

TEZİN ADI
GÖRÜNTÜ İŞLEME İLE MODÜLER CİSİM
AYRIŞTIRMA OTOMASYON SİSTEMİ

2018
LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

Ad SOYAD

İlkay ALTAY	2013010219031
Yunus Sürücü	2013010219028
Hakan KANIK	2013010219060
Fatih Erkam ÇORUMLUOĞLU	2013010219070

GÖRÜNTÜ İŞLEME İLE MODÜLER CİSİM AYRIŞTIRMA OTOMASYON SİSTEMİ

Ad SOYAD:

İlkay ALTAY	2013010219031
Yunus SÜRÜCÜ	2013010219028
Hakan KANIK	2013010219060
Fatih Erkam ÇORUMLUOĞLU	2013010219070

**Karabük Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümünde
Lisans Tezi
Olarak Hazırlanmıştır**

KARABÜK

Mayıs 2018

İlkay ALTAY, Yunus SÜRÜCÜ, Hakan KANIK ve Fatih Erkam ÇORUMLUOĞLU tarafından hazırlanan “GÖRÜNTÜ İŞLEME İLE MODÜLER CİSİM AYRIŞTIRMA OTOMASYON SİSTEMİ” başlıklı bu tezin Lisans Tezi olarak uygun olduğunu onaylarım.

Unvan Dr.Öğr.Ü Hüseyin ALTINKAYA

.....

Tez Danışmanı,

Karabük Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü

Ünvan. Dr. Ad SOYAD

.....

xxxx Üniversitesi xxx Mühendisliği Bölümü

Ünvan. Dr. Ad SOYAD

.....

Karabük Üniversitesi xxxx Bölümü

“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.”

İlkay ALTAY, Yunus SÜRÜCÜ, Hakan KANIK, Fatih Erkam ÇORUMLUOĞLU

ÖZET

Bugünlerde fabrikaların birçok insan gücüne ihtiyacı vardır. Bu ihtiyaçla birlikte, fabrikalarda insan gücünü ve maliyetlerini azaltmak için otomasyon sistemlerine ihtiyaç duyulmaktadır. Bazı fabrikalarda, çeşitli malzemeler aynı konveyör bant üzerinde gelir. Bu işlem, insan gücünün ve zamanının kaybolmasına, görüntü işleme ile yapılmamasına neden olur, çünkü aksi takdirde insan müdahalesi ile yapılır. Bu sorunu aşmak için görüntü işleme teknikleri kullanılır. Temel olarak görüntü işleme, bilgisayar yazılımı kullanarak kameraların elde ettiği görüntünün işlenmesini ve anlamlandırılmasını sağlamaktır. Görüntü işleme uygulamaları hemen hemen her sektörde kullanılmakta ve bu uygulamaların kullanım alanı gün geçtikçe artmaktadır.

Görüntü işleme ve otomasyon sistemleri tıp, elektronik, tekstil ve savunma gibi birçok sektörde birlikte kullanılmaktadır.

Bu çalışmada amacımız; görüntü işleme yazılımı ile aynı üretim bandından gelen çeşitli ürünlerin standart bir kamera kullanarak üretilmesidir. Biz renk ve şekiller için farklı bölümlere tanımlanan malzemeler ayırmak ve istediğiniz şekilde gelen kutulara dağıtmaktır.

Görüntü işlemede, C# programı ve Aforge kütüphanesinden faydalanarak cisimlerin şekil ve renklerinin algılanması sağlanır. Görüntüdeki cisimlerin şekilleri algılandıktan sonra STEP7 kütüphanesi aracılığıyla PLC'ye uygun girişler verilir. PLC, TIA portal yazılımında yazılan kodlara göre malzemeleri ayıracak motorları çalıştırmak için kullanılır.

Lisans Tezi

GÖRÜNTÜ İŞLEME İLE MODÜLER CİSİM AYRIŞTIRMA OTOMASYON SİSTEMİ

Ad SOYAD

İlkay ALTAY	2013010219031
Yunus SÜRÜCÜ	2013010219028
Hakan KANIK	2013010219060
Fatih Erkam ÇORUMLUOĞLU	2013010219070

Karabük Üniversitesi

Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

Tez Danışmanı:

Ünvan. Dr.Öğr.Ü Hüseyin ALTINKAYA

Mayıs 2018, XX sayfa

Bu çalışmada amacımız, görüntü işleme programı ile aynı üretim bandından gelen çeşitli ürünlerin standart bir kamera kullanarak şekillerine ve rengine göre ayrıştırılmasını sağlamaktır. Çalışmada, cisimlerin şekillerine göre farklı bölmelere ayrıştırılması sağlanmaktadır.

Anahtar Kelimeler : PLC, Konveyör Bant, Görüntü İşleme, Paketleme

ABSTRACT

Nowadays, factories need a lot of man power. Together with this need, to decrease man power and costs in factories the need for automation systems arrived. In some factories, various materials come on the same conveyer band. This process causes loss of human power and time when not done with image processing because otherwise it would be done by human intervention. To overcome this issue image processing techniques are used. Image processing in its basic is to process and make meaning of image obtained by cameras using computer software. Applications of image processing are used in almost every sector and the usage area of this applications increases day by day.

Image processing and automation systems are used together in many sectors like; Medicine, electronic, textile and defence.

In this study, our objective is to define various products coming from the same production band in manufacturing with image processing software by using a standard camera. We want to separate the defined materials into different sections for their color and shapes and distribute them to coming boxes in the desired way.

In image processing, C # program and Aforge library are used for detecting shapes and colors of objects. Once the shapes of the objects in the image have been detected, inputs are sent to the PLC via the STEP7 library. PLC is used for running motors that will separate the materials according to the codes written in the TIA portal software.

B. Sc. Thesis

MODULAR BODY DECOMPOSITION AUTOMATION SYSTEM WITH IMAGE PROCESSING

Name SURNAME

İlkay ALTAY	2013010219031
Yunus SÜRÜCÜ	2013010219028
Hakan KANIK	2013010219060
Fatih Erkam ÇORUMLUOĞLU	2013010219070

Karabük University
Faculty of Engineering
Department of Electrical-Electronics Engineering

Thesis Advisor:
Assoc.Dr. Name SURNAME
May 2018, XX pages

In this study, our objective is to define various products coming from the same production band in manufacturing with image processing software by using a standard camera. We want to separate the defined materials into different sections for their shapes and distribute them to coming boxes in the desired way.

Key Words : PLC, conveyor belt, Image-Processing, Packaging

TEŞEKKÜR

Bu çalışmanın gerçekleştirmesinde, değerli bilgilerini bizlerle paylaşan, kıymetli zamanını ayırıp sabırla büyük bir ilgiyle bizlere yardımcı olan, gelecekteki mesleki hayatımızda da bizlere verdiği değerli bilgilerden faydalanacağımızı düşündüğümüz kıymetli hocamız sayın Dr.Öğr.Üyesi Hüseyin ALTINKAYA hocamıza teşekkürü bir borç biliyoruz.

İÇİNDEKİLER

	Sayfa
ÖZET	4
ABSTRACT	6
TEŞEKKÜR	8
İÇİNDEKİLER.....	9
ÇİZELGELERİN LİSTESİ	11
ŞEKİLLERİN LİSTESİ.....	11
BÖLÜM 1 GİRİŞ	12
BÖLÜM 2 NE YAPACAĞIZ?.....	13
2.1. Projede Kullanılacak Olan Malzemeler	14
2.2. Problem	14
2.3. Sistemin Autocad Prototipi	15
2.4. Olay Örgüsü	16
2.5. Devre Şeması	17
BÖLÜM 3 C#.....	19
3.1 Uygulamaya Ait Örnek Görüntü	19
3.2 C# Form Tasarımı	19
3.3 C# ile Yazılmış Kodlar	20
BÖLÜM 4 PLC	36
4. SİSTEMİN LADDER DİAGRAMI.....	36
4.1.Tag table	36
4.1.1Ana Blok.....	36

4.1.2 Motor Yön Kontrol Bloğu.....	36
4.1.3Çalışma Takvimi.....	38
BÖLÜM 5.....	39
5.KULLANILAN MALZEMELER.....	39
5.1 İskelet Kısım.....	39
5.2. PLC – Siemens s7-1200.....	39
5.3 24V DC Motor.....	40
5.4 Güç Kaynağı.....	40
5.5 Röle.....	41
5.6 RGB Kamera.....	41
5.7 Butonlar.....	41
5.8 Bant.....	42
5.9 Kapılar.....	42
5.10 Kutu.....	42
BÖLÜM 6.....	43
7. SONUÇ VE ÖNERİLER.....	43
KAYNAKLAR.....	44
ÖZGEÇMİŞ.....	45

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 1 Çalışma Takvimi Tablosu	38

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1. Projenin Prototip çizimi	15
Şekil 2. PLC Devre şeması	17
Şekil 3. DC Motor Röle Bağlantısı	18
Şekil 4. Tag Table	36
Şekil 5. Projenin İskeleti.....	39
Şekil 6. PLC	39
Şekil 7. 24V DC Motor	40
Şekil 8. 24V DC Güç Kaynağı	40
Şekil 9. Röle	41
Şekil 10. Buton	41
Şekil 10. Kapılar ve Bant	42

BÖLÜM 1 GİRİŞ

Bu çalışmada görüntü işleme ile modüler cisim ayrıştırma yapılmaktadır. Burada yapacağımız işlem ise imalatı tek üretim bandı üzerinden gelen çeşitli ürünlerin standart bir kamera aracılığıyla görüntü işleme programlarında tanımlanmasıdır. Tanımlanan cisimler plc aracılığıyla renk ve şekillerine göre farklı bölmelere ayrıştırılmasıdır.

Görüntü işleme: Ölçülmüş veya kaydedilmiş olan elektronik ortamda amaca uygun şekilde değiştirmeye yönelik yapılan bilgisayar çalışmasıdır.

PLC (Programmable Logic Controller): Algılayıcılardan aldığı bilgiyi kendine verilen programa göre işleyen ve iş elemanlarına aktaran bir mikroişlemci tabanlı bir cihazdır.

TIA portal V14: Temel programlama tekniklerinin yanında gelişmiş komutları da içinde barındıran bir arayüzdür.

Profinet: Endüstriyel Ethernet protokolü Profinet, multiprotokol Input/Output cihazları ve Turck'un sistemleri gibi Giriş/Çıkış cihazları ile Profinet master özellikli kontrollörler arasındaki tüm alış-veriş işini tamamlar

C#: Yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiş olan bir yazılım dilidir.

Burada cismin şeklini ve rengini tanımlayacağımız için görüntü işleme yöntemi kullanılmaktadır. Bu görüntüleri C# programı ile kodlaması yapılır. Girilen bu kodlar Profinet yöntemiyle PLC programına aktarılır. Otomasyonda kullanılacak olan paketleme sistemi için PLC (Programmable Logic Controller) kullanılır. PLC'de yapılacak işlemler TIA Portal V14 programında oluşturulup PLC'ye aktarılır.

BÖLÜM 2

2. NE YAPACAĞIZ ?

Bu çalışmada, aynı konveyör bant üzerinden gelen cisimlerin şekillerine göre kare ve yuvarlak ve renklerine göre kırmızı yeşil ve mavi olarak ayrıştırılması ve ayrıştırılan cisimlerin daha önceden şekil ve renklere göre belirlenen bölmelere aktararak paketlenmesi amaçlanmaktadır.

Tia Portal aracılığıyla yazılan koda göre çalışan PLC, konveyör bandı harekete geçirecek ve konveyör bandın üzerindeki cisim kameranın altında geldiğinde konveyör bant belirli bir süre bekletilecek ve kameranın cismin görüntüsünü sağlıklı bir şekilde alabilmesi sağlanacaktır. Kamera aracılığıyla elde edilen görüntünün rengi ve şekli C# programında yazılan kod aracılığıyla tanımlanacaktır. Cisim tanımlandıktan sonra tanımlı renk ve şekle uygun alt kodun çalışması sağlanacaktır. Her renk ve şekil kombinasyonu için ayrı bir alt kod bulunacak ve bu alt kodlar Tia Portal programında o şekil ve renge karşılık gelen uygun kodun çalıştırılmasını sağlayacaktır.

C#'tan Tia Portal'a veri aktarımı ProfiNet aracılığıyla yapılmaktadır. Tia Portal'da da her renk ve şekil kombinasyonuna karşılık gelen ve cisimlerin uygun bölmelere ayrıştırılmasını sağlamak için belirli motorların belirli zamanda çalıştırılmasını sağlayacak kod çalıştırılır. Bu kodlar C#dan gelen veriye göre çalışmaktadır. Örneğin; C#'da cisim yeşil yuvarlak olarak tanımlandığında ve sonuç olarak, yeşil yuvarlak kodunun çalıştırılması komutu gönderildiğinde Tia Portalda yeşil yuvarlağa karşılık gelen kod çalışır. Tia Portal, PLCnin belirli çıkışlarından belirli zamanlarda çıkış vermesini sağlayarak o çıkışa bağlı olan motorun istenilen zaman aralığında çalışmasını sağlar. İstenilen motorların istenilen zamanda çalışması nesnelerin uygun bölmelere ayrıştırılması için gereklidir.

Konveyör bant üzerindeki tahta kollar motorlara bağlıdır ve bağlı oldukları motor çalıştığında açılıp kapanırlar. PLC den gelen sinyale göre çalışan motor, tahta kolun kapanmasını ve konveyör bant üzerinden gelen cismin tahta kola takılması cismin istenilen bölmeye itilmesini sağlar.

