

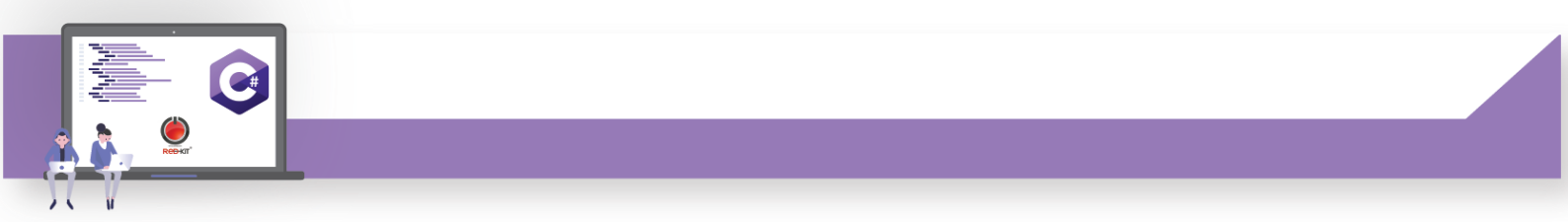
## ARDUİNO İLE C#

### ARDUİNO İLE C#

- ✓ *C# Haberleşmesi ile Led Yakma*

### KAZANIMLAR

- ✓ *Seri İletişim kavramını kavrar*
- ✓ *Seri Haberleşme ile Led Yakmayı yapar.*
- ✓ *C# Form Uygulaması Oluşturur.*



## Arduino ve C# Haberleşmesi ile Led Yakma

**Seri haberleşme** yardımıyla ve **C#** kullanarak oluşturduğumuz arayüzlerle projelerimizi kