veri tipleri değişkenlik göstermiştir.



Şekil 4.19. Server Interface Üzerindeki Verilerin Görüntülenmesi

İstenilen taglar eklendikten sonra yukarıda bulunan Quick Client butonuna basılmıştır. Açılan OPC Quick Client üzerinde KepServerEX in üzerinde oluşan kanal yoluyla bağlantısının sağlandığı ve Server Interface üzerinde bulunan taglar PLC’deki ladder devresinde online durumunda iken 0 durumunda aynı anda Client üzerinde de 0 değeri görülmüştür. KepServerEX üzerinde Value değeri 1 yapıldığında PLC üzerinde ilgili input’un 1 değerini aldığı gözlemlenmiştir. Burada bulunan Quality verisinin Good olması beklenmiştir. Unknown veya Bad durumlarında PLC ile veri alışverişi sağlanılmayacaktır. Bağlantının düzgün olduğu teyit edilmiştir.



Şekil 4.20. OPC Quick Client

BÖLÜM 5

# SİSTEMİN PANOLANMASI VE KABLO DÜZENİ

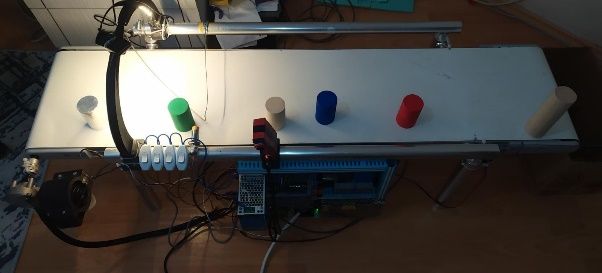
Fiziksel olarak sistem tamamlandıktan sonra kablo kalabalığından kurtulmak ve sistemde kullanılan S7-1200 PLC, 2 adet röle, step motor sürücüsünü, 12 V güç kaynağı, 24 V güç kaynağı ve 24 V güç kaynağından kablo ile çekilen 9 klemens + bara oluşturacak şekilde ve 8 adet klemens – bara oluşturacak şekilde düzenleme sağlanmıştır.

Pano şeklini alacak tahta parçasının üzerine ray sabitlenmiştir. Raylara sırasıyla sistemde kullanılan S7-1200 PLC, 2 adet röle, step motor sürücüsünü, 24 V güç kaynağı ve 24 V güç kaynağından kablo ile çekilen 9 klemens + bara oluşturacak şekilde ve 8 adet klemens – bar raya yerleştirilmiştir. Rayın 5 cm uzaklığında içi delikli 3 adet kablo kanalı yerleştirilmiştir. Sistemdeki kablolar kablo kanallarından geçirilerek kablo kalabalığından kurtarılmıştır.

Elektrik kabloları, elektronik donanım, makine, kablo içeren bir resim

Açıklama otomatik olarak oluşturuldu12 V’luk güç kaynağında ray sabitleyicisi olmadığı için köşelerinde bulunan vida yerlerinden tahtaya sabitlenmiştir. Step motor sürücüsün altında fan olduğundan dolayı 2 cm yükseklikte olacak şekilde tahtaya sabitlenmiştir.

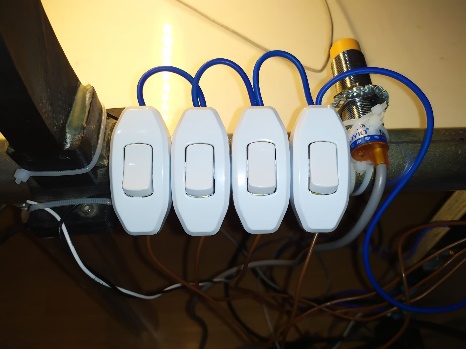
Şekil 5.1. Sistem Panosu



Şekil 5.2. Genel Sistem Görünümü

## **Butonlarla Sistemin Kontrolü**

Butonların girişi 24 volt – baradan alınmıştır. Her bir buton birbiriyle köprülenmiştir. Bu sayede butonların her bir girişi – 24 V ile beslenmiştir. Butonların çıkışları ise PLC’nın belirlendiği girişlere bağlanmıştır. Motor start için PLC’nin 3.çıkışına, boyut start için PLC’nin 4.çıkışına, materyal start için PLC’nin 5.çıkışına ve görüntü start için PLC’nin 6.çıkışına bağlanılmıştır. Bu bağlantılar sayesinde fiziksel bir buton oluşturularak kolaylıkla sistemin çalışması sağlanılmıştır.