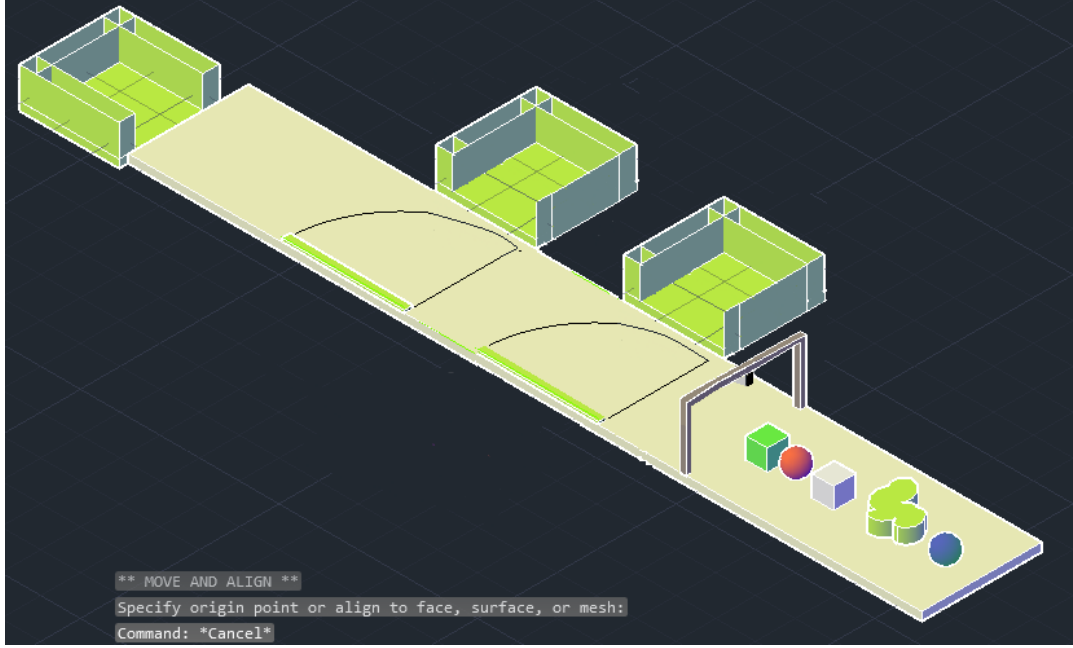
2.1 Proje De Kullanılacak Olan Malzemeler

- PLC-s7 1200
- Motorlar
- Konveyör bant
- RGB kamera
- Bant üzerindeki, cisimlerin ayrıştırılmasını sağlayan kapılar,
- Sistemde kullanılacak farklı renk ve şekilde cisimle

2.2 Problem

Bir fabrikada konveyör bant üzerinden gelen cismi kamera gördüğünde farklı 2 şekil ve bozuk şeklin tek renk seçilerek ayrıştırılma işlemi yapılacaktır. Start butonuna basıldığında m1 motoru çalışmaya başlayacaktır. Herhangi bir anda stop butonuna basıldığında sistem hemen duracaktır. Şekiller kamera yardımı ile ayrıştırılacak ve şekil kare ise m2 çalışarak 1nci Kapı kapanacak, şekil yuvarlak ise m2 çalışarak 2nci kapı kapanacaktır, eğer şekil bozuk ise m2 ve m3 motorları çalışmayarak kapılar açık konumda kalacaktır. Eğer cisim kare ise m2, cisim yuvarlak ise m3 motoru çalışacaktır. Cisim kare ve mavi ise ; m2 motoru 2.3 saniye, cisim yuvarlak ve mavi ise m2 motoru 4 saniye kapı kapanacaktır, cisim şekil olarak kare ve yuvarlak renk olarakta mavi dışında ise direk olarak kapılar açık konumda kalara cisim bozuk bölmeye düşecektir. Ayrıştırılma işleminden sonra m2 ve m3 kollarını çalıştıran motorlar başlangıç konumlarına geri alınacaktır.

2.3 SİSTEMİN AUTOCAD PROTOTİPİ



Şekil 1 Autocad proje prototipi

2.4 OLAY ÖRGÜSÜ

Olay1: Start düğmesine basılmasının ardından m1 motoru(konveyör bant) çalışacak ve konveyör bant harekete geçecektir.

Olay2: Konveyör bandın üzerine konulan cisimlerin şekli kamera aracılığıyla C# programında görüntü işleme yöntemleri ile belirlenecektir.

Olay3: Cismin C# tarafından kare ve mavi olarak algılanması durumunda m2 motoru(1. Kapı) çalışacak ve kapı kapanacaktır. Bu şekilde cismin uygun bölmeye düşmesi sağlanacaktır.

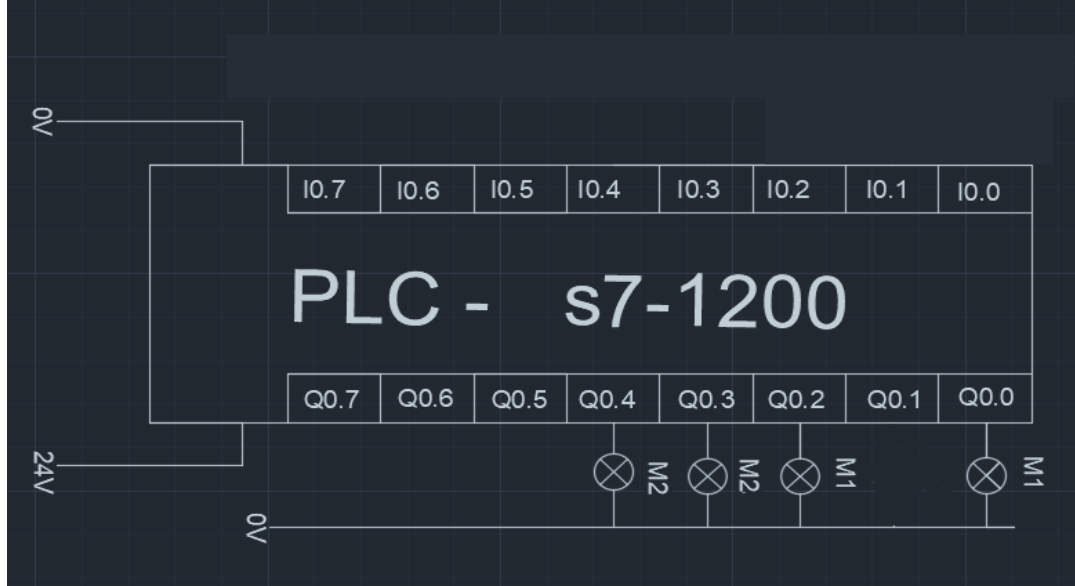
Olay4: Kapalı konumda bulunan 1. Kapı m2 motorunun ters yönde çalıştırılması ile başlangıç konumuna döndürülecektir.

Olay5: Cismin C# tarafından yuvarlak ve mavi olarak algılanması durumunda m3 motoru(2. Kapı) çalışacak ve kapı kapanacaktır. Bu şekilde cismin uygun bölmeye düşmesi sağlanacaktır.

Olay6: Kapalı konumda bulunan 2. Kapı m3 motorunun ters yönde çalıştırılması ile başlangıç konumuna döndürülecektir.

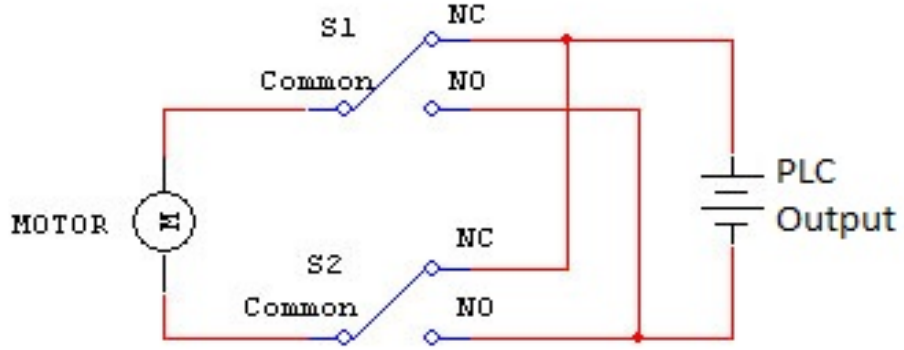
Olay7: Cismin C# tarafından şekil olarak kare veya yuvarlak, renk olarak mavi dışında olduğu belirlenmesi durumunda kapılar açık kalacak ve m1 motorunun çalışmasıyla konveyöre bant ilerlemeye devam edecek ve bozuk şeklin bandın sonundaki kutuya düşmesi sağlanacaktır.

2.5 DEVRE ŞEMASI



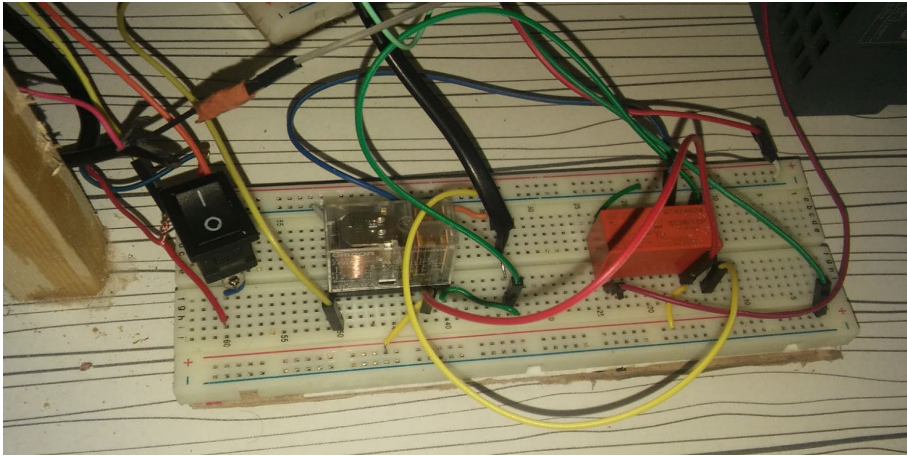
Şekil 2 PLC bağlantısı

Motorların PLC'ye bağlantı şemasıdır. PLC'den gelen çıkışa göre kapıları açıp kapanmasını sağlayan M1 motoru ve M2 motorları ileri veya geri önde hareket ettirilecektir. Bu, aynı motorun aynı anda 2 çıkışa birden bağlanması sonucu hata vermektedir. Bu yüzden aynı anda M1 ve M2 motoruna çıkış verilmemesi sağlanmaktadır. Motorun yön değiştirmesini PLC üzerinden değilde röle devresi kullanarak yapmaktayız.



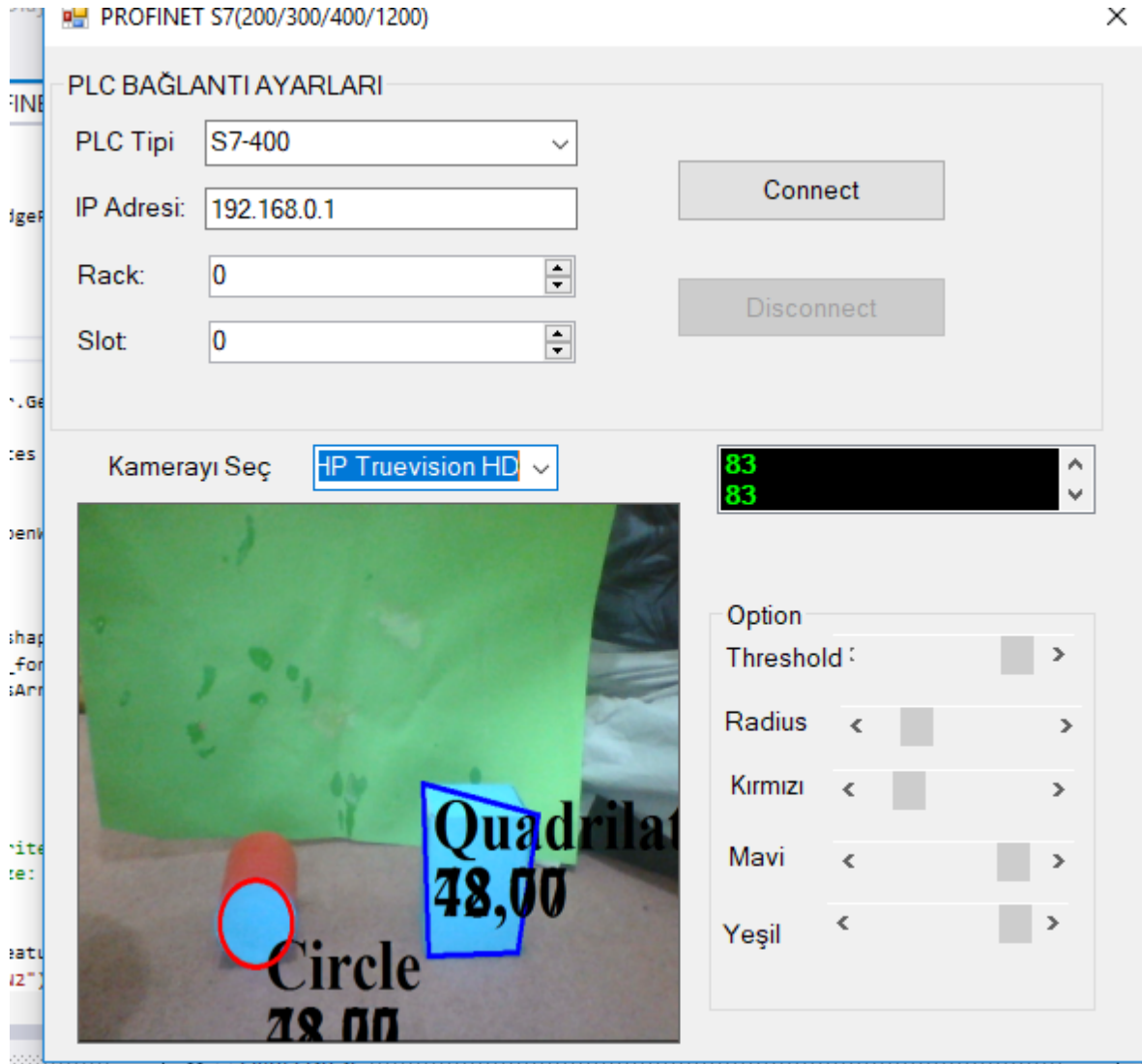
Şekil 3 Röle bağlantısı

Motorların ileri veya geri yönde hareketi röle devresi ile sağlanır. Röle devresinde normalde açık uçlar kapanır, normalde kapalı uçlar açık konuma getirilerek motorun ters yönde çalışmasını sağlar.



