

Proje Alanı : **Matematik- Bilgisayar**
Proje Adı : **POLYCODE 4.0**

Özet:

Projemiz, tek tip programlamanın ötesinde birçok programlama (kodlama) çeşidini ve birçok teknolojiyi kullanan, Endüstri 4.0 ve milli sanayi hamlesi için eğitimde rehber olacak, sıra dışı bir eğitim setinin imalatı ve kodlamasıdır.

Literatür taraması sonunda, setin 3D modellemesi SolidWorks ile tasarlandı. Gerekli parçalar üretim teknolojileri (3D Printer, Torna, Freze gibi) kullanılarak üretildi. Setin sensör, motor, yürüyen bant ve buton gibi donanımları PLC'ye bağlandı. PLC programı yazılarak sistem test edildi. Setin bilgisayar ile kontrolü için C# ile yazılım yazıldı. Form ekranında simülasyon için setin görüntüsü modellendi ve anlık olarak senkron bir SCADA yazılımı yazılarak sistem görseli simule edildi. SQL veri tabanında ürün değerleri ve zaman bilgisi saklandı. ASP.NET ile tek sayfalık bir web uygulaması ile veri tabanından çekilen veriler anlık olarak tabloda ve grafikte gösterildi. Arttırılmış gerçeklik uygulaması için Artoolkit Kütüphanesi kullanıldı. Ürün ve zaman bilgileri veri tabanından çekilerek analiz ve raporlama için Excel'e aktarıldı. Robotik kodlama için Arduino elektronik kartı kullanıldı. Ortamın sıcaklık ve nem bilgisini gösteren bir uygulama sete eklendi. AForge.NET görüntü işleme kütüphanesi ile iş sağlığı ve güvenliğine vurgu yapan, web cam kullanan bir güvenlik yazılımı yazıldı. Son olarak ta Orange Pi gömülü sistem kartı kullanılarak setin kamera görüntüsü wifi ve IP numarası ile cep telefonu ve ekranda izlendi.

Kullanılan programlama çeşitleri:

- ✓ PLC Programlama
- ✓ Nesnelerin interneti (Bant - PLC - PC Haberleşmesi)
- ✓ C# Programlama
- ✓ Web programlama
- ✓ Veri Tabanı Sorgulama Dili
- ✓ Excel'e Kayıt ve Raporlama
- ✓ Arttırılmış Gerçeklik Uygulaması
- ✓ Robotik Kodlama
- ✓ Orange Pi ile görüntü aktarma
- ✓ Görüntü işleme yazılımı

Anahtar Kelimeler: Kodlama,Endüstri4.0, Kodlama, PLC, SCADA, Arttırılmış Gerçeklik, Programlama

Amaç:

Projemiz, birçok kodlama çeşidini, üretim metot ve konularını içermesi sebebiyle etkili bir kodlama eğitimine güzel bir materyal olmuştur. Öğrenci Endüstri 4.0 kavramını ve çok disiplinli teknolojileri kullanmayı bu proje ile kavrayabilecektir. Disiplinler arası çalışmaya ve birden çok teknolojinin kullanımına örnek olacak, milli teknoloji hamlesi ve Endüstri 4.0 adımları için eğitime referans olacak bir proje amaçlanmıştır.

Mevcut kodlama kursları ve eğitimleri, tam olarak disiplinler arası çalışmayı kapsamamaktadır. Halbuki mili teknoloji hamlesi ve sanayi 4.0 adımları için üretime odaklı, milli üretimi amaçlayan, katma değer değeri olan yazılımlar yapılmalıdır. Tek düze kodlama

eğitimleri ile öğrenciler robotik alanında bir düşünceye varabilse de birçok teknolojiyi birleştirememektedir. Bu projenin amacı birçok kodlama çeşidini beraber kullanarak üretim odaklı yazılımı destekleyecek bir eğitim farkındalığını oluşturmaktır.

Takım çalışması ve 21.yy becerilerini de kazandırması amaçlanan projemiz, öğrencileri üretim odaklı yazılım üzerine yönlendirecektir.

1. Giriş

Endüstri 4.0, modüler yapıllı akıllı fabrikalarda, fiziksel işlemleri siber-fiziksel sistemler ile izleyerek, nesnelerin birbirleriyle ve insanlarla iletişime geçmesini ve bu sayede de merkezi olmayan kararların verilmesini hedefliyor.

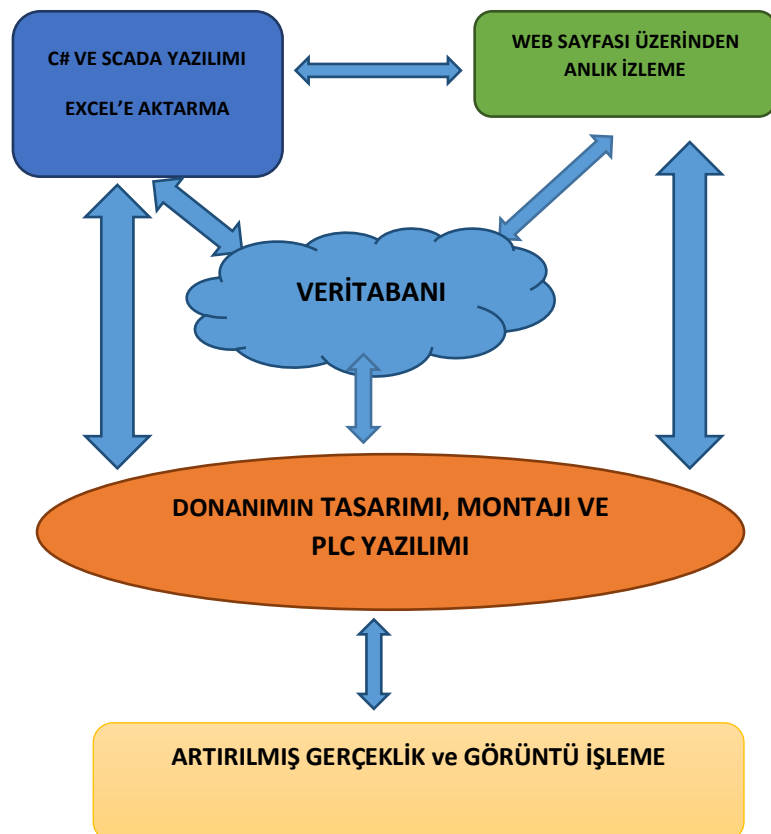
Dünya dijitalleşirken, bu teknolojileri üretmek yerine dışarıdan alan ve dolayısıyla dışarıya bağımlı kalan ülkelerin, dijital dünyayı tasarlayan ve üreten ülkelerin bir anlamda sömürgesi haline geleceği açıktır. O nedenle fabrikaların bu yeni anlayışı benimsemesi ve Endüstri 4.0 vizyonuna uygun hale getirilmesi yanında mutlaka eğitimde de gerekli düzenlemelerin yapılması ve bu vizyona uygun eğitim materyallerinin kullanılması gerekiyor.

PolyCode4.0 projesi, isminden de anlaşılacağı üzere birçok kodlama çeşidini içinde barındırır. PLC programlama, görsel programlama, robotik kodlama, arduino kodlama, web programlama, görüntü işleme gibi birçok alanda kodlama uygulamasını içeren projemiz minimize bir fabrika ortamını da simule edecektir. Özgün bir çalışma olan PolyCode 4.0 projesi kodlama alanında Endüstri 4.0 devrimi eğitimlerine farklı bir bakış açısı katacaktır.

2. Yöntem

Projenin yapımı altı aşamadan oluşmuştur. İlk aşama donanımın tamamlanması, ikinci aşama ise form uygulama yazılımının yazılması ve testi, üçüncü aşama web sayfası ve veri tabanı teknolojilerinin kullanımı, dördüncü aşama artırılmış gerçeklik uygulaması, beşinci aşama Arduino ve Orange Pi kullanma, altıncı aşama görüntü işleme kütüphanelerinin kullanılmasıdır.

