

**KOCAELİ ÜNİVERSİTESİ\*MÜHENDİSLİK FAKÜLTESİ**

**Akıllı Çiftlik Uygulaması**

**Mühendislik Tasarımı-3 Proje Raporu**

**Metehan Karahalilöz  
180207028**

**Bölümü: Elektronik ve Haberleşme Mühendisliği**

**Danışman: Doç. Dr. Oğuzhan Karahan**

**KOCAELİ, 2023**

## İÇİNDEKİLER

1.GİRİŞ.....	3
1.1.Akıllı Çiftlik Teknolojileri.....	3
1.2.Akıllı Çiftliğin Yararları .....	4
2.Kullanılan Platform ve Malzemeler .....	5
2.1.Malzemeler .....	5
2.1.1.....ESP 32.....	5
2.1.2.....DHT 11.....	5
2.1.3.....MQ-2.....	6
2.1.4.....LDR.....	6
2.1.5.SERVO MOTOR.....	6
2.1.6.DC MOTOR .....	7
2.2.Kullanılan Yazılım .....	7
2.2.1.....C#.....	7
2.2.2.C# Form Uygulaması .....	10
3.IOT .....	11
3.1.Seri Haberleşme .....	12
4.Sistem ve Sonuç .....	13
4.1.Sonuç.....	13
Kaynakça .....	16

## 1. Giriş

Günümüz dünyasında teknolojiye ve teknolojiye ayak uydurmanın insan hayatındaki kolaylıklarını hepimiz biliyoruz. Projemizde köy ve kasaba gibi yerleşim yerlerinde yaşayan, geçimini ve geçimini tarım ve hayvancılıkla sağlayan çiftçilerimizi düşünerek, hayatlarındaki insan gücünü azaltmak için tasarlanmış bir projedir. Akıllı tarım, tarım endüstrisine operasyonları izlemek, izlemek, otomatikleştirmek ve analiz etmek için büyük veri, bulut ve nesnelerin interneti (IoT) dahil olmak üzere ileri teknolojiden yararlanmak için altyapı sağlamaya odaklanan bir yönetim konseptidir. Hassas tarım olarak da bilinen akıllı tarım, yazılımla yönetilir ve sensörlerle izlenir. Artan küresel nüfus, daha yüksek mahsul verimi için artan talep, doğal kaynakları verimli kullanma ihtiyacı, bilgi ve iletişim teknolojisinin artan kullanımı ve karmaşıklığı ve artan iklim-akıllı ihtiyaç kombinasyonu nedeniyle akıllı tarımın önemi artıyor.

### 1.1. Akıllı çiftlik teknolojileri

Akıllı çiftlik, aşağıdaki gibi teknolojilerin kullanımını içerir: Toprak taraması ve su, ışık, nem ve sıcaklık yönetimi için sensörler. Gelişmiş ağ ve GPS gibi telekomünikasyon teknolojileri. Özel uygulamalar ve IoT tabanlı çözümler, robotik ve otomasyon sağlamak için donanım ve yazılım. Karar verme ve tahmin için veri analitiği araçları. Mahsul verimi, toprak haritalama, iklim değişikliği, gübre uygulamaları, hava durumu verileri, makine ve hayvan sağlığından elde edilen verilerin miktarı artmaya devam ettiğinden, veri toplama akıllı çiftiliğin önemli bir parçasıdır. Tüm bir alan için günün her saati veri toplamak için uydular ve dronlar. Bu bilgiler, uzaktan izlemeyi mümkün kılan "göz tarlada" veya "göz ahırda" vermek üzere izleme ve analiz için BT sistemlerine iletilir.

Bu teknolojilerin birleşimi, makineden makineye (M2M) türetilmiş verileri kolaylaştırır. Bu veriler, bir karar destek sistemine beslenir, böylece çiftçiler neler olup bittiğini geçmişte olduğundan daha ayrıntılı bir şekilde görebilirler. Örneğin, bir tarladaki farklılıkları hassas bir şekilde ölçerek ve stratejiyi buna göre uyarlayarak, çiftçiler pestisitlerin ve gübrelerin etkinliğini büyük ölçüde artırabilir ve bunları daha akıllıca kullanabilir. Benzer şekilde, akıllı çiftçilik teknikleri, çiftçilerin bireysel hayvanların ihtiyaçlarını daha iyi izlemelerine ve hastalıkları önlemek ve sürü sağlığını iyileştirmek için beslenmelerini ayarlamalarına yardımcı olur.

## **1.2. Akıllı çiftliğin yararları**

Hassas tarım, çiftçiliği daha bağlantılı ve akıllı hale getirerek, genel maliyetleri düşürmeye ve ürünlerin kalitesini ve miktarını, tarımın sürdürülebilirliğini ve tüketici deneyimini geliştirmeye yardımcı olur. Üretim üzerinde artan kontrol, daha iyi maliyet yönetimi ve atık azaltımına yol açar. Örneğin, ekin büyümesi veya hayvan sağlığındaki anormallikleri izleme yeteneği, verim kaybı riskini ortadan kaldırmaya yardımcı olur. Ek olarak, otomasyon verimliliği artırır. Akıllı cihazlarla, aynı anda birden fazla işlem etkinleştirilebilir ve otomatikleştirilmiş hizmetler, üretim süreçlerini daha iyi kontrol ederek ürün kalitesini ve hacmini artırır. Akıllı tarım sistemleri aynı zamanda talep tahmininin dikkatli bir şekilde yönetilmesini ve israfı azaltmak için malların tam zamanında pazara sunulmasını sağlar. Hassas tarım, talep edilen doğru ürün için üretim sağlamak üzere arazi arzını yönetmeye ve arazinin durumuna bağlı olarak nem, gübre veya malzeme içeriği gibi doğru büyüme parametrelerine odaklanmaya odaklanır. Uygulanan hassas tarım sistemlerinin türleri, işletmenin yönetimi için yazılım kullanımına bağlıdır. Kontrol sistemleri, ortaya çıkan sorunlara yanıt vermek ve üretim desteği için makine ve ekipmanların otomasyonuna ek olarak, tedarik ve karar desteği için uzaktan bilgi sağlayarak sensör girdisini yönetir.

## **2. KULLANILAN PLATFORM**

### **2.1. Malzemeler**

Akıllı çiftlik kurabilmek için verileri okumamız gerekiyor, verilerimizi sensörler yardımıyla alıyoruz. Aynı zamanda bu verilere cevap verebilmek için oluşturacağımız sistemde belirli materyallere ihtiyacımız var.

#### **2.1.1. ESP 32**

ESP32, tamamen bağımsız bir sistem olarak veya bir ana bilgisayar MCU'suna bağlı bir cihaz olarak çalışabilir ve ana uygulama işlemcisindeki iletişim yığını yükünü azaltır. ESP32, SPI / SDIO veya I2C / UART arayüzleri aracılığıyla Wi-Fi ve Bluetooth işlevselliği sağlamak için diğer sistemlerle arayüz oluşturabilir.



Figure 1 ESP32

#### **2.1.2. DHT 11**

DHT11, temel, ultra düşük maliyetli bir dijital sıcaklık ve nem sensörüdür. Çevreleyen havayı ölçmek için kapasitif bir nem sensörü ve bir termistör kullanır ve veri pinine bir dijital sinyal verir (analog giriş pinlerine gerek yoktur). Kullanımı oldukça basittir, ancak verileri almak için dikkatli zamanlama gerektirir.



Şekil 2 DHT11

### 2.1.3. MQ-2

MQ-2, Winsen'den bir duman ve yanıcı gaz sensörüdür. 300 - 10000ppm aralığında yanıcı gazları algılayabilir. En yaygın kullanımı evsel gaz kaçağı alarmları ve propan ve dumana karşı yüksek hassasiyete sahip dedektörlerdir.



Şekil 3 MQ-2

### 2.1.4. LDR

Işığa bağımlı dirençler (LDR) olarak da bilinen fotodirençler, genellikle ışığın varlığını veya yokluğunu belirtmek veya ışık yoğunluğunu ölçmek için kullanılan ışığa duyarlı cihazlardır.



Şekil 4 LDR

### 2.1.5. SERVO MOTOR

Servo motorlar veya bilindikleri adıyla "servolar", bir makinenin parçalarını hassas bir şekilde döndüren ve iten elektronik cihazlar ve döner veya doğrusal aktüatörlerdir. Servolar esas olarak açısal veya doğrusal pozisyonda ve belirli hız ve ivme için kullanılır.



Şekil 5 SERVO MOTOR

### 2.1.6. DC MOTOR

Bir DC motor, doğru akım (DC) elektrik enerjisini mekanik enerjiye dönüştüren herhangi bir döner elektrik motoru sınıfından biridir.



Şekil 6 DC MOTOR

## 2.2. Yazılım

Yazılım, bilgisayarları çalıştırmak ve belirli görevleri yürütmek için kullanılan bir dizi talimat, veri veya programdır. Bir bilgisayarın fiziksel yönlerini tanımlayan donanımın tam tersidir. Yazılım, bir cihazda çalışan uygulamalara, komut dosyalarına ve programlara atıfta bulunmak için kullanılan genel bir terimdir.

### 2.2.1. C#

C# ("Keskin Gör" olarak telaffuz edilir), modern, nesne yönelimli ve tür açısından güvenli bir programlama dilidir. C#, geliştiricilerin . AĞ. C#'ın kökleri C dil ailesindedir ve C, C++, Java ve JavaScript programcıları tarafından hemen tanındık gelecektir.

.NET Framework'ün geliştirilmesi sırasında, sınıf kitaplıkları orijinal olarak "Simple Managed C" (SMC) adlı bir yönetilen kod derleyici sistemi kullanılarak yazılmıştır.[20][21] Ocak 1999'da Anders Hejlsberg, o zamanlar "C benzeri Nesne Yönelimli Dil" anlamına gelen Cool adlı yeni bir dil oluşturmak için bir ekip kurdu.[22] Microsoft, "Cool" adını dilin son adı olarak tutmayı düşünmüştü, ancak ticari marka nedenleriyle bunu yapmamayı seçti. .NET projesi Temmuz 2000 Profesyonel Geliştiriciler Konferansı'nda halka duyurulduğu zaman, dil C# olarak yeniden adlandırılmıştı ve sınıf kitaplıkları ve ASP.NET çalışma zamanı C#'a taşınmıştı.

- **Taşınabilirlik**

Tasarım gereği C#, temeldeki Ortak Dil Altyapısını (CLI) en doğrudan yansıtan programlama dilidir.[62] İçsel türlerinin çoğu, CLI çerçevesi tarafından uygulanan değer türlerine karşılık gelir. Bununla birlikte, dil belirtimi, derleyicinin kod oluşturma gereksinimlerini belirtmez: yani, bir C# derleyicisinin bir Ortak Dil Çalışma Zamanını hedeflemesi veya Ortak Ara Dil (CIL) oluşturmaya veya başka herhangi bir özel biçim oluşturmaya gerektiğini belirtmez. Teorik olarak, bir C# derleyicisi, geleneksel C++ veya Fortran derleyicileri gibi makine kodu üretebilir.

- **Yazım**

C# dili, genel değişkenlere veya işlemlere izin vermez. Tüm yöntemler ve üyeler sınıflar içinde bildirilmelidir. Genel sınıfların statik üyeleri, genel değişkenlerin ve işlevlerin yerini alabilir.

Yerel değişkenler, C ve C++'dan farklı olarak çevreleyen bloğun değişkenlerini gölgeleyemez.

- **Metod ve fonksiyonlar**

C# dilinde bir yöntem, yalnızca bir sınıf özelliğinin değer tutma yeteneğinden ziyade bir işlev (bir yönerge dizisi) olarak çağrılabilen bir sınıfın üyesidir. C++ ve ANSI C gibi sözdizimsel olarak benzer diğer dillerde olduğu gibi, bir yöntemin imzası sırasıyla şunları içeren bir bildirimdir: herhangi bir isteğe bağlı erişilebilirlik anahtar sözcüğü (özel gibi), dönüş türünün açık belirtimi (int gibi) veya anahtar kelime geçersiz), yöntemin adı ve son olarak, her biri bir parametrenin türünden, resmi adından ve isteğe bağlı olarak hiçbir



olmadığında kullanılacak varsayılan bir değerden oluşan, virgülle ayrılmış parametre belirtimlerinin parantez içindeki bir dizisi. tedarik edilen. Dönen değer veya atama yoluyla bir sınıf özelliğini basitçe alan veya ayarlayanlar gibi belirli belirli türde yöntemler, tam imza gerektirmez, ancak genel durumda, bir sınıfın tanımı, yöntemlerinin tam imza bildirimini içerir.