BÖLÜM 3

KAMERADAN GELEN GÖRÜNTÜYE GÖRE CİSİMLERİN ŞEKLİNİ
ALGILAYAN C# KODU VE ÇALIŞIR HALİYLE İLGİLİ ÖRNEK GÖRÜNTÜ
C# FORM TASARIMI



C# KOD KISMI

Kullanılan kütüphaneler :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using PROFINET_STEP_7.Profinet;
using AForge;
using AForge.Imaging;
using AForge.Video;
using AForge.Video.DirectShow;
using AForge.Imaging.Filters;
using AForge.Math.Geometry;

namespace PROFINET_STEP_7
{
    public partial class FormMain : Form
    {
        { Kamera ayarlarının olduğu kısım
            FilterInfoCollection _device;
            VideoCaptureDevice _captureDevice;
            Bitmap _bitmapEdgeImage, _bitmapBinaryImage, _bitmapGreyImage,
            _bitmapBlurImage, _colorFilterImage;
            //ColorFiltering _colorFilter = new ColorFiltering();
            EuclideanColorFiltering _colorFilter = new EuclideanColorFiltering();
            System.Drawing.Font _font = new System.Drawing.Font("Times New Roman", 48,
            FontStyle.Bold);
```

```

        System.Drawing.SolidBrush _brush = new
System.Drawing.SolidBrush(System.Drawing.Color.Black);
        SobelEdgeDetector _edgeFilter = new SobelEdgeDetector();
        bool _blurFlag = false;
        int ipenWidth = 5, iFeatureWidth;
        int iThreshold = 40, iRadius = 40;
        int iColorMode = 1, iRedValue = 220, iGreenValue = 30, iBlueValue = 30;

```

PLC'nin tanımlandığı kısım;

```

private PLC plc = null;
private ExceptionCode errCode;

public FormMain()
{
    InitializeComponent();
    CheckForIllegalCrossThreadCalls = false;
}

```

Tia portal'a bağlanan PLC'nin IP adresinin tanımlandığı kısım

```

private void FormMain_Load(object sender, EventArgs e)
{
    try
    {
        this.SetEnabledBotton(true);
        cboxPLCs.SelectedIndex = 2;
        txtIPAddress.Text = "192.168.0.1";
    }
    catch (Exception ex)
    {
        MessageBox.Show(this, ex.Message, "Bağlanamadı", MessageBoxButtons.OK,
MessageBoxIcon.Error);

```

```
}
```

Kullanılacak kameranın seçildiği kısım:

```
try
```

```
{
```

```
    _device = new FilterInfoCollection(FilterCategory.VideoInputDevice);
```

```
    for (var i = 0; i < _device.Count; i++)
```

```
        comboBox1.Items.Add(_device[i].Name);
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    MessageBox.Show(ex.Message);
```

```
}
```

```
}
```

```
private void FormMain_FormClosing(object sender, FormClosingEventArgs e)
```

```
{
```

```
    try
```

```
    {
```

```
        StopCameras();
```

```
    }
```

```
    catch
```

```
    {
```

```
        return;
```

```
    }
```

```
}
```

Kameranın başlatılma kısmı:

```
#region MyMethods
private void StartCameras(int deviceindex)
{
    try
    {
        _CaptureDevice = new
VideoCaptureDevice(_device[deviceindex].MonikerString);
        _CaptureDevice.NewFrame += new NewFrameEventHandler(get_Frame);
        //Kamerayı başlat
        _CaptureDevice.Start();
    }
}
```

Belirlenen renk değerlerinin ve bağlantı durumlarının ekranda gösterildiği kısım:

```
listBox1.Items.Add("Webcam connect");
ScrollDown();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
private void StopCameras()
{
    try
    {
        _CaptureDevice.Stop();
    }
    catch (Exception)
    {
        return;
    }
}
```

```

    }
    void ScrollDown()
    {
        try
        {
            listBox1.SelectedIndex = listBox1.Items.Count - 1;
            listBox1.SelectedIndex = -1;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

Kamerada algılanan cismin şekline göre etrafının çerçeveye alınması :

```

private void get_Frame(object sender, NewFrameEventArgs eventArgs)
{
    Bitmap _BsourceFrame = (Bitmap)eventArgs.Frame.Clone();
    pictureBox1.Image = BlobDetection(_BsourceFrame);
}

private System.Drawing.Point[] ToPointsArray(List<IntPoint> points)
{
    System.Drawing.Point[] array = new System.Drawing.Point[points.Count];

    for (int i = 0, n = points.Count; i < n; i++)
    {
        array[i] = new System.Drawing.Point(points[i].X, points[i].Y);
    }
    return array;
}

```


Cismin çapının belirlenme kısmı :

```
private void sbRadius_Scroll(object sender, ScrollEventArgs e)
{
    iRadius = 76;
    listBox1.Items.Add(iRadius.ToString());
    ScrollDown();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        StopCameras();
        StartCameras(comboBox1.SelectedIndex);
        //Kameradan anlık görüntü yakalaması
        new NewFrameEventHandler(CaptureDevi_NewFrame);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Toleransın belirlenme kısmı :

```
private void sbThreshold_Scroll(object sender, ScrollEventArgs e)
{
    iThreshold = 233;
    listBox1.Items.Add(iThreshold.ToString());
    ScrollDown();
}
```

Cismin uzaklığının belirlenme kısmı :

```
private double FindDistance(int _pixel)
{
    double _distance;
    double _ObjectWidth = 10, _focalLength = 604.8;

    _distance = (_ObjectWidth * _focalLength) / _pixel;

    return _distance;
}
```

Connect butonuna basıldığında C# ile PLC arasındaki bağlantının gerçekleştirildiği kısım :

```
private void btnConnection_Click(object sender, EventArgs e)
{
    try
    {
        if (string.IsNullOrEmpty(txtIPAddress.Text)) throw new Exception("IP adresi uygun değil");
        int selectionIndex = cboPLCs.SelectedIndex;
        CPU_Type cpuType = CPU_Type.S7200;
        string ipAddress = txtIPAddress.Text;
        switch (selectionIndex)
```

Kullanılacak PLC CPU'sunun seçildiği kısım:

```
{
    case 0:
        cpuType = CPU_Type.S7200;
        break;
    case 1:
        cpuType = CPU_Type.S7300;
```

```

        break;
    case 2:
        cpuType = CPU_Type.S7400;
        break;
    case 3:
        cpuType = CPU_Type.S71200;
        break;
    default:
        cboxPLCs.SelectedIndex = 3;
        cpuType = CPU_Type.S71200;
        break;
}

```

PLC ayarlarının atandığı kısım :

```

        plc = new PLC(cpuType, ipAddress, (short)numericUpDownRack.Value,
(short)numericUpDownSlot.Value);
        if (!plc.IsAvailable) throw new Exception("PLC uygun değil!");
        errCode = plc.Open();
        if (errCode != ExceptionCode.ExceptionNo) throw new
Exception(plc.lastErrorString);
        this.SetEnabledBotton(false);
    }
    catch (Exception ex)
    {
        MessageBox.Show(this, ex.Message, "Hata!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

```

```

private void btnDisconnection_Click(object sender, EventArgs e)

```

```

{
    try
    {
        this.SetEnabledBotton(true);
    }
    catch (Exception ex)
    {
        MessageBox.Show(this, ex.Message, "Hata!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void SetEnabledBotton(bool flag)
{
    btnConnection.Enabled = flag;
    txtIPAddress.Enabled = flag;
    cboxPLCs.Enabled = flag;
    numericUpDownRack.Enabled = flag;
    numericUpDownSlot.Enabled = flag;
    btnDisconnection.Enabled = !btnConnection.Enabled;
}

private void comboBox1_SelectedIndexChanged_1(object sender, EventArgs e)
{
    {
        try
        {
            StopCameras();
            SrartCameras(comboBox1.SelectedIndex);
            //Anlık görüntü yakalayan kısım
            new NewFrameEventHandler(CaptureDevi_NewFrame);

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}

```

Algılanacak cismin tolerans değerinin belirlendiği kısım :

```

private void sbThreshold_Scroll_1(object sender, ScrollEventArgs e)
{
    iThreshold = sbThreshold.Value;
    listBox1.Items.Add(iThreshold.ToString());
    ScrollDown();
}

```

Algılanacak cismin yarıçap değerinin belirlendiği kısım :