2.1. Projenin Yapım Basamakları



a) Tasarım, Montaj ve PLC Yazılımı

- a.1. Solidworks ile tasarımının yapılması
- a.2. Solidworks ile setin 3D montajının yapılması
- a.3. Setin parçalarının montajının bitirilmesi
- a.4. PLC yazılımının tamamlanması

b) C# form uygulama yazılımının yazılması

- b.1. Form tasarımının yapılması
- b.2. C# programının yazılması
- b.3. SCADA yazılımı, sistemin simülasyonu ve anlık benzetimi

c) Web ve veri tabanı teknolojilerinin kullanımı

- c.1. SQL veri tabanının yapılandırılması
- c.2. Ürün ve zaman bilgilerinin veri tabanına aktarılması ve formda görüntülenmesi
- c.3. ASP.NET ortamında C# dili ile web sayfası oluşturma
- c.4. Web sayfası üzerinde verileri anlık gösterme ve grafik
- c.5. Excel'e veri tabanından veri aktarımının gerçekleştirilmesi

d) Artırılmış gerçeklik uygulaması

- d.1. Google SketchUp ile 3D tasarımın indirilmesi
- d.2. Pattern(desen) resimlerin sete eklenmesi ve uygulama

e) Arduino ve Orange Pi kullanımı

- e.1. Arduino tabanlı uygulama geliştirme
- e.2. Orange Pi ile görüntü aktarımı

f) AForge.Net Görüntü İşleme Kütüphanesi Kullanımı

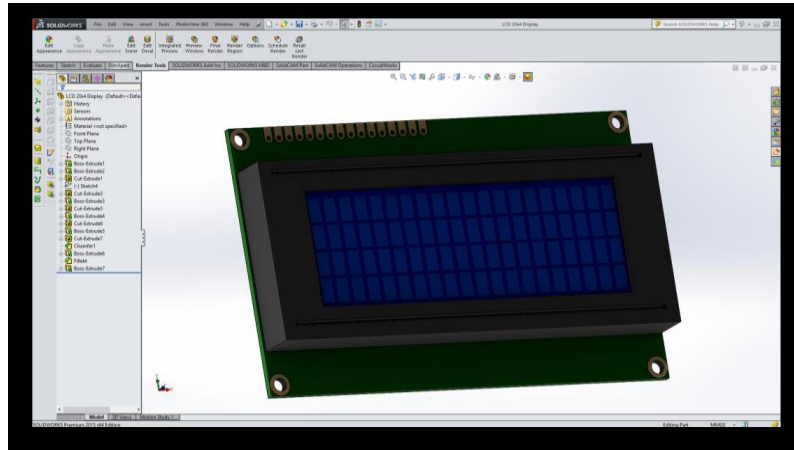
- f.1. AForge.Net ile görüntü işleme yazılımı

3. Bulgular ve Gerçekleme

3.1. Tasarım, Montaj ve PLC yazılımının gerçekleştirilmesi

a.1. Solidworks ile parçaların tasarımının yapılması

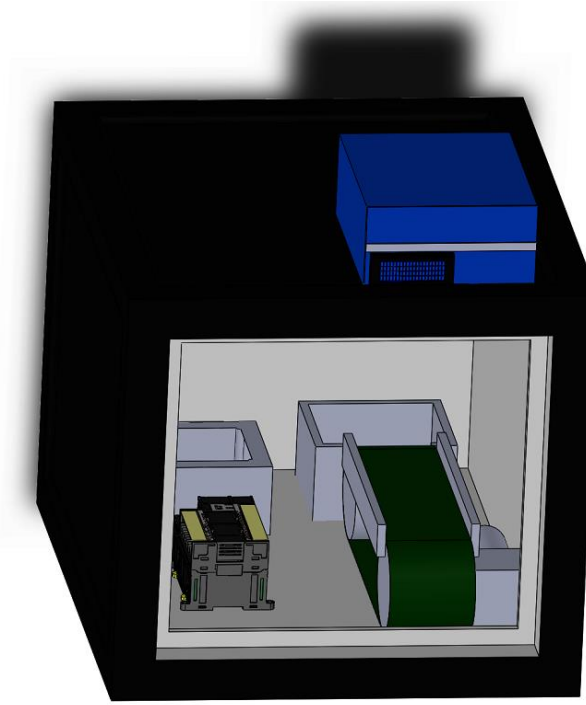
Bu aşamada setin parçaları Solidworks programında tek tek tasarlandı (Görsel-1)



Görsel-1. Setin parçalarının teknik resim çizimleri, 3D tasarımları ve 3D baskısı.

a.2. SolidWorks ile setin 3D montajının yapılması

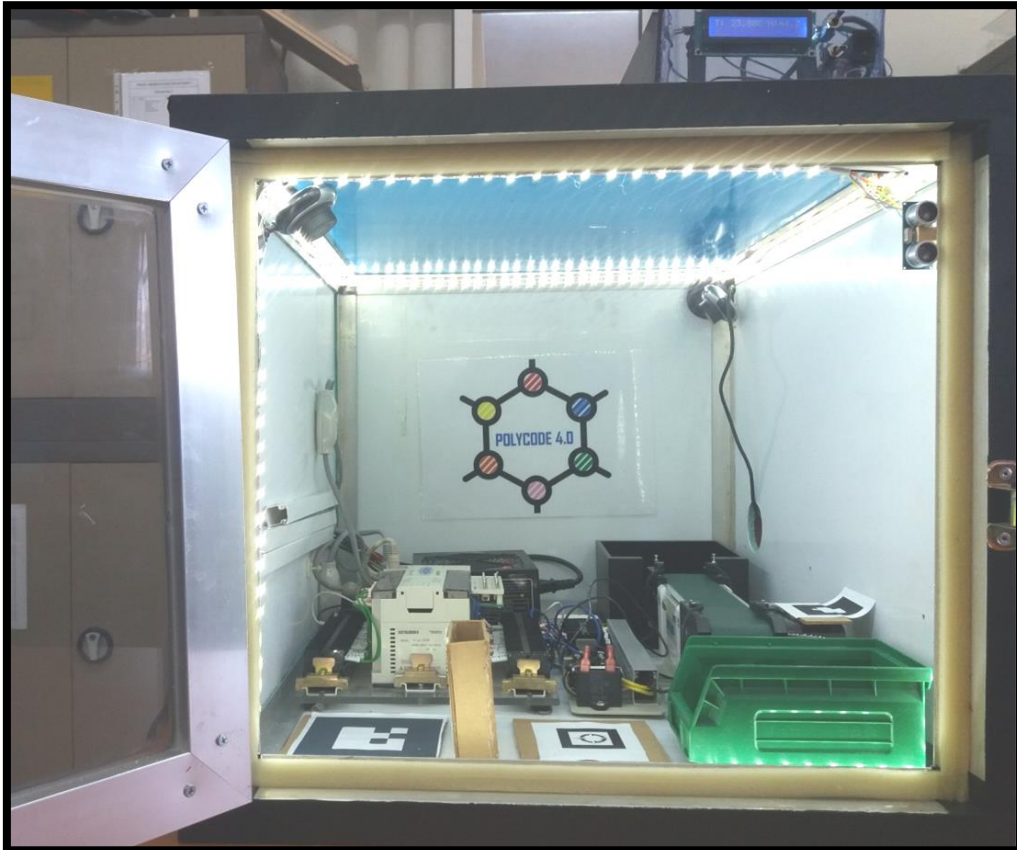
Bu aşamada setin parçaları Solidworks ile birleştirildi. (Görsel-2)



Görsel-2. Setin 3D modellemesi

a.3. Setin parçalarının montajının bitirilmesi

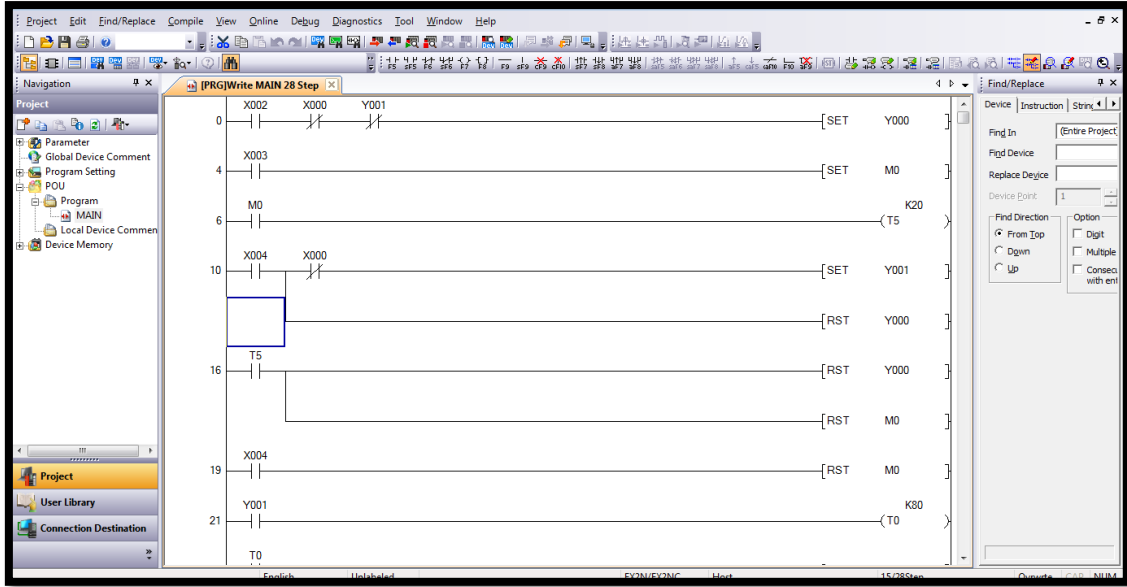
Bu aşamada setin tüm parçaları montajlandı. (Görsel-3)



Görsel-3.Setin montajlanması

a.4. PLC yazılımının tamamlanması

Bu aşamada önce giriş çıkışlar belirlendi. Sonra Mitsubishi PLC yazılımı ladder diyagram ile yazıldı (Görsel-4).