- **Özellik**

C#, özelliklere sahip sınıfları destekler. Özellikler, bir destek alanına sahip basit erişimci işlevler olabilir veya alıcı ve ayarlayıcı işlevleri uygulayabilir.

C# 3.0'dan bu yana, erişimcinin (alıcı) ve değiştiricinin (ayarlayıcının) işlemleri bir sınıfın tek bir özniteliği üzerinde kapsüllediği otomatik uygulanan özelliklerin sözdizimsel şekeri mevcuttur.

- **İsim alanı**

Bir C# ad alanı, bir pakete çok benzer kurallar ve özelliklerle bir Java paketi veya bir C++ ad alanıyla aynı düzeyde kod yalıtımı sağlar. Ad alanları, "kullanma" söz dizimi ile içe aktarılabilir.

- **Bellek**

C#'ta, bellek adres işaretçileri yalnızca özel olarak güvensiz olarak işaretlenmiş bloklar içinde kullanılabilir[69] ve güvenli olmayan kod içeren programların çalışması için uygun izinler gerekir. Çoğu nesne erişimi, her zaman ya "canlı" bir nesneye işaret eden ya da iyi tanımlanmış boş değere sahip olan güvenli nesne referansları aracılığıyla yapılır; "ölü" bir nesneye (çöp toplanmış bir nesne) veya rastgele bir bellek bloğuna başvuru elde etmek imkansızdır. Güvenli olmayan bir işaretçi, çöpten toplanan nesnelere, diziye, dizeye veya yığınla ayrılmış bellek bloğuna herhangi bir başvuru içermeyen 'yönetilmeyen' bir değer türünün örneğine işaret edebilir. Güvensiz olarak işaretlenmemiş kod, System.IntPtr türü aracılığıyla işaretçileri depolayabilir ve işleyebilir, ancak bunların başvurusunu kaldıramaz.

- **İstisnalar**

Programcılar için bir dizi standart istisna mevcuttur. Standart kitaplıklardaki yöntemler, bazı durumlarda düzenli olarak sistem istisnaları atar ve atılan istisnaların aralığı normalde belgelenir. Sınıflar için özel istisna sınıfları tanımlanabilir ve gerektiğinde belirli durumlar için işleme konulmasına izin verilir.

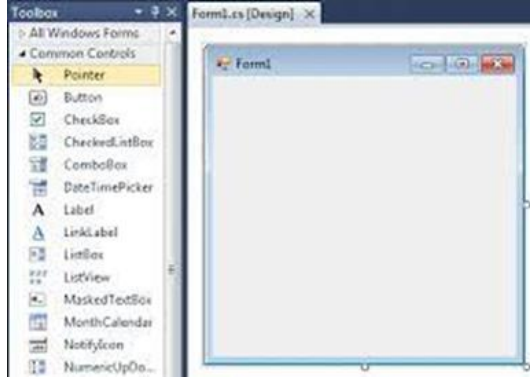
Kontrol edilen istisnalar C#'ta mevcut değildir (Java'nın aksine). Bu, ölçeklenebilirlik ve sürümlenebilirlik konularına dayanan bilinçli bir karar olmuştur.

- **Mimari**

C++'dan farklı olarak, bir sınıf herhangi bir sayıda "arayüz" (tamamen soyut sınıflar) uygulayabilse de, C# çoklu kalıtımı desteklemez. Bu, komplikasyonlardan kaçınmak ve CLI boyunca mimari gereksinimleri basitleştirmek için dilin baş mimarı tarafından verilen bir tasarım kararıydı.

### 2.2.2. C# FORM UYGULAMASI

C# uygulamasındaki bir Windows formu, bir bilgisayarın masaüstünde çalışan formdur. C# ile birlikte Visual Studio Form, bir Windows Forms uygulaması oluşturmak için kullanılabilir. Denetimler, Visual Studio'daki Araç Kutusu aracılığıyla Windows C# formlarına eklenebilir. Etiketler, onay kutuları, radyo düğmeleri vb. gibi kontroller.