```

private void sbRadius_Scroll_1(object sender, ScrollEventArgs e)
{
    iRadius = sbRadius.Value;
    listBox1.Items.Add(iRadius.ToString());
    ScrollDown();
}

```

Algılanacak cismin RGB kırmızı değerinin belirlenme kısmı :

```

private void sbRedColor_Scroll(object sender, ScrollEventArgs e)
{
    iRedValue = sbRedColor.Value;

    listBox1.Items.Add("Red: " + iRedValue.ToString());
    ScrollDown();
}

```

Algılanacak cismin RGB mavi değerinin belirlenme kısmı :

```
private void sbBlueColor_Scroll(object sender, ScrollEventArgs e)
{
    iBlueValue = sbBlueColor.Value;

    listBox1.Items.Add("Blue: " + iBlueValue.ToString());
    ScrollDown();
}
```

Algılanacak cismin RGB yeşil değerinin belirlenme kısmı :

```
private void sbGreenColor_Scroll(object sender, ScrollEventArgs e)
{
    iGreenValue = sbGreenColor.Value;

    listBox1.Items.Add("Green: " + iGreenValue.ToString());
    ScrollDown();
}
```

```
private void textBox1_TextChanged(object sender, EventArgs e)
{
}

private void listBox1_SelectedIndexChanged_1(object sender, EventArgs e)
{
}

private int db;
private int startByteAddr;
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}
```

Kaydırma çubuklarından gelen değerin cismin RGB renk değerlerine atandığı kısım :

```
private Bitmap BlobDetection(Bitmap _bitmapSourceImage)
{
    switch (iColorMode)
    {
        case 1:
            iRedValue = sbRedColor.Value;
            iBlueValue = sbBlueColor.Value;
            iGreenValue = sbGreenColor.Value;

            _colorFilter.CenterColor = new RGB((byte)iRedValue, (byte)iGreenValue,
(byte)iBlueValue);
            _colorFilter.Radius = (short)iRadius;
            _colorFilterImage = _colorFilter.Apply(_bitmapSourceImage);

            break;
        case 2:
            iRedValue = sbRedColor.Value;
            iBlueValue = sbBlueColor.Value;
            iGreenValue = sbGreenColor.Value;

            _colorFilter.CenterColor = new RGB((byte)iRedValue, (byte)iGreenValue,
(byte)iBlueValue);
            _colorFilter.Radius = (short)iRadius;
            _colorFilterImage = _colorFilter.Apply(_bitmapSourceImage);
            break;
        case 3:
            iRedValue = sbRedColor.Value;
```

```

        iBlueValue = sbBlueColor.Value;
        iGreenValue = sbGreenColor.Value;

        _colorFilter.CenterColor = new RGB((byte)iRedValue, (byte)iGreenValue,
(byte)iBlueValue);
        _colorFilter.Radius = (short)iRadius;
        _colorFilterImage = _colorFilter.Apply(_bitmapSourceImage);
        break;
    }
    Grayscale _grayscale = new Grayscale(0.2125, 0.7154, 0.0721);
    _bitmapGreyImage = _grayscale.Apply(_colorFilterImage);

    //köşe belirleme olayının gerçekleştiği kısım
    if (_blurFlag == true)
    {
        GaussianBlur _blurfilter = new GaussianBlur();
        _bitmapBlurImage = _blurfilter.Apply(_bitmapGreyImage);
        _bitmapEdgeImage = _edgeFilter.Apply(_bitmapBlurImage);
    }
    else if (_blurFlag == false)
    {
        _bitmapEdgeImage = _edgeFilter.Apply(_bitmapGreyImage);
    }

    Threshold _threshold = new Threshold(iThreshold);
    _bitmapBinaryImage = _threshold.Apply(_bitmapEdgeImage);

    //Blob sayıcı algoritmasının bir örneğini oluşturur
    BlobCounter _blobCounter = new BlobCounter();
    //filtre belirlenir

```



```

_blobCounter.MinWidth = 70;
_blobCounter.MinHeight = 70;
_blobCounter.FilterBlobs = true;

_blobCounter.ProcessImage(_bitmapBinaryImage);
Blob[] _blobPoints = _blobCounter.GetObjectsInformation();

```

```

Graphics _g = Graphics.FromImage(_bitmapSourceImage);

```

Şekil belirleyici kısmı :

```

SimpleShapeChecker _shapeChecker = new SimpleShapeChecker();
for (int i = 0; i < _blobPoints.Length; i++)
{
    List<IntPoint> _edgePoint = _blobCounter.GetBlobsEdgePoints(_blobPoints[i]);
    List<IntPoint> _corners = null;
    AForge.Point _center;
    float _radius;

```

Kare şeklinin belirlendiği ve PLC'ye uygun çıkışın verildiği kısım:

```

if (_shapeChecker.IsQuadrilateral(_edgePoint, out _corners))
{
    plc.Write("M100.3", 0);
    plc.Write("M100.1", 1);
    plc.Write("M100.1", 0);
    plc.Write("M0.2", 0);

    Rectangle[] _rects = _blobCounter.GetObjectsRectangles();

    System.Drawing.Point[] _coordinates = ToPointsArray(_corners);
    int _x = _coordinates[0].X;
    int _y = _coordinates[0].Y;
    Pen _pen = new Pen(Color.Blue, ipenWidth);

```

```

if (_coordinates.Length == 4)
{
    string _shapeString = "" + _shapeChecker.CheckShapeType(_edgePoint);
    _g.DrawString(_shapeString, _font, _brush, _x, _y);
    _g.DrawPolygon(_pen, ToPointsArray(_corners));
}
// dörtgen boyutu
foreach (Rectangle rc in _rects)
{

    iFeatureWidth = rc.Width;
    double dis = FindDistance(iFeatureWidth);
    _g.DrawString(dis.ToString("N2"), _font, _brush, _x, _y + 60);
}

```

Yuvarlak şeklinin belirlendiği ve PLC'ye uygun çıkışın verildiği kısım:

```

}
if (_shapeChecker.IsCircle(_edgePoint, out _center, out _radius))
{
    plc.Write("M100.3", 1);
    plc.Write("M100.3", 0);
    plc.Write("M100.1", 0);
    plc.Write("M0.2", 0);
    Rectangle[] _rects = _blobCounter.GetObjectsRectangles();

    string _shapeString = "" + _shapeChecker.CheckShapeType(_edgePoint);
    Pen _pen = new Pen(Color.Red, ipenWidth);
    int _x = (int)_center.X;
}

```

```

int _y = (int)_center.Y;
_g.DrawString(_shapeString, _font, _brush, _x, _y);
_g.DrawEllipse(_pen, (float)(_center.X - _radius),
               (float)(_center.Y - _radius),
               (float)(_radius * 2),
               (float)(_radius * 2));

//dörtgen boyutu
foreach (Rectangle rc in _rects)
{
    iFeatureWidth = rc.Width;
    double dis = FindDistance(iFeatureWidth);
    _g.DrawString(dis.ToString("N2"), _font, _brush, _x, _y + 60);
}
}