Görsel-4.PLC programından parçalar.

3.2. C# form uygulama yazılımının yazılması

b.1. Form tasarımının yapılması

Bu aşamada Visual Studio.NET ortamında Windows form tasarımı yapıldı. (Görsel-5)

Pil	Role	Zaman
3	3	09.01.2020 14:
2	3	09.01.2020 14:
2	2	09.01.2020 14:
1	2	09.01.2020 14:
1	1	09.01.2020 14:
1	0	09.01.2020 14:
11	3	30.12.2019 15:
10	3	30.12.2019 15:
10	2	30.12.2019 15:
10	1	30.12.2019 15:
10	0	30.12.2019 15:
9	0	30.12.2019 15:
8	0	30.12.2019 15:
7	0	30.12.2019 15:
6	0	30.12.2019 15:
5	0	30.12.2019 15:

Görsel-5.Form Tasarımı

b.2.C# programının yazılması

Bu aşamada öncelikle plc ve yürüyen banda Mitsubishi MX Component kütüphanesi ile erişim sağlandı. Aşağıdaki kodlar bu aşamayı gösteriyor:

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        axActProgType1.ActCpuType = 0x205;
        axActProgType1.ActUnitType = 0x0F;
        axActProgType1.ActProtocolType = 0x04;
        axActProgType1.ActNetworkNumber = 0;
        axActProgType1.ActStationNumber = 255;
        axActProgType1.ActUnitNumber = 0;
        axActProgType1.ActConnectUnitNumber = 0;
        axActProgType1.ActIONumber = 0;
        axActProgType1.ActPortNumber = 15;
        axActProgType1.ActBaudRate = 19200;
        axActProgType1.ActDataBits = 0x0007;
        axActProgType1.ActParity = 0x0002;
        axActProgType1.ActStopBits = 0x00000;
        axActProgType1.ActControl = 0x08;
        ireturncode = axActProgType1.Open();
        if (ireturncode == 0)
        {
            ovalShape4.BackColor = Color.Green;
            MessageBox.Show("Bağlantı Kuruldu", Name, MessageBoxButtons.OK,
            MessageBoxIcon.Information);

            timer4.Enabled = true;
        }
    }
    catch (Exception exeption)
    {
        MessageBox.Show(exeption.Message, Name, MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return;
    }
}
```

Gelen verilere göre PLC giriş ve çıkışları anlık olarak kontrol edildi. Sensörler okundu ve banda alınan ürünler ayıklandı kontrol edildi. Aşağıda kodlardan bir parça görülmektedir:

```
private void timer4_Tick(object sender, EventArgs e)
{
    int veri1, veri2, veri3, veri4, veri_oku2, veri_oku3, veri_oku4, veri_oku7, veri_oku1,
    veri_oku5, veri5, veri6, veri7, veri_oku6;
    veri_oku4 = axActProgType1.ReadDeviceRandom("X4", 1, out (veri4));
    veri_oku3 = axActProgType1.ReadDeviceRandom("X3", 1, out (veri3));
    veri_oku2 = axActProgType1.ReadDeviceRandom("X2", 1, out (veri2));
```

```

veri_oku1 = axActProgType1.ReadDeviceRandom("Y0", 1, out (veri1));
veri_oku5 = axActProgType1.ReadDeviceRandom("Y1", 1, out (veri5));
veri_oku6 = axActProgType1.ReadDeviceRandom("Y2", 1, out (veri6));
veri_oku7 = axActProgType1.ReadDeviceRandom("Y3", 1, out (veri7));

if (veri1 == 1) { lineShape4.Visible = true; lineShape5.Visible = true;
lineShape6.Visible = true; lineShape7.Visible = false; lineShape8.Visible = false;
lineShape9.Visible = false; label8.Text = "BANT İLERİ GİDİYOR"; }
if (veri5 == 1) { lineShape7.Visible = true; lineShape8.Visible = true;
lineShape9.Visible = true; lineShape4.Visible = false; lineShape5.Visible = false;
lineShape6.Visible = false; label8.Text = "BANT GERİ GELİYOR"; }
if ((veri5 == 0) && (veri1 == 0)) { lineShape4.Visible = false; lineShape5.Visible =
false; lineShape6.Visible = false; lineShape7.Visible = false; lineShape8.Visible = false;
lineShape9.Visible = false; label8.Text = "BANT DURDU"; label7.Text = ""; }
if (veri3 == 1) { ovalShape6.BackColor = Color.Green; } else {
ovalShape6.BackColor = Color.Red; }
if (veri4 == 1) { ovalShape7.BackColor = Color.Green; } else {
ovalShape7.BackColor = Color.Red; }

if (veri3 == 1)
{

if(veri4==1)
{

ovalShape6.BackColor = Color.Green;

label7.Text = ("PİL ALGILANDI..."); ovalShape8.Visible = false;
rectangleShape4.Visible = true; ovalShape7.BackColor = Color.Green;
Pil = Pil + 1; textBox1.Text = Pil.ToString(); timer6.Enabled = true;

}

else

{

label7.Text = ("RÖLE ALGILANDI..."); ovalShape8.Visible = true;
rectangleShape4.Visible = false;
Role = Role + 1; textBox2.Text = Role.ToString(); timer6.Enabled = true;
}
}

```

Böylece setin PC ile kontrolü sağlanmış oldu.

b.3. SCADA yazılımı ile sistemin simülasyonu ve anlık benzetimi

Bu aşamada sıra SCADA simülasyonunun yapılmasına geldi. Form ekranında (Görsel-5) görüldüğü üzere setin görüntüsü çizildi. Giriş ve çıkışlar tek tek forma işaretlendi. Böylelikle sistemin PC üzerinden adım adım testi gerçekleşti. PLC bağlantısı yapılarak PLC giriş ve çıkışlarına göre anlık benzetim simule edildi. Yazılan kodlardan bir parça:

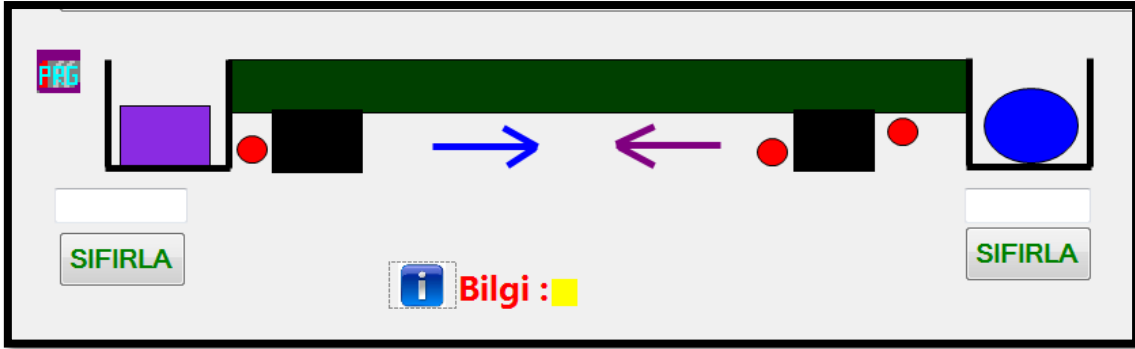
```

        if (veri2 == 1)
        {
            ovalShape5.BackColor = Color.Green;
            ovalShape6.BackColor = Color.Red;
            ovalShape7.BackColor = Color.Red;
            rectangleShape4.Visible = false;
            // Thread.Sleep(2000);
            label7.Text = "";
        }

        if (veri2 == 0)
        {
            ovalShape5.BackColor = Color.Red;
        }
        if (veri4 == 1)
        {
            ovalShape7.BackColor = Color.Green; //Thread.Sleep(1000);
        }
    }
}

```

Aşağıda Form ekranının SCADA bölümü görülmüyor (Görsel-6)

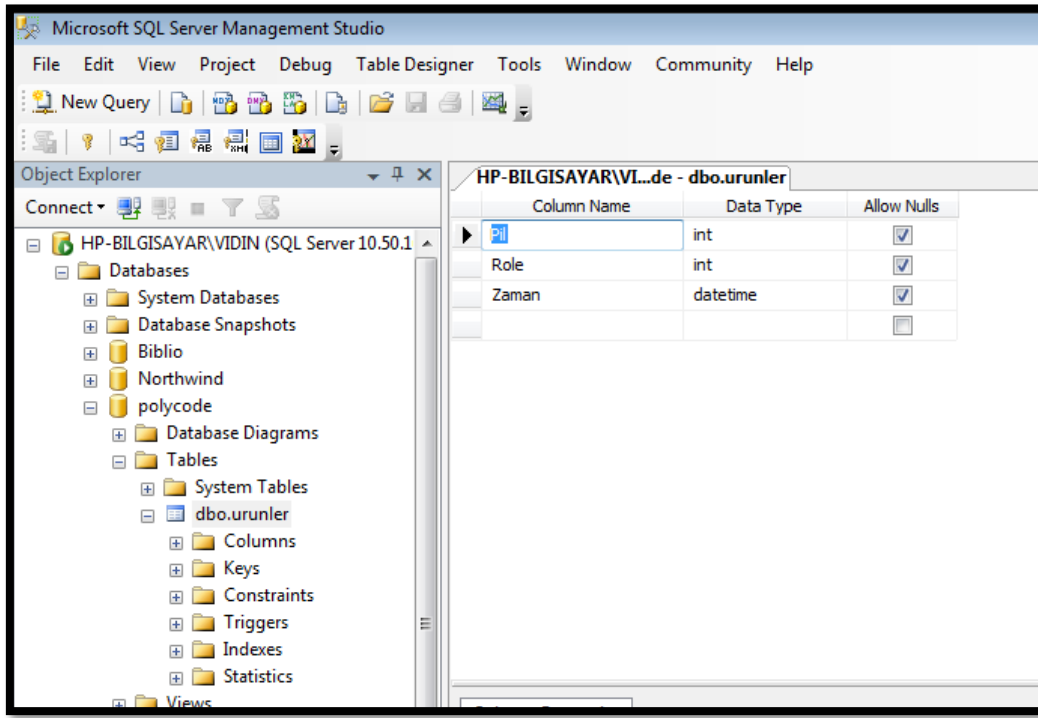


Görsel-6.SCADA Ekranı.

3.3. Web ve veritabanı teknolojilerinin kullanımı

c.1.SQL veritabanının yapılandırılması

SQL veri tabanı tablosu oluşturuldu. (Görsel-7)



Görsel-7.SQL Veritabanı Tablosu

Alınan veri tabanına erişim için aşağıdaki kodlar ile bağlantı sağlandı.

```
private void listele()
{
    SqlConnection bag=baglanti();
    bag=new SqlConnection("Data Source=HP-BILGISAYAR\\VIDIN;Initial
Catalog=polycode;Integrated Security=True");
    da=new SqlDataAdapter("Select *from urunler ORDER BY Zaman DESC",bag);
    ds = new DataSet();
    bag.Open();
    da.Fill(ds,"urunler");
    dataGridView1.DataSource=ds.Tables["urunler"];
}
```

c.2. Ürün ve zaman bilgilerinin veri tabanına aktarılması ve formda görüntülenmesi

Ürün verileri ve zaman bilgisi aşağıdaki kodlar ile SQL veri tabanına atıldı.

```
private void listele()
{
    SqlConnection bag=baglanti();
    bag=new SqlConnection("Data Source=HP-BILGISAYAR\\VIDIN;Initial
Catalog=polycode;Integrated Security=True");
    da=new SqlDataAdapter("Select *from urunler ORDER BY Zaman DESC",bag);
    ds = new DataSet();
    bag.Open();
    da.Fill(ds,"urunler");
    dataGridView1.DataSource=ds.Tables["urunler"];
}
```

Formda DataGridView nesnesinde (Görsel-8) veriler izlendi.

	Pil	Role	Zaman
►	11	3	30.12.2019 15:22
	10	3	30.12.2019 15:22
	10	2	30.12.2019 15:22
	10	1	30.12.2019 15:22
	10	0	30.12.2019 15:21
	9	0	30.12.2019 15:21
	8	0	30.12.2019 15:21
	7	0	30.12.2019 15:21
	6	0	30.12.2019 15:21
	5	0	30.12.2019 15:21

VERİ TABANINI TEMİZLE

Görsel-8.Form ekranında SQL veritabanından çekilen anlık veriler.

c.3. ASP.NET ortamında C# dili ile web sayfası oluşturma

Bu aşamada ASP.NET ortamında bir adet web sayfası yapılmasına karar verildi.

```
<% @ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="polycode_4._0.WebForm1" %>
```

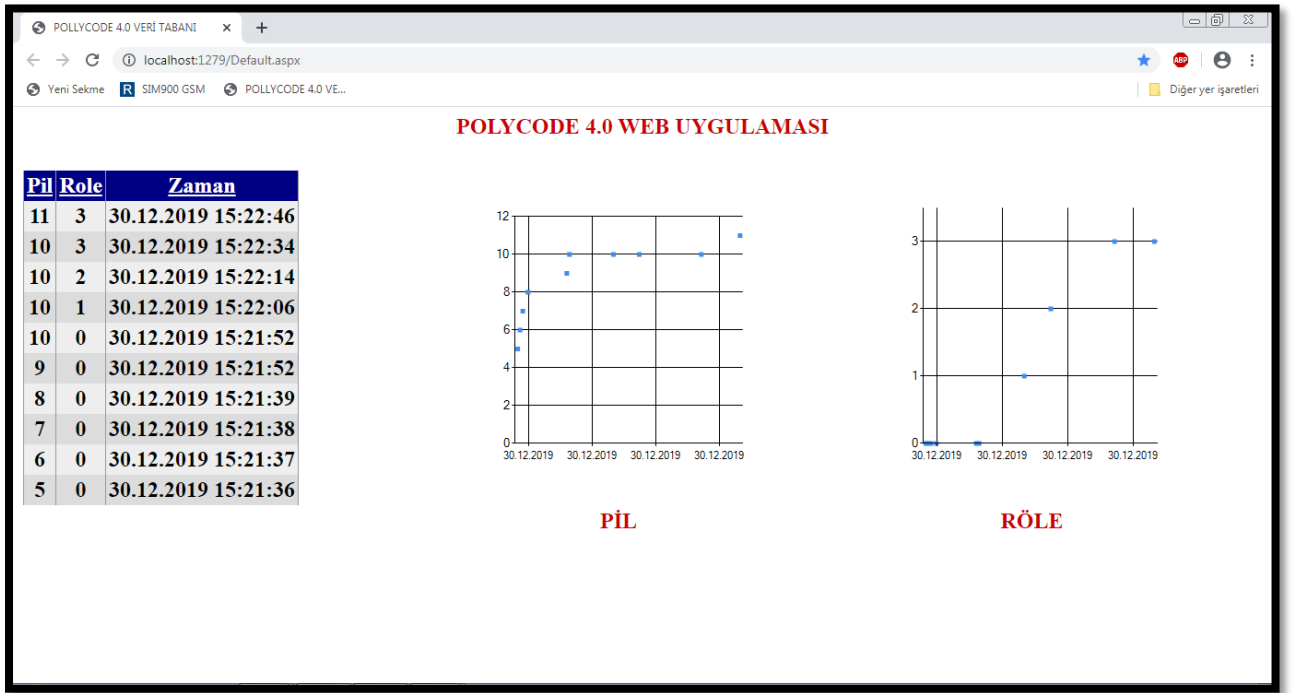
```
<% @ Register assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
namespace="System.Web.UI.DataVisualization.Charting" tagprefix="asp" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>POLLYCODE 4.0 VERİ TABANI</title>
<style type="text/css">
.style1
{
color: #CC0000;
font-size: x-large;
}
</style>
```

c.4. Web sayfası üzerinde verileri anlık gösterme ve grafik

SQL veri tabanından çekilen veriler localhost'ta sayfa üzerinde listelendi ve beraberinde grafik gösterimi yapıldı. (Görsel-9)



Görsel-9.Default.aspx sayfası

Bu aşamada SQL veritabanına bağlantı kuruyoruz.Sql sorgusuyla verileri tarihe göre yakından uzağa sıralayarak çekiyoruz.Son olarak ta grafik nesnesi sayfaya ekliyoruz.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:polycodeConnectionString %>"
SelectCommand="SELECT * FROM [urunler] ORDER BY [Zaman] DESC"></asp>
```

c.5. Excel'e veri tabanından veri aktarımının gerçekleştirilmesi

Bu aşamada SQL veritabanından çekilen veriler (Form'da ve web sayfasında izlenen veriler) bir buton yardımıyla çeşitli uygulama, hesaplama ve raporlama işlemleri için Excel' e aktarıldı. (Görsel -10)

```
private void button13_Click(object sender, EventArgs e)
{
    myExcel._Application excel = new myExcel.Application();
    excel.Visible = true;
    object Missing = Type.Missing;
    Workbook workbook = excel.Workbooks.Add(Missing);
    Worksheet sheet1 = (Worksheet)workbook.Sheets[1];
    int StartCol = 1;
    int StartRow = 1;
    for (int j = 0; j < dataGridView1.Columns.Count; j++)
    {
        Range myRange = (Range)sheet1.Cells[StartRow, StartCol + j];
        myRange.Value2 = dataGridView1.Columns[j].HeaderText;
    }
}
```