Şekil 7 C# FORM ÖRNEĞİ

### 3. IoT(Nesnelerin interneti)

Nesnelerin İnterneti veya İngilizce Nesnelerin İnterneti'nin kısaltması olarak kullanılan IoT, fiziksel nesnelerin birbiriyle veya daha büyük sistemlerle bağlantılı olduğu bir iletişim ağıdır. İnternet üzerinden diğer cihazlara ve sistemlere bağlanmak ve bunlarla veri alışverişi yapmak için sensörler, yazılım ve diğer teknolojilerle gömülüdür. Nesnelerin tek bir anahtar ile işaretlenebileceği ve internet altyapısı üzerinden birlikte çalışabilecekleri, böylece küçük parçaların toplamından daha büyük değerler oluşturabilecekleri öngörülmüştür. Askeri sistem için "askeri nesnelerin interneti" oluşturuldu.

Geleneksel gömülü sistemler, kablosuz sensör ağları, kontrol sistemleri, otomasyon (ev otomasyonu ve bina otomasyonu dahil) ve diğer alanlar, Nesnelerin İnterneti'nin etkinleştirilmesine katkıda bulunur.

Nesnelerin interneti açısından nesne kavramı çok geniş bir anlama sahiptir. Herhangi bir izleme cihazı, sensör, biyoçip veya erişim cihazı nesne olarak nitelendirilir. Bir cihazın "akıllı" olarak kabul edilmesi ve bir nesne olarak nitelendirilmesi için gerekli koşullar: Benzersiz bir adı (benzersiz id), bağlanabilir olması ve bir sensöre sahip olması gerekir. Bu sayede akıllı nesne dünyanın her yerinden erişilebilir ve kontrol edilebilir hale geliyor. Günümüzde yaygın olarak kullanılan IP adreslerinin sınırlı sayıda olması nedeniyle nesnelerin benzersiz bir isme sahip olması mümkün değildir ve bu durumu aşmak için tasarlanan IPv6 kullanımı ile nesneler gerçekten benzersiz hale gelebilmektedir.

IoT uygulamaları, sensörleri tek tek erişilebilir hale getirmenin yanı sıra birçok sensörün verilerini birleştirerek değer üretmek amacıyla da kullanılmaktadır. Fiziksel ortamlardan akan yüksek miktardaki sensör verilerinin (data) değerlendirmeler sonrasında operatörlere veya ilgili kişilere bilgi olarak iletilmesi veya verilerin sistemler yardımıyla işlenerek bir faaliyetin yürütülmesi sağlanır.

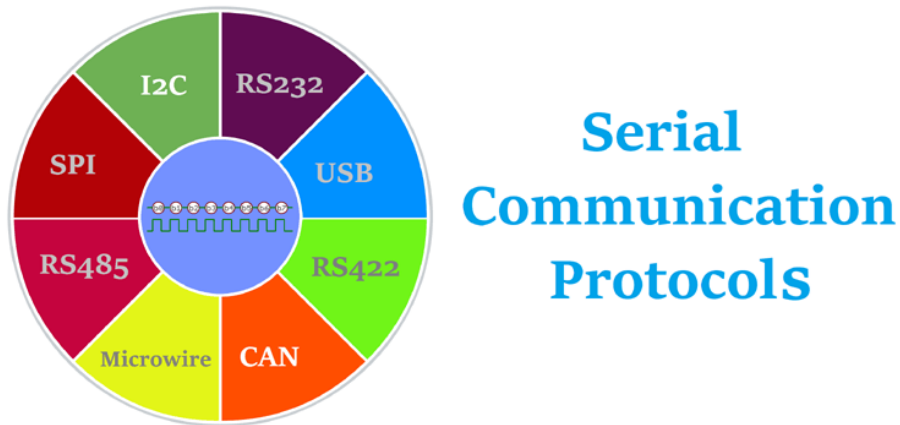
Küresel ağa bağlanan ilk cihazlar 1982'de ortaya çıktı. Makinenin sıcaklığını kontrol edebilen ve içindeki şişe sayısını takip edebilen bir Coca-Cola otomatıydı. “Nesnelerin İnterneti” teriminin, RFID teknolojisi araştırmacısı Kevin Ashton tarafından 1999 yılında formüle edildiği düşünülmektedir.1990'larda IoT ile ilgili tüm faaliyetler teorik

kavramlara, tartışmalara ve bireysel fikirlere indirgendi. 2000'ler ve 2010'lar, IoT projelerinin başarılı olmaya başladığı ve bazı pratik uygulamalar bulduğu hızlı bir gelişim dönemiydi. Akıllı lambalar ve spor takip cihazlarından sürücüsüz arabalara ve akıllı şehirlere kadar birçok küçük ve büyük proje oluşturuldu. Bu, uzun mesafelerde bilgi iletebilen kablosuz bağlantıların ortaya çıkması ve İnternet iletişiminin artan bant genişliği nedeniyle mümkün oldu.