Şekil 5.3. Fiziksel Butonlar

## Ürünlerin Ayrıldığı Bölme

Endüktif metal sensörü, MZ80 kızıl ötesi mesafe sensörü ve görüntü işleme sonrasından gelen ürünler step motora bağlı ürün yönlendirme aparatı belirlenen derecelere göre ürünlerin bölgelere düşmesi sağlamıştır. Görüntü işleme sonrası düşen ürünlerde kırmızı ürün sol bölmeye, yeşil ürün orta bölmeye mavi ürü sağ bölmeye; boyut ayrıştırma sonrasında büyük ürün sol bölmeye, küçük ürün sağ bölmeye; metal ayrıştırma sonrasında metal olmayan ürün sol bölmeye, metal ürün ise orta bölmeye düşmesi amaçlanmıştır.



Şekil 5.4. Ürünlerin Ayrıldığı Bölme

BÖLÜM 6

# SONUÇ VE ÖNERİLER

Üç farklı gruptan oluşan python tabanlı görüntü işleme kullanılarak yapılan ürün ayrıştırma projesinde ilk olarak kullanılan sensörlerin 5 V olduğu görüldü. Bu sebeple ayarlanabilir güç kaynağı kullanıldı. Fakat bu sistemin karmaşık ve kararsız çalıştığı kanaatine varıldı. Tahrip olan ve doğru sonuç vermeyen sensörler yerine, aynı zamanda PLC nin beslemesini aldığı 24 V güç kaynağı ile çalışabilecek sensörler temin edildi. Bu şekilde sistemin daha sadeleşmesi sağlandı. Kullanılan sensörlerin NPN ve PNP tipi olması sebebiyle PLC üzerinde bir tip kullanılması gerektiğinden PNP türü, NPN türüne dönüştürülmesi için röle kullanılması gerektiği anlaşıldı. Motoru, PLC ile kontrol edebilmek adına röle ile anahtarlama yapıldı. Bu sayede motorun istenilen durumda enerjilendirilmesi ve yine manuel bir şekilde enerjinin kesilmesi sağlandı. Görüntü işlemede başlangıç olarak BGR to HSV yöntemi kullanılarak HUE değişimine bağlı renk algılama çalışması yapıldı ancak ortamdaki ışığın değiştiği durumda veya konveyör üzerinde en ufak bir renk değişiminde sistemin bunu algılaması ve ortada ürün yokken step motora hareket verdiği görüldü. Bu yöntemin kararsız ve sorunlu çalıştığı anlaşıldı. Derin öğrenme yoluyla konveyör üzerinden geçen ürünlerin YOLO v5 modeli ile %98 lere varan doğruluk oranıyla çalıştığı anlaşıldı. Günün farklı zamanlarında yapılan denemelerde dışarıdaki ışık değişiminden dolayı görüntü işlemeye negatif etki yaptığı görüldü. Siyah köprü altına LED yapıştırılması ile kameranın baktığı alana dışarıdan gelen ışık değişse bile sıkıntısız çalıştığı görüldü. 3 farklı grupta ürün ayrıştırma yapıldığından ve TIA Portal projesindeki blokların birbirleriyle çakışmaması adına ve sistemin karmaşıklığının giderilmesi için gruplara ayrılan bloklar fiziksel buton grubu ile çalıştırılması sağlanmıştır. Ayrıca start butonu eklenerek konveyör bantın çalıştırılması sağlanmıştır. Örneğin boyut ayrıştırma yapılmak istendiğinde sistem çalışması adına start butonuna basılır ve boyut ayrıştırma butonuna basılarak görüntü işleme ve materyal ayrıştırma kısmının devre dışı bırakılması sağlandı. Sistemin sorunsuz bir şekilde çalıştığı görüldü. Diğer konveyör bant sistemlerindeki ürün ayrıştırma projelerinden farklı olarak derin öğrenme kullanılması tez adına önemli bir gelişme olmuştur. Ayrıca daha büyük konveyör bantlar ile aynı sistemin daha büyük cisimler taşıyarak geliştirilmesi mümkündür. Gelişime çok açık olan bu proje birçok alanda fayda sağlayacaktır. Sonuç olarak farklı özelliklere sahip cisimler, aynı konveyör bant üzerinde ayrıştırılabilir ve bu sistem insan gücüne ihtiyaç duyulmadan seri bir şekilde hata payını en aza indirgenmesi sağlanmıştır.

# KAYNAKLAR

# Kaynakça

|  |  |
| --- | --- |
| [1] | M.-O. Y. Sistemleri, MS-OTOMASYON Yazılım Sistemleri, [Çevrimiçi]. Available: http://www.ms-otomasyon.com/siemens-s7-400/p/66. [Erişildi: 12 12 2022]. |
| [2] | H. Altınkaya, PDF, Hüseyin Altınkaya. |
| [3] | V. H. Technology, «Valx,» [Çevrimiçi]. Available: http://valx.com.tr/urun/valx-vma-1210-12v-10a-120w-metal-kasa-adaptor/. [Erişildi: 12 12 2022]. |
| [4] | V. H. Technology, «Valx,» [Çevrimiçi]. Available: http://valx.com.tr/urun/valx-vma-2410-24v-10a-metal-kasa-adaptor/. [Erişildi: 12 12 2022]. |
| [5] | MuratTas, 12 12 2022. [Çevrimiçi]. Available: https://www.mekinfo.net/s7-1200-plc-ders-iv-npn-pnp-baglanti/. |
| [6] | Roboflow, Des Moines, 500 Locust St, United States: Roboflow, Inc., 2023. |
| [7] | ultralytics, United States of America: ultralytics, Inc., 2023. |
| [8] | H. Demirel, Ayarlanabilir Güç Kaynağı Tasarımı. |

## EKLER

## models

best.pt #Eğitilen model

## Config.py

import torch

DEVICE = 'cuda' if torch.cuda.is\_available() else 'cpu'

MODEL\_PATH = "./models/best2.pt"

CAMERA\_SOURCE = 0

INPUT\_SHAPE = 640

PLC\_URL = "opc.tcp://192.168.0.1:4840"

## DetectionMain.py

import Config

from ProductDetector import \*

from PLCUtils import \*

def main():

    camera\_source = Config.CAMERA\_SOURCE

    model\_path = Config.MODEL\_PATH

    device = Config.DEVICE

    input\_shape = (Config.INPUT\_SHAPE, Config.INPUT\_SHAPE)

    plc\_url = Config.PLC\_URL

    video\_cap = cv2.VideoCapture(camera\_source)

    if not video\_cap.isOpened():

        print('Unable to open camera from source :', camera\_source)

        return

    client = client\_connection(url=plc\_url)

    if client == None:

        #return

        pass

    detector = ProductDetector(model\_name=model\_path, device=device, conf\_th=0.90)

    while True:

        ret, frame = video\_cap.read()

        if not ret:

            break

        frame = cv2.resize(frame, input\_shape)

        #Detection

        start\_time = time()

        results = detector.predict(frame)

        end\_time = time()

        #Görselleştirme

        fps = 1 / (end\_time - start\_time)

        obj\_class, frame = detector.plot\_boxes(results, frame)

        cv2.putText(frame, f'FPS: {int(fps)}', (20,70), cv2.FONT\_HERSHEY\_SIMPLEX, 1.5, (0,255,0), 2) # print(f"her saniye frame yaz : {fps}")

        cv2.imshow("YOLOv5 Detection", frame)

        if obj\_class == "Red":

            write\_value\_bool(client, 'ns=4;i=5', True)

            write\_value\_bool(client, 'ns=4;i=5', False)

        elif obj\_class == "Green":

            write\_value\_bool(client, 'ns=4;i=6', True)

            write\_value\_bool(client, 'ns=4;i=6', False)

        elif obj\_class == "Blue":

            write\_value\_bool(client, 'ns=4;i=6', True)

            write\_value\_bool(client, 'ns=4;i=6', False)

        else:

            continue

        if cv2.waitKey(1) == 27:

            break

    video\_cap.release()

    cv2.destroyAllWindows()

    if device == "cuda":

        detector.clear\_gpu\_cache()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