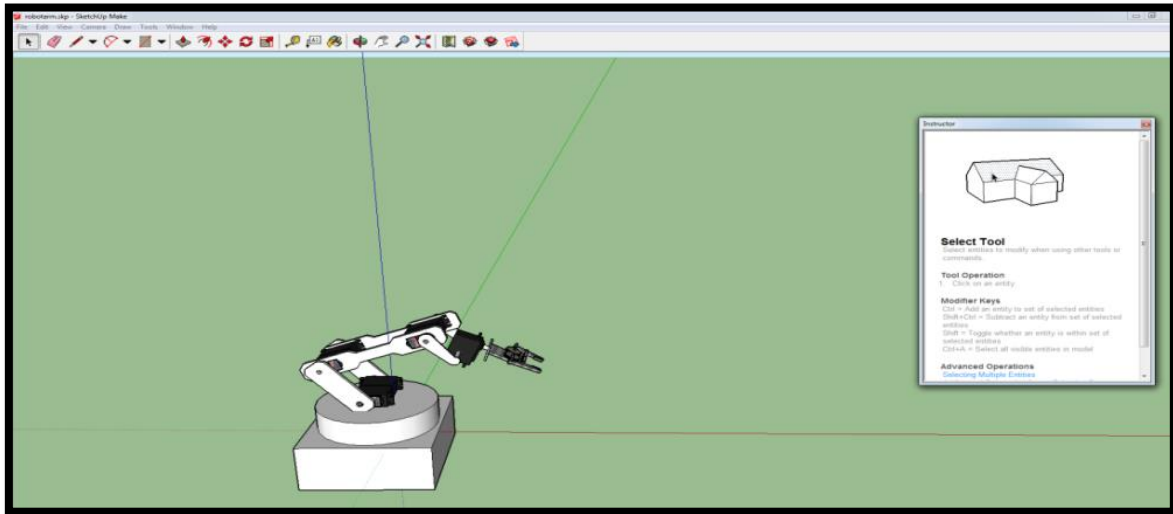
	A	B	C	D	E	F
1	Pil	Role	Zaman			
2		11	3 30.12.2019 15:22:46			
3		10	3 30.12.2019 15:22:34			
4		10	2 30.12.2019 15:22:14			
5		10	1 30.12.2019 15:22:06			
6		10	0 30.12.2019 15:21:52			
7		9	0 30.12.2019 15:21:52			
8		8	0 30.12.2019 15:21:39			
9		7	0 30.12.2019 15:21:38			
10		6	0 30.12.2019 15:21:37			
11		5	0 30.12.2019 15:21:36			
12						
13						
14						
15						

Görsel -10. Verilerin Excel'e aktarımı

3.4 . Artırılmış gerçeklik uygulaması

d.1. Google SketcUp ile 3D tasarımın indirilmesi

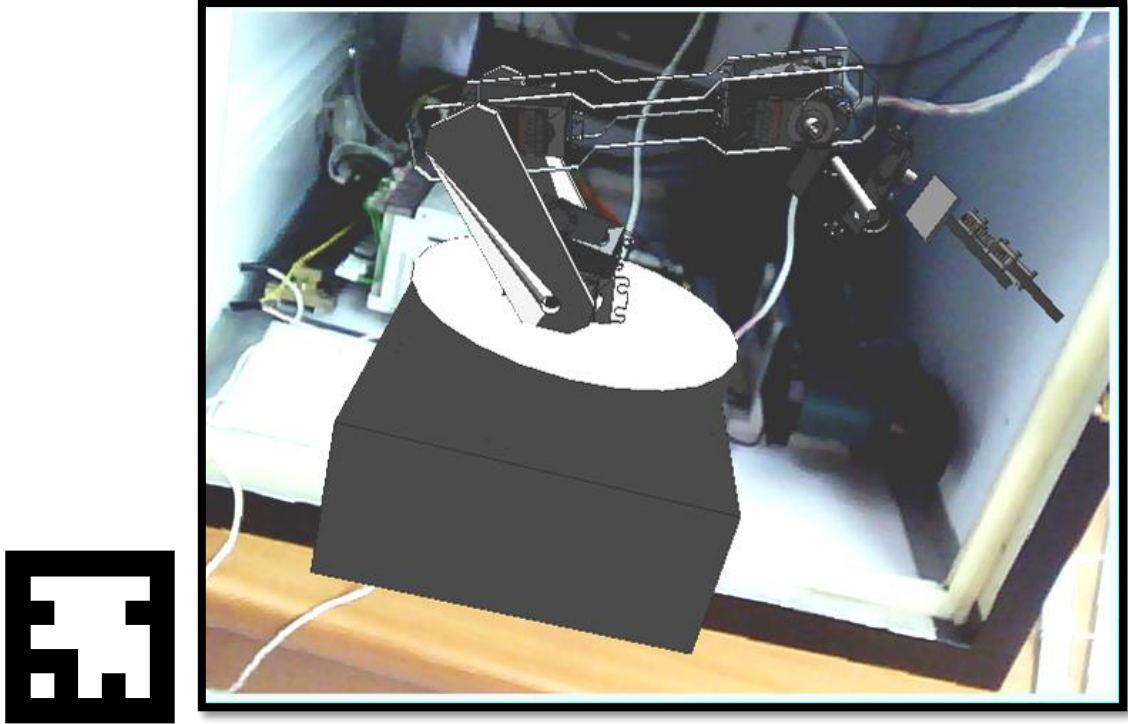
Bu aşamada artırılmış gerçeklik uygulamasında ürünlerin banda alınıp konulmasını temsil etmek için, yürüyen bandın yanına iki adet 3D robot kol eklemek gerekti. Artoolkit kütüphanesi kullanıldı. ARToolKit, programcıların Artırılmış Gerçeklik uygulamalarını kolayca geliştirebilmelerini sağlayan bir yazılımdır. Artırılmış Gerçeklik (AR), bilgisayar tarafından üretilen içeriğin doğal çevreye gömülmesidir ve eğlence, medya, reklamcılık, endüstri ve akademik araştırmalarda pek çok potansiyel uygulama içerir. Google Sketchup programı ile 3D robot kol tasarımı bulundu.c:\Artoolkit\skp klasörüne kopyalandı.Örnek : robotarm.skp(Görsel-11)



Görsel-11. Google Sketchup programında 3D robot kol

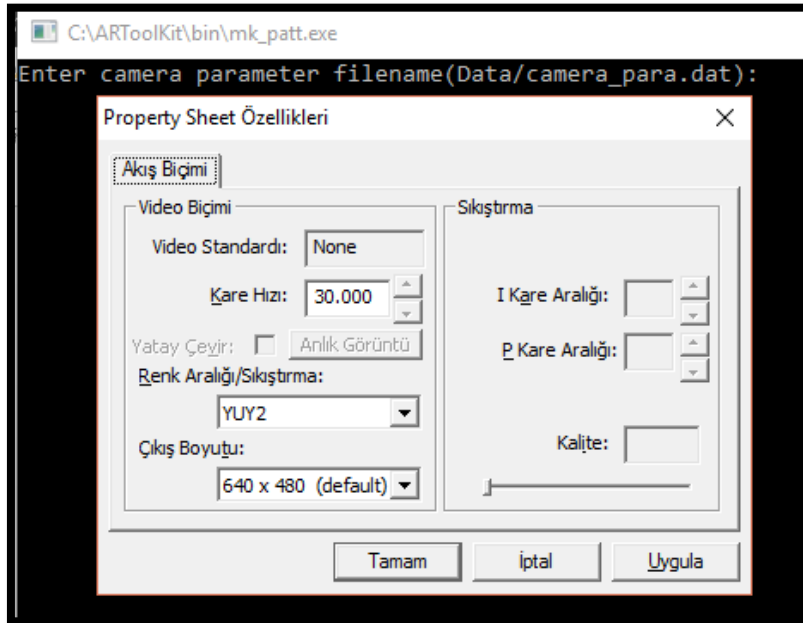
d.2. Pattern resimlerin sete eklenmesi ve uygulama