IoT tamamen “farklı bir internet”e dönüşmüştür ki mevcut tüm protokoller ihtiyaçlarını karşılayamaz ve kesintisiz bağlantı sağlayamaz. Bu nedenle, özelleştirilmiş IoT iletişim protokolleri ve standartları oluşturmak hayati bir gereklilik haline geldi. Ancak bazı mevcut teknolojiler (ör. HTTP) Nesnelerin İnterneti tarafından da kullanılmaktadır.

### 3.1. Seri Haberleşme

Seri iletişimde, veriler ikili darbeler şeklinde gönderilir. Diğer bir deyişle, ikili 1'in mantık yüksek veya 5 volt, 0'ın mantık düşük veya 0 volt olduğunu söyleyebiliriz. Seri iletişim, iletim moduna ve veri iletim tipine bağlı olarak çeşitli şekillerde olabilir. Bu iletim modu tek yönlü, yarı çift yönlü ve tam çift yönlü olarak sınıflandırılır. Her iletim modu için bir verici ve alıcı vardır. Her iletim modu için bir verici ve alıcı vardır.



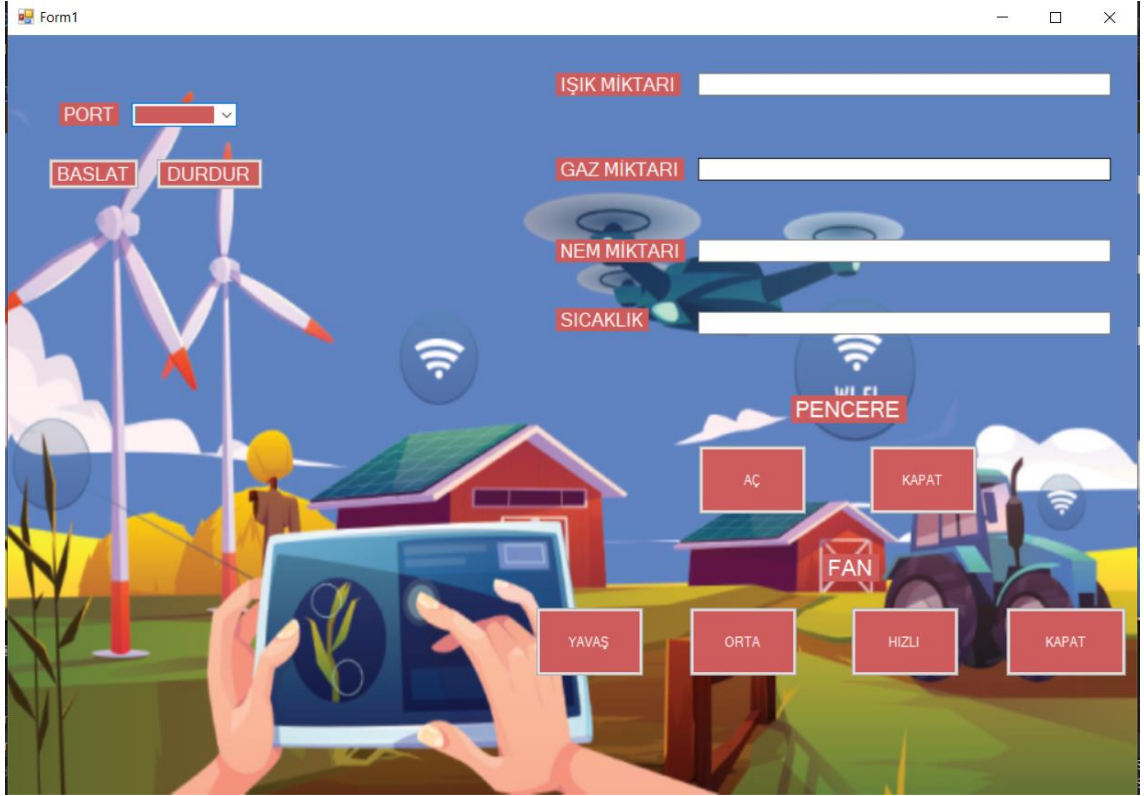
Şekil 8 Seri Haberleşme

#### 4. Sistemin Kurulması

Kurulan elektronik sistemle birlikte UI'den verilerimizi okuyarak aynı anda sisteme veri göndereceğiz.