## PLCUtils.py

from opcua import Client, ua

def read\_input\_value(client, node\_id):

    client\_node = client.get\_node(node\_id)  # get node

    client\_node\_value = client\_node.get\_value()  # read node value

    print("Value of : " + str(client\_node) + ' : ' + str(client\_node\_value))

def write\_value\_int(client, node\_id, value):

    client\_node = client.get\_node(node\_id)  # get node

    client\_node\_value = value

    client\_node\_dv = ua.DataValue(ua.Variant(client\_node\_value, ua.VariantType.Int16))

    client\_node.set\_value(client\_node\_dv)

    print("Value of : " + str(client\_node) + ' : ' + str(client\_node\_value))

def write\_value\_bool(client, node\_id, value):

    client\_node = client.get\_node(node\_id)  # get node

    client\_node\_value = value

    client\_node\_dv = ua.DataValue(ua.Variant(client\_node\_value, ua.VariantType.Boolean))

    client\_node.set\_value(client\_node\_dv)

    print("Value of : " + str(client\_node) + ' : ' + str(client\_node\_value))

def client\_connection(url):

    try:

        client = Client(url)

        client.connect()

        print("Connected to OPC UA Server with url :", url)

    except:

        client = None

        print("Unable to connect to OPC UA Server with url :", url)

    return None

## ProductDetector.py

import torch

import numpy as np

import cv2

import sys

from opcua import Client, ua

from time import time

import Config

import gc

class ProductDetector:

    def \_\_init\_\_(self, model\_name, device, conf\_th = 0.8):

        self.model = self.load\_model(model\_name)

        self.device = device

        self.model.to(self.device)

        self.model.eval()

        self.classes = self.model.names

        self.conf\_th = conf\_th

        print("Using Device: ", self.device)

    @staticmethod

    def load\_model(model\_name=None): # Pytorch hub'dan Yolov5 modelini indiriyoruz ve bunu modüle geri döndürüyoruz.

        if model\_name:

            model = torch.hub.load('ultralytics/yolov5', 'custom', path=model\_name)

        else:

            model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)

        return model

    def predict(self, frame): # Kameradan aldığı görüntüyü modele sokarak ondan tahmin oranı alıyoruz.

        frame = [frame]

        with torch.no\_grad():

            results = self.model(frame)

        labels, cord = results.xyxyn[0][:, -1], results.xyxyn[0][:, :-1]

        return labels, cord

    def class\_to\_label(self, x): # Classlarımızı labela dönüştürüyoruz.

        return self.classes[int(x)]

    def plot\_boxes(self, results, frame): # Aranan objenin hangi konumlar içinde olduğunu buluyoruz.

        labels, cord = results

        n = len(labels)

        x\_shape, y\_shape = frame.shape[1], frame.shape[0]

        obj\_class = ""

        for i in range(n):

            row = cord[i]

            if row[4] >= self.conf\_th: # Hangi doğruluk değerinin üstünü ekrana yazdırmak istiyorsak yazıyoruz.

                obj\_class = self.class\_to\_label(labels[i])

                if obj\_class == "Mavi":

                    obj\_class = "Red"

                elif obj\_class == "Red":

                    obj\_class = "Blue"

                x1, y1, x2, y2 = int(row[0]\*x\_shape), int(row[1]\*y\_shape), int(row[2]\*x\_shape), int(row[3]\*y\_shape)

                bgr = (0, 255, 0)

                cv2.rectangle(frame, (x1, y1), (x2, y2), bgr, 2)

                cv2.putText(frame, obj\_class, (x1, y1), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, bgr, 2)

        return obj\_class, frame

    def clear\_gpu\_cache(self):

        print('Clearing gpu cache ...')

        self.model = None

        gc.collect()

        with torch.no\_grad():

            torch.cuda.empty\_cache()

## requirements.txt

# YOLOv5 requirements

# Usage: pip install -r requirements.txt

# Base ------------------------------------------------------------------------

gitpython>=3.1.30

opcua

opcua.client

matplotlib>=3.3

numpy>=1.18.5

opencv-python>=4.1.1

Pillow>=7.1.2

psutil  # system resources

PyYAML>=5.3.1

requests>=2.23.0

scipy>=1.4.1

thop>=0.1.1  # FLOPs computation

torch>=1.7.0  # see https://pytorch.org/get-started/locally (recommended)

torchvision>=0.8.1

tqdm>=4.64.0

# protobuf<=3.20.1  # https://github.com/ultralytics/yolov5/issues/8012

# Logging ---------------------------------------------------------------------

# tensorboard>=2.4.1

# clearml>=1.2.0

# comet

# Plotting --------------------------------------------------------------------

pandas>=1.1.4

seaborn>=0.11.0

# Export ----------------------------------------------------------------------

# coremltools>=6.0  # CoreML export

# onnx>=1.10.0  # ONNX export

# onnx-simplifier>=0.4.1  # ONNX simplifier

# nvidia-pyindex  # TensorRT export

# nvidia-tensorrt  # TensorRT export

# scikit-learn<=1.1.2  # CoreML quantization

# tensorflow>=2.4.0  # TF exports (-cpu, -aarch64, -macos)

# tensorflowjs>=3.9.0  # TF.js export

# openvino-dev  # OpenVINO export

# Deploy ----------------------------------------------------------------------

setuptools>=65.5.1 # Snyk vulnerability fix

# tritonclient[all]~=2.24.0

# Extras ----------------------------------------------------------------------

# ipython  # interactive notebook

# mss  # screenshots

# albumentations>=1.0.3

# pycocotools>=2.0.6  # COCO mAP

# ultralytics  # HUB https://hub.ultralytics.com

# ÖZGEÇMİŞ

**Kişisel Bilgiler**

Soyadı, adı : Ahmet Erdem KENET

Uyruğu : Türkiye Cumhuriyeti

Doğum tarihi ve yeri : 20.07.1999 Kadıköy İSTANBUL

Telefon : 05350714794

e-mail : ahmet.kenet999@gmail.com

**Eğitim**

|  |  |  |
| --- | --- | --- |
| **Derece** | **Eğitim Birimi** | **Mezuniyet Tarihi** |

|  |  |  |
| --- | --- | --- |
| Lise | Mustafa Kemal Anadolu Lisesi | 2017 |

**Yabancı Dil**

İngilizce B2

Almanca A1

**Kişisel Bilgiler**

Soyadı, adı : Ayşeana KOCATÜRK

Uyruğu : Türkiye Cumhuriyeti

Doğum tarihi ve yeri : 10.12.1998 / Karapınar KONYA

Telefon : 05466605344

e-mail : aysekocaturk2@gmail.com

**Eğitim**

|  |  |  |
| --- | --- | --- |
| **Derece** | **Eğitim Birimi** | **Mezuniyet Tarihi** |

|  |  |  |
| --- | --- | --- |
| Lise | Karapınar Anadolu Öğretmen Lisesi | 2017 |

**Yabancı Dil**

İngilizce B2

**Kişisel Bilgiler**

Soyadı, adı : Halil Cem AYDIN

Uyruğu : Türkiye Cumhuriyeti

Doğum tarihi ve yeri : 01.01.19992 / KARABÜK

Telefon : 05453417518

e-mail : halilcemaydin@gmail.com

**Eğitim**

|  |  |  |
| --- | --- | --- |
| **Derece** | **Eğitim Birimi** | **Mezuniyet Tarihi** |

|  |  |  |
| --- | --- | --- |
| Lise | Zübeyde Hanım Anadolu Lisesi | 2017 |

**Yabancı Dil**

İngilizce B2

**Kişisel Bilgiler**

Soyadı, adı : Muhammet Çağrı ÖZBEK

Uyruğu : Türkiye Cumhuriyeti

Doğum tarihi ve yeri : 27.02.2000 / ESKİŞEHİR

Telefon : 05377267315

e-mail : mcagriozbek@gmail.com

**Eğitim**

|  |  |  |
| --- | --- | --- |
| **Derece** | **Eğitim Birimi** | **Mezuniyet Tarihi** |

|  |  |  |
| --- | --- | --- |
| Lise | Eskişehir Anadolu Lisesi | 2018 |

**Yabancı Dil**

İngilizce B2