Pattern (desen) denilen küçük resimler sete yapıştırıldı ve web cam aracılığıyla pattern üstüne 3D robot kol ve motor resmi ile PLC üzerinde de plc yazılımının bir kısmı gösterildi. (Görsel-12)



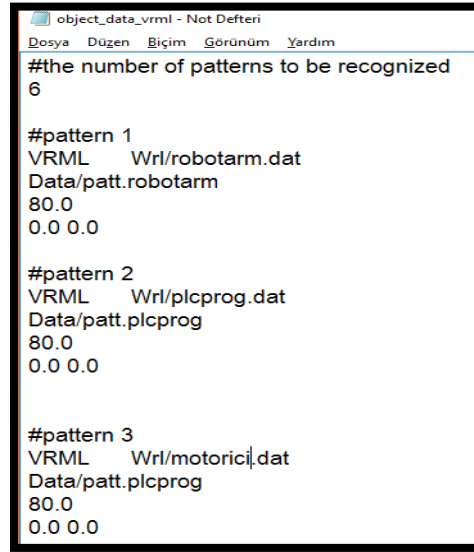
GÖRSEL – 12. Örnek bir pattern(desen) ve set üzerinde 3D robot kol görüntüsü

Bu işlemler için c:\Artoolkit\bin klasörü içindeki mk_patt.exe dosyası ile pattern kamerada yakalandı ve c:\Artoolkit\bin\wrl klasörüne iki dosya olarak kaydedildi. Örnek robotarm.wrl ve robotarm.dat (Görsel-13)



Görsel-13. mk_patt.exe dosyası ile pattern yakalanıyor.

Kaydedilen bu iki dosya ise c:\Artookit\bin\Data klasörü içindeki object.data.vrml dosyası içinde tanımlanır. (Görsel-14). Kaç pattern kullanılacaksa her pattern için aynı işlemler uygulanır ve bu dosyada tanımlanır. Projede 6 adet desen kullanıldı.



```
#the number of patterns to be recognized
6

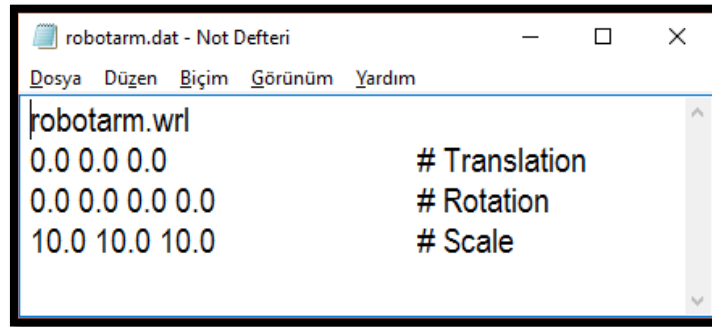
#pattern 1
VRML      Wrl/robotarm.dat
Data/patt.robotarm
80.0
0.0 0.0

#pattern 2
VRML      Wrl/plcprog.dat
Data/patt.plcprog
80.0
0.0 0.0

#pattern 3
VRML      Wrl/motoricil.dat
Data/patt.plcprog
80.0
0.0 0.0
```

Görsel – 14. object_data_vrml dosyası içinde uygulanacak pattern tanımlamaları

Böylelikle ilgili desene ilgili 3D tasarım yerleştirilmiş olur. Konum koordinatları .dat dosyası içinde ilgili .wrl dosyası yazılarak tanımlanır.(Görsel-15)



```
robotarm.wrl
0.0 0.0 0.0      # Translation
0.0 0.0 0.0 0.0  # Rotation
10.0 10.0 10.0   # Scale
```

Görsel-15. 3D objenin pattern yerleştirme koordinatları

3.5. Arduino ve Orange Pi Kullanımı

e.1. Arduino tabanlı uygulama geliştirme

Mini bir fabrika ortamının sıcaklık, nem, güvenlik gibi kavramların ve diğer robotik kodlama uygulamalarında kullanılabilecek sensörlerin yazılım kontrolleri için güncel bir elektronik kart olan Arduino kartı ve sensörler programlandı.

```
int ledPin = 6;

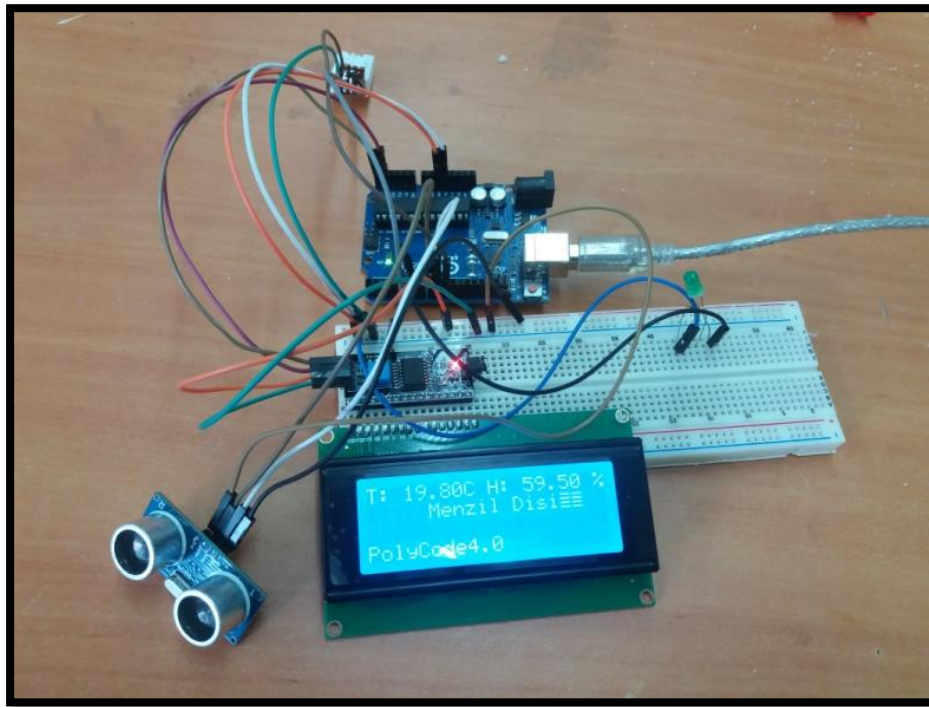
int buzzerPin=7;

//Kütüphaneler
#include <DHT.h>;

#define role_nem 13

#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 4);  
#include <Wire.h>  
#define trigPin 9  
#define echoPin 8      Ultrasonik Mesafe Pin Tanımlamaları  
#define DHTPIN 10  
#define DHTTYPE DHT22  Dht Sensörü Tanımlamalar  
DHT dht(DHTPIN, DHTTYPE);  
void setup(){  
  lcd.begin();  
  pinMode(role_nem, OUTPUT);  //Röle'nin çıkış elemanı olduğunu belirtiyoruz  
  pinMode(ledPin,OUTPUT); //LED'in çıkış elemanı olduğunu belirtiyoruz  
  Serial.begin(9600); //9600 Baundluk bir seri haberleşme başlatıyoruz  
  pinMode(trigPin,OUTPUT); //Ultrasonik Mesafe Sensörü'nün çıkış elemanı olduğunu  
belirtiyoruz  
  pinMode(echoPin,INPUT); dht.begin();  
Programlanan ve yazılımı test edilen devre (Görsel-16), sonrasında sete moontajlandı.
```



Görsel-16. Arduino Kartı ve devresinin resmi

e.2. Orange Pi ile görüntü aktarımı

Fabrika ortamının anlık görüntüsünü izlemek için Orange Pi gömülü sistem kartı ile Motion uygulaması kullanıldı.


```

Welcome to Armbian Bionic with Linux 5.4.8-sunxi

System load: 0.69 0.17 0.06 Up time: 0 min
Memory usage: 13 % of 491MB IP:
CPU temp: 36°C
Usage of /: 6% of 15G

root@orangeplite:~# armbian-config
Warning: Configuration cannot work properly without a working internet connection.
any key to ignore and continue.^CFailed to import the site module

Traceback (most recent call last):
  File "/usr/lib/python3.6/site.py", line 79, in <module>
    root@orangeplite:~# import os
  File "/usr/lib/python3.6/os.py", line 652, in <module>
    from _collections_abc import MutableMapping
  File "/usr/lib/python3.6/_collections_abc.py", line 388, in <module>
    class Collection(Sized, Iterable, Container):
  File "/usr/lib/python3.6/abc.py", line 133, in __new__
    cls = super().__new__(cls, name, bases, namespace, **kwargs)
KeyboardInterrupt
^C
root@orangeplite:~# apt install motion
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
fontconfig fontconfig-config fonts-dejavu-core libavcodec57 libavformat57 libavutil55
libcroco3 libdatrie1 libdrm-common libdrm2 libfontconfig1 libfreetype6 libgdk-pixbuf2.0
libgraphite2-3 libgs1 libharfbuzz0b libjpeg6 libjpeg-turbo8 libjpeg6 libpng3 lame0 libmp
libopenjp2-7 libopenmpt0 libopus0 libpango-1.0-0 libpango-turbo8 libpangocairo-1.0-0 libpangoft2-1.0-0
libshim3 libswaypy105 libsoxr0 libspeex1 libssh-gcrypt-4 libswresample2 libswscale4 li
libtiff5 libtwolame0 libva-drm2 libva-x11-2 libva2 libvdpau libvorbis0a libvorbisenc2 li
libwebp6 libwebpmux3 libx11-6 libx11-data libx264-152 libx265-146 libxau6 libxcb-render
libxext6 libxfixes3 libxrender1 libxvidcore4 libxvbi-common libxvbi0 mysql-common shared
Suggested packages:
libbluray-bdj opus-tools librsvg2-bin speex default-mysql-client postgresql-client
Recommended packages:
libaac0 libadk-ntfs2

```

IP adresi olarak 172.24.1.1:8081 girilerek ekrandan gelen görüntü cep telefonundan ve ekrandan izlendi. (Görsel-17)



Görsel-17.Orange Pi ile Görüntü aktarımı

Orange Pi ile yapılan çalışmanın işlem basamakları:

UYGULAMAYI KURMA

1. Motion uygulamasını ("apt-get install motion") komutu ile kuruyoruz
2. Motion uygulamasını çalıştırabilmek için config dosyasını değiştirmemiz gerekiyor bunun için ("nano /etc/motion/motion.conf") komutu ile nano editorünü kullanarak dosyayı açıyoruz. Ardından "stream_localhost = True" satırını bulup "True" değerini "False" olarak değiştiriyoruz.
3. Bu aşamadan sonra terminale ("motion") komutunu yazmamız yeterlidir. Cihazın bağlı olduğu ağa bağlanan her cihaz ip adresini ve portu "<ip_adress>:8081" yazdıktan sonra cihaza bağlanabilir.

OTOMATİK BAŞLATMA

1. Bunun için motion'un daemon özelliğini kullanıyoruz Bu özelliği açmak için config dosyası açıyoruz("nano /etc/motion/motion.conf") ve "daemon = off" satırını buluyoruz ve "off" değerini "on" ile değiştiriyoruz .
2. 1. maddeyi tamamladıktan sonra "/etc/default/motion" dosyasını ("nano /etc/default/motion") komutu ile açıyoruz ve "off" değerini "on" ile değiştiriyoruz.
3. Cihazı yeniden başlatıyoruz ("reboot").
4. Daha sonra ("systemctl enable motion") komutunu çalıştırıyoruz.
5. Cihazı yeniden başlatıyoruz ("reboot").

Wi-Fi YAYINI OLUŞTURMA

1. Terminale ("armbian-config") yazarak konfigrasyon arayüzünü açıyoruz
2. Sırasıyla Network>Hotspot a giriyoruz
3. Basitleştirilmiş arayüz üzerinden ssid ve şifre belirliyoruz
4. Cihazı yeniden başlatıyoruz ("reboot").

3.6. AForge.NET Görüntü İşleme Kütüphanesi Kullanımı

f.1. AForge.Net ile görüntü işleme yazılımı

İş sağlığı ve güvenliği (İSG) konusunda farkındalık oluşturmak için güvenlik amaçlı kullanışlı bir program arayüzü hazırlandı. Kodlaması AForge.NET Kütüphanesi ve C# ile yapıldı. (Görsel-18,19)

```

// Video Kaynağı Açılıyor
private IVideoSource videoSource = null;
// Hareket Algılama
MotionDetector detector = new MotionDetector(
    new TwoFramesDifferenceDetector( ),
    new MotionAreaHighlighting( ) );
// Algoritmalar
private int motionDetectionType =1;
private int motionProcessingType =0;

// tazeleme için sayıcı kullanımı
private int flash = 0;
private float motionAlarmLevel = 0.015f;

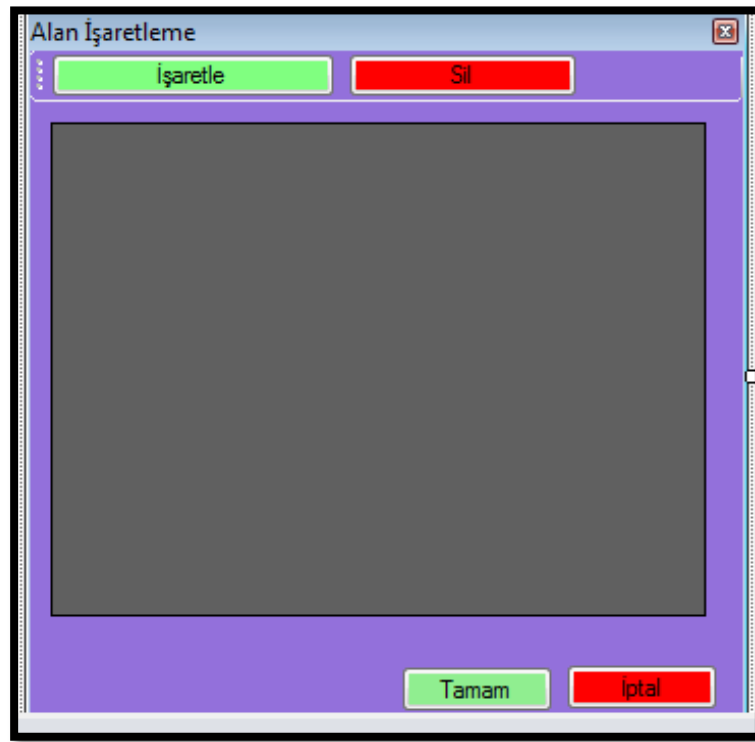
private List<float> motionHistory = new List<float>( );
private int detectedObjectsCount = -1;

// Ana Form
public MainForm( )
{
    InitializeComponent( );
    Application.Idle += new EventHandler( Application_Idle );
}

```



Görsel-18. Görüntü İşleme Arayüz Formu



Görsel-19. Görüntü İşleme Alan İşaretleme Arayüz Formu

4. İş - Zaman Çizelgesi

[illegible]

5. Bulgular

Proje boyunca saptanan bulgular:

- 5.1. Disiplinler arası (Mekanik, Elektrik, Elektronik, Yazılım) çalışma gerçekleştirildi.
- 5.2. Görsel-2 'de görüldüğü üzere SolidWorks kullanılarak 3D tasarım becerilerinde gelişme sağlandı.
- 5.3. Görsel-3 'te görüldüğü üzere mekanik ve elektronik parçalar birleştirildi. Sanayi odaklı düşüncede artış oldu.
- 5.4. Görsel-4'te görüldüğü üzere PLC yazılımları kullanıldı. Böylece üretime yönelik fabrika yazılımlarına karşı ilgi oluştu.
- 5.5. Görsel-5'te ve Görsel'-6'da görüldüğü üzere Görsel Programlama becerisi gelişti. .NET ortamını kullanma ve C# ile port programlama, grafik programlama becerileri kazanıldı.
- 5.6. Görsel-7'de ve Görsel-8'de görüldüğü üzere SQL Server veri tabanı kullanımı, tablo oluşturma ve sql sorgulama dili kullanma gibi veri tabanı olgusu kavrandı.
- 5.7. Görsel-9'da görüldüğü üzere web sayfası tasarımı, HTML, ASP.NET ve veri tabanı-web sayfası etkileşimi konularında gelişme sağlandı.
- 5.8. Görsel-10'da görüldüğü üzere Excel'e veri aktarma ve bu işlemin raporlama ve analiz işlemlerinde önemi algılandı.
- 5.9. Görsel-11'de ve Görsel-12'de görüldüğü üzere arttırılmış gerçeklik uygulamaları, Artoolkit kütüphanesi üzerinde çalışmalar yapıldı.
- 5.10. Görsel-15'de ve Görsel-16'da görüldüğü üzere Arduino ve robotik uygulamalar üzerinde yazılım çalışmaları yapıldı.
- 5.11. Görsel-17'de görüldüğü üzere Orange Pi ile gömülü sistem kullanma, kamera ile görüntü aktarımı gibi konularda çalışmalar yapıldı.
- 5.12. Görsel-18'de ve Görsel-19'da görüldüğü üzere AForge.NET görüntü işleme kütüphanesi kullanma konusunda bilgi sahibi olunarak, yazılım geliştirildi.

6. Sonuç ve Tartışma

- 6.1. Projenin montaj aşamasında detaylara önem verilmiştir. İş güvenliği de önemsenmiştir. Mekanik parçalar Solidworks programında 3D ile çizilmiş işlenerek hazır hale getirilmiştir.
- 6.2. PLC yazılımında birçok kez programa müdahale edilmiş, en sonunda istenilen işlemleri gerçekleştirme tamamlanmış ve test edilmiştir.
- 6.3. Form tasarımı yavaş yavaş şekillenmiştir. Öncelikle bağlantı butonları eklenmiş, PLC' den seri RS232 protokolüyle veriler alınmıştır.
- 6.4. Veri tabanı uygulamaları localhost ortamında SQL Server ile çalıştırılmıştır.
- 6.5. SCADA ortamı hazırlanmış, birçok denemeden sonra nihai hali verilmiştir.
- 6.6. Excel ile uyum da sağlanmıştır.
- 6.7. Arttırılmış gerçeklik uygulamasına geçilmiş, Artoolkit kütüphanesinden yararlanılmış, Google Sketcup ile 3D hazır bir robot kol seçilmiş, web cam ile gerekli ayarlamalar yapıldıktan sonra projeye eklenmiştir.
- 6.8. Projenin C# kodlama kısmında özellikle sensörlerden gelen anlık veri yapısının karmaşıklığı timer içinde string komutları kullanılarak giderilmiş ve istenilen değer okunup forma aktarılabilmıştır.

- 6.9. PLC-PC bağlantısı için Mitsubishi MX-Component Kütüphanesi kullanılmıştır. PLC-PC-Bant iletişimi sağlanarak veri tabanında veriler saklanabilmektedir.
- 6.10. Web ortamında veri takibi ve izlenmesi için ise ASP.NET ortamında C# diliyle tek sayfalık küçük bir web uygulaması yapılmıştır. Sayfada veritabanından çekilen veriler grafiksel olarak gösterilmiştir.
- 6.11. Artırılmış gerçeklik uygulamasında ise ışık şiddeti ve yönü, parlaklığı etkilediği için bazen problem çıkarmakta olsa da uygun ışık şiddetindeki ortamlarda problem çıkarmamaktadır.
- 6.12. Görüntü işleme yazılımında benzer yazılımlardan faydalanılmıştır.
- 6.13. Arduino ve Orange Pi yazılımlarında birçok problemlerle karşılaşıldı. IP uyumu, görüntü aktarımı, elektronik problemler gibi karşılaşılan sorunlar giderilerek projeye eklendi.
- 6.14. Projenin amacına uygun olarak tamamlandığı görüldü.
- 6.15. Benzer deney setlerinin de uygun maliyetle ve yerli imkânlarla üretilebileceği kanaatine varılmıştır. Ayrıca okuldaki öğrenciler, severek setin içine aldığı konuları set üzerinde öğrenmenin tadına varmış ve şu anda geleceğe umut ve özgüvenle bakmaktadırlar.

7. Öneriler

- 7.1. Öncelikle proje geliştirebilir, değiştirilebilir, yaygınlaştırılabilir esnek bir yapıya sahiptir. Mesela kullanılan plc başka bir firma plc ürünü ile değiştirilebilir ve bu sayede farklı plc programlama eğitimleri set üzerinden verilebilir. Ya da bant yerine seri iletişim sağlayan başka bir ölçüm cihazı bağlanabilir. Bu sayede programlama dili yeni bağlanan cihaza ve verilerine göre şekillenebilir.
- 7.2. Veri tabanında bulut ortamına geçiş yapılabilir.
- 7.3. Asp.net yerine phpve javagibi uygulamalara kapı açılır. Veri tabanı ve SQL sorgulama komutları yeni elde edilen sisteme göre şekillenebilir.
- 7.4. Esnek yapısından dolayı öğrencilere farklı farklı fakat benzer projeler, setin sınırları içerisinde verilip öğrencilere proje tabanlı öğrenme metoduyla takım çalışmaları ile yetiştirilebilir.
- 7.5. Öğrenciler artırılmış gerçeklik ve sanal gerçeklik uygulamalarını set üzerinde başka denemeler ile kendilerini geliştirebilir.
- 7.6. Tasarım ve mekaniğe meraklı öğrenciler ise farklı 3d modellemeler tasarlayıp, 3d baskı alıp sete özgünlük katabilir.
- 7.7. Projeye artırılmış gerçeklik ile çeşitli oyunlar eklenebilir ve set daha da eğlenceli hale gelebilir.
- 7.8. Sadece lise öğrencileri için değil, üniversite öğrencileri ve sanayi çalışanlarına set üzerinden eğitimler verilebilir. Fabrikaların eğitim ve ar-ge birimlerinde kolaylıkla kullanılabilir.
- 7.9. Okullarda bu gibi setler ile Endüstri 4.0 laboratuvarları kurulabilir. Bu Sanayi 4.0 vizyonuna ülkemizi taşıyacak nesiller yetiştirilebilir.
- 7.10. Bu projenin daha da verimli olması için eğitim materyalleri ve dokümanları hazırlanabilir. Android ve javadesteği, sanal gerçeklik (VR) uygulamaları eklenebilir. Görsel ve sesli dersler ile bir çok kitleye Endüstri 4.0 bileşenlerinin dersleri ve tanıtımı yapılabilir. Daha güzel ürünler için set geliştirilebilir.

Projenin video linki:

https://www.youtube.com/watch?v=QW_cGY0wO4

Kaynakça

- [1] Schildt, H. (2013), Herkes İçin C#, Alfa Yayınları, 786p
- [2] Çakal M.A.,Eymirli E.B., Artırılmış Gerçeklik Teknolojisi (AugmentedReality), KUDAKA, 2012
- [3] Mitsubishi FX2N Programming Manual,
http://dl.mitsubishielectric.com/dl/fa/document/manual/plc_fx/jy992d48301/jy992d48301j.pdf, Son erişim 20.9.2019
- [4] MX Component V.3 Programming Manual ,
<http://dl.mitsubishielectric.com/dl/fa/document/manual/plc/sh080272/sh080272o.pdf>, Son erişim, 11.10.2019
- [5] ARToolKit web page , <http://www.hitl.washington.edu/artoolkit/>, Son Erişim 20.09.2019
- [6] AForge.NET web page , <http://www.aforgenet.com/>, Son Erişim 17.12.2019
- [7] AForce.NET Framework Documentation, <http://www.aforgenet.com/framework/docs/>, Son erişim 04.10.2019
- [8] C# ile Renkli Nesne Takibi, <http://uguryalcin.org/wordpress/?p=372&lang=tr>, Son erişim 9.10.2019
- [9] AForge.NET, <http://mekatronik.wikidot.com/aforge>, Son erişim 11.10.2019
- [10] AForge.NET Open Source Framework,
<http://www.codeproject.com/Articles/16859/AForge-NET-open-source-framework>, Son erişim 08.11.2019
- [11] Çakır H., Babacan H.K., (2011), Hareketi Algılayan Kamera Destekli Güvenlik Programı, Gazi Üniversitesi Bilgisayar Eğitimi Bölümü Bilişim Teknolojileri Dergisi, cilt: 4, sayı: 2, (Makale)
- [12] Orange Pi,<http://www.orange-pi.org/>, Son erişim 15.10.2019
- [13] Arduino,<https://www.arduino.cc/>, Son erişim 18.10.2019

TEŞEKKÜR

Bu projede bizlere her konuda yardımcı olup rehberlik eden Danışman Öğretmenimiz İbrahim AYDIN'a,

Okul yöneticilerimize ve alan öğretmenlerimize katkılarından dolayı teşekkür ederiz.