#endregion

```

BÖLÜM 4 PLC

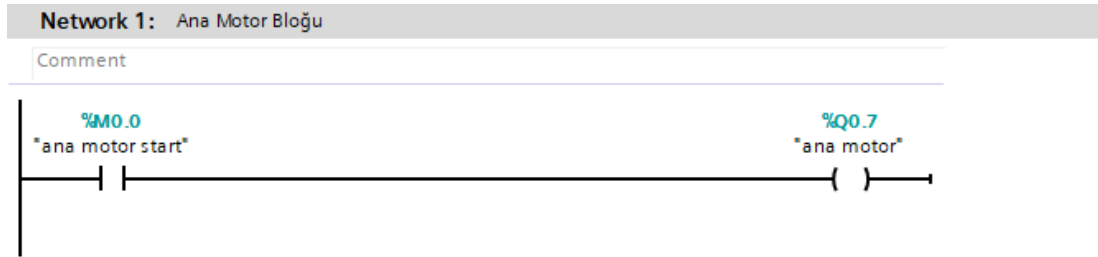
4. SİSTEMİN LADDER DİAGRAMI

4.1 Tag Table

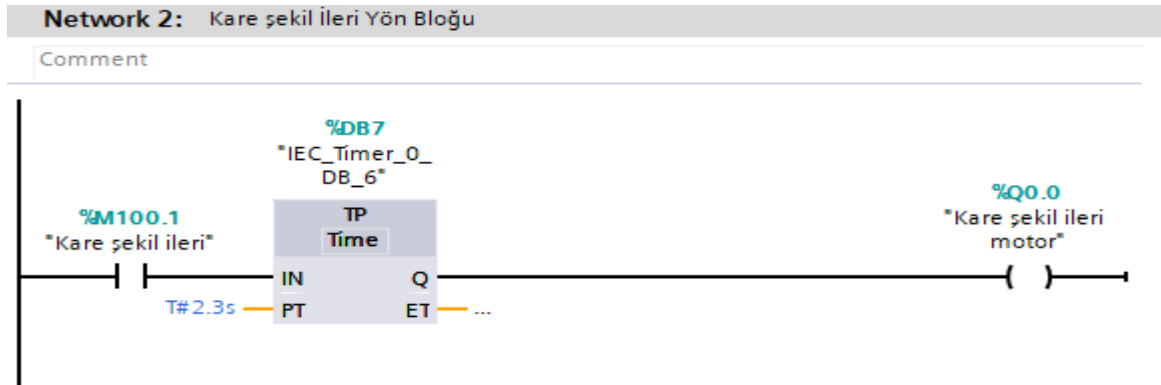
PLC tags								
	Name	Tag table	Data type	Address	Retain	Access...	Write...	Visible...
	ana motor start	default tag table	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	ana motor	default tag table	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Kare şekil ileri	default tag table	Bool	%M100.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Kare şekil ileri motor	default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Kare şekil geri motor	default tag table	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Yuvarlak şekil ileri	default tag table	Bool	%M100.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Yuvarlak şekil ileri motor	default tag table	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Yuvarlak şekil geri motor	default tag table	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Şekil 4 PLC'deki kullanılan malzemelerin isimleri

4.1.1 Ana Blok

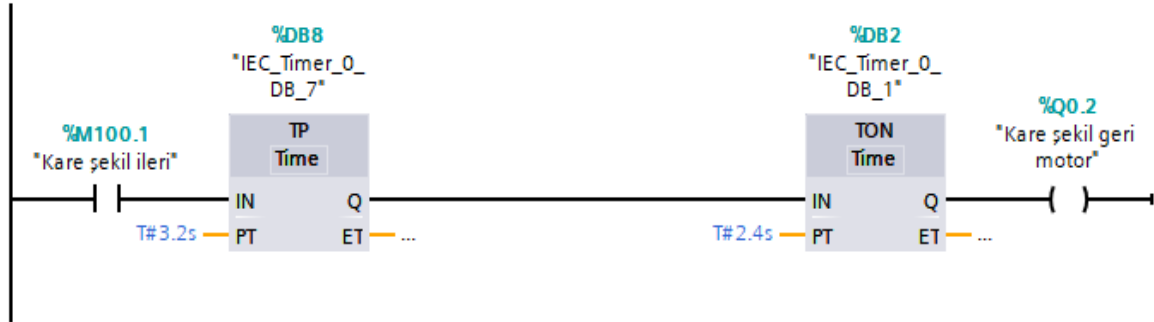


4.1. Motor Yön Kontrol Bloğu



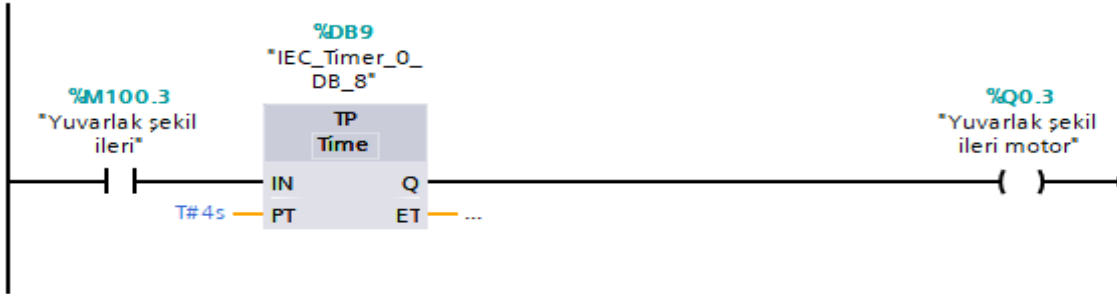
Network 3: Kare Şekil Geri Yön Bloğu

Comment



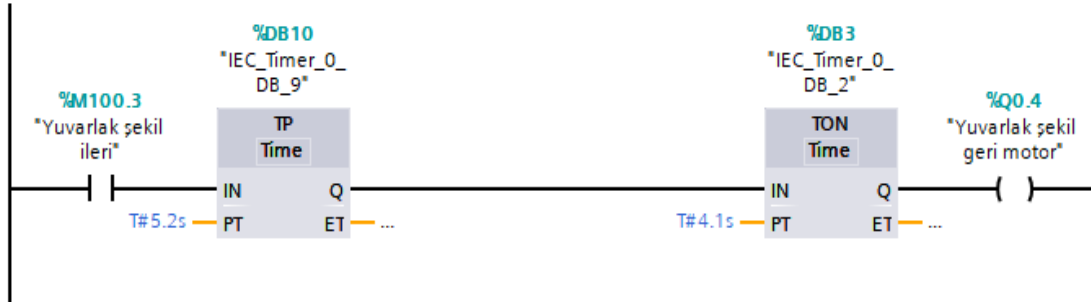
Network 4: Yuvarlak Şekil İleri Yön Bloğu

Comment



Network 5: Yuvarlak Şekil Geri Yön Bloğu

Comment



4.1.2 ÇALIŞMA TAKVİMİ

	MART		NİSAN				MAYIS				HAZİRAN
	3. Hafta	4. Hafta	1. Hafta	2. Hafta	3. Hafta	4. Hafta	1. Hafta	2. Hafta	3. Hafta	4. Hafta	1. Hafta
Çalışma için teorik araştırma	X										
Çalışmada kullanılacak malzeme seçimi için teorik analiz ve piyasa araştırması		X									
Uygun malzeme seçimi ve siparişi		X									
Kullanılan aygıtlar hakkında teorik araştırma ve kütüphane taraması			X			X					
Pratik çalışma			X	X		X	X	X	X		
Bitirme kitapçığı yazımı										X	
Sunum											X

Çalışma Takvimi

Çizelge 1 Çalışma Takvimi Tablosu

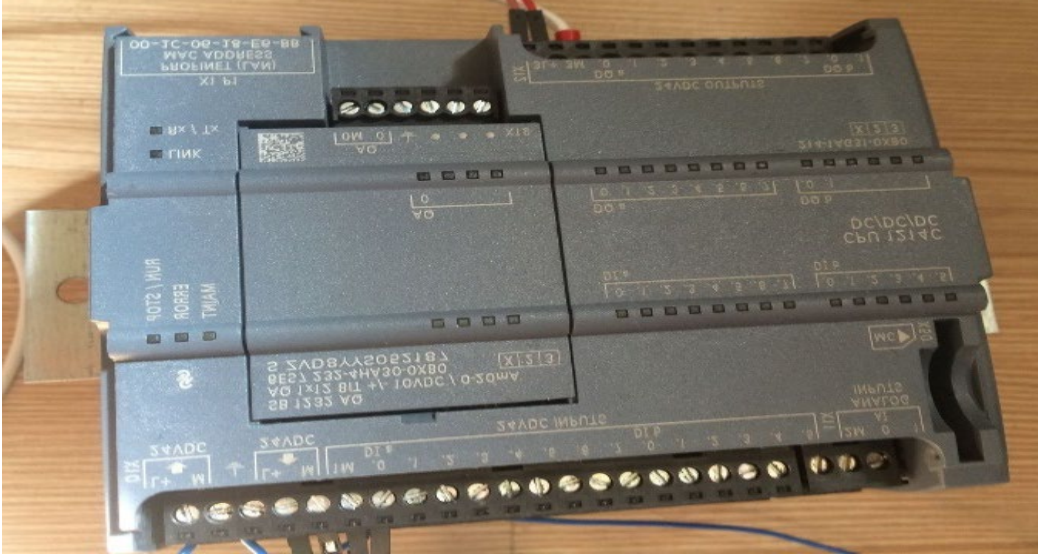
BÖLÜM 5 KULLANILAN MALZEMELER

5.1 İskelet kısım



Şekil 5 İskelet kısım

5.2 PLC – Siemens s7-1200: Projemizde PLC kullanma amacımız, yazılımsal olarak kurduğumuz sistemi proje üzerinde otomatik olarak gerçekleştirmektir.



Şekil 6 PLC

5.3 24V DC Motor: Projemizde toplam 3 adet 24 Volt DC motor kullanıldı. Motorlardan 1 tanesi bandın hareketini sağlamak, 2 tanesi cisimleri kutulara itecek kapıları hareket ettirmek için kullanıldı.



Şekil 7 DC Motor

5.4 Güç Kaynağı: Projemizde 1 adet 24 Volt çıkışlı DC güç kaynağı kullanıldı. Güç kaynağının amacı, PLC'nin enerji ihtiyacını karşılamaktır.



Şekil 8 Güç Kaynağı

5.5 Rôle: Projemizde 1 adet koruma rölesi kullanıldı. Rölenin amacı PLC ve diğer elektronik malzemeleri aşırı akım ve gerilimlere karşı korumaktır.



Şekil 9 Rôle

5.6 RGB Kamera: Projemizde 1 adet RGB Kamera kullanıldı. Cisimlerin renk ve şeklinin belirlenmesini sağlayarak C# programında işlenmesini sağlayacaktır.

5.7 Butonlar: Projemizde 2 adet buton kullanıldı. Butonlardan biri başlat butonu, diğeri ise durdur butonu olarak kullanıldı.



Şekil 10 Butonlar

5.8 Bant: Projemizde 2 metre uzunluğunda bant kullanıldı. Bandın amacı ise konveyör bant işlevi görerek cisimlerin gerekli yerlere ulaşmasını sağlamaktır.

5.9 Kapılar: Projemizde 2 adet 14cm uzunluğunda kapı kullanıldı. Kapıların amacı ise, cisimlerin çarkın içine düşmesini sağlamaktır.



Şekil 11 Kapılar ve bant

5.10 Kutu: Projemizde 20cmx20cmx10cm boyutlarında 1 adet kutu kullanıldı. Program tarafından kriterlere uymadığı tespit edilen cisimlerin ayrılacağı kutudur.

BÖLÜM 6

6. SONUÇ VE ÖNERİLER

Bu çalışmamızda plc ile c# birlikte kullanarak cisimlerin ayrıştırılmasını mümkün olduğunu gözlemleyerek deneyim sahibi olduk. Plc de DC motorların röle kullanılarak ileri-geri yönde çalıştırılmasını sağladık.

Öneri olarak; Kodların daha iyi geliştirilerek daha fazla şekillerin ve renklerin ayrıştırılmasını sağlayabilirdi. Ayrıca röle devresinde breadboard üzerine kurmak yerine kontaktör kullanılabilirdi. Plc de aynı anda iki çıkış kullanılamadığı için ladder diyagramının buna göre yazılması ve sürekli çalışacak olan konveyör bant motorunun harici bir güç kaynağı ile beslenmesi sağlandı. Kullanılan malzemeler daha kaliteli olabilirdi. Projede kullanılacak motorların kullanım amaçlarına dikkat edilmelidir.

KAYNAKÇA

<http://www.sandipointe.com/automotive/reverse-polarity-switch>

<https://factoryio.com/docs/tutorials/siemens/sample-s7-plcsim-v13/>

<https://www.codeproject.com/Articles/826377/Rapid-Object-Detection-in-Csharp>

https://www.aforgenet.com/articles/shape_checker/

<https://www.codeproject.com/Articles/616262/PLC-Communication-Using-NET>

<https://www.youtube.com/watch?v=tYTjNG8YL-c&t=1379s>

https://www.youtube.com/watch?v=0MPXaDmgTyA&list=PLgWzXjmTxAVPpN_GrYCLno1vFNE8LQjlA&index=6

<http://www.bilimsel.com.tr/c-dilinde-s7-net-ve-snap7-kutuphaneleri-ile-s71200-plc-haberlesme/>

<https://www.kontrolkalemi.com/>

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : SÜRÜCÜ ,YUNUS
Uyruğu : TC
Doğum tarihi ve yeri : ADIYAMAN /01.11.1993
Telefon : 05347188215
e-mail :ysurucu542@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lise	Adıyaman Atatürk Üniversitesi	2012

Yabancı Dil İNGİLİZCE

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KANIK ,HAKAN
Uyruğu : TC
Doğum tarihi ve yeri : İSTANBUL /01.03.1994
Telefon : 05370530531
e-mail :hakan.kanik.34@gmail.com

Eğitim

Derecesi	Eğitim Birimi	Mezuniyet Tarihi
Lise	Kadriye Moroğlu Lisesi	2012

Yabancı Dil İNGİLİZCE

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ALTAY ,İLKAY
Uyruğu : TC
Doğum tarihi ve yeri : İSTANBUL /20.05.1995
Telefon : 0545940 7612
e-mail : ilkayaltaye-em@outlook.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lise	Gazi Emet Anadolu Lisesi	2013

Yabancı Dil İNGİLİZCE ,ALMANCA,

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÇORUMLUOĞLU ,FATİH ERKAM
Uyruğu : TC
Doğum tarihi ve yeri : NewCastle/17.04.1996
Telefon : 05396958419
e-mail : fenererkam@gmail.com

Eğitim

Derecesi	Eğitim Birimi	Mezuniyet Tarihi
Lise	Şehit Ali Karaoğlan Lisesi	2014

Yabancı Dil İNGİLİZCE ,ALMANCA,