##### 4.1. Sonuç

Arayüzümüzü Şekil 9'da görebilirsiniz.



Şekil 9 UI

Şekil 10'da sistemimiz ile form arasında iletişim kurmak ve formumuzu düzgün bir şekilde çalıştırabilmek için gerekli kütüphaneler eklenmiştir.

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.IO.Ports;
7  using System.Linq;
8  using System.Reflection.Emit;
9  using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;

```

Şekil 10 Library

Sistem ve bilgisayar iletişimi için gerekli kodlar Şekil 11'da yazılmıştır.

```

private string data;

public Form1()
{
    InitializeComponent();

private void Form1_Load(object sender, EventArgs e)
{
    string[] ports = SerialPort.GetPortNames(); //Port isimlerini al
    foreach (string port in ports)
        comboBox1.Items.Add(port); //Port isimlerini comboBox içine yaz
    serialPort1.DataReceived += new SerialDataReceivedEventHandler(SerialPort1_DataReceived);
}

private void SerialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    data = serialPort1.ReadLine(); //Veriyi al
    this.Invoke(new EventHandler(displayData_event));
}

```

Şekil 11 Seri Haberleşme

Şekil 12'de sistemdeki verilerin okunduğu kodları ve bu değerlerin olması gereken değerlerle karşılaştırmasını görebilirsiniz.

```

private void displayData_event(object sender, EventArgs e)
{
    string[] value = data.Split('*'); // '*' gördüğün yerlerden stringi ayır ve diziye ata
    textBox1.Text = DateTime.Now.ToString() + " " + "IŞIK DEĞERİ " + value[0];
    textBox2.Text = DateTime.Now.ToString() + " " + "GAZ DEĞERİ " + value[1];
    textBox3.Text = DateTime.Now.ToString() + " " + "NEM DEĞERİ % " + value[2];
    textBox4.Text = DateTime.Now.ToString() + " " + "SICAKLIK DEĞERİ " + value[3] + "°C";
    int a = 4095;
    int ısı = Int32.Parse(value[0]);
    int b = 650;
    int gaz = int.Parse(value[1]);
    int c = 50;
    int nem = int.Parse(value[2]);
    int d = 20;
    int sıcaklık = int.Parse(value[3]);
    if (ısı >= a)
    {label10.Text = "IŞIĞI AÇ";}
    else
    {label10.Text = string.Empty;}
    if (gaz > b)
    {label11.Text = "CAMI AC";}
    else
    {label11.Text = string.Empty;}
    if (nem > c)
    {label12.Text = "FANI ÇALIŞTIR";}
    else
    {label12.Text = string.Empty;}
    if (sıcaklık > d)
    {label13.Text = "FANI ÇALIŞTIR";}
    else
    {label13.Text = string.Empty;}
}
}

```

Şekil 12 Veri alma

## **Kaynakça**

- [1] Güneş, H., Bıçakcı, S., (2018). Akıllı Evler İçin Sesli Komut Algılama Yöntemleri. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 20(2), 561-568.
- [2] Jamalabad, M. S., (2014). Enerjisini İklımden Üreten Akıllı Evler. Yüksek Lisans Tezi, Danışman Işık, B., Aydın Üniversitesi, İstanbul, 8-21.
- [3] Yumurtacı, M., Keçebaş, A., (2009). Akıllı Ev Teknolojileri ve Otomasyon Sistemleri Smart Building Technologies And Its Automation Systems, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), Karabük, 1-6.
- [4] Melek T. ,(2019). Konutta Yenilikçi Tasarım: Akıllı Evler Üzerine Bir Değerlendirme. Yüksek Lisans Tezi, Danışman Prof. Dr. Neslihan D., İstanbul Kültür Üniversitesi, İstanbul, 26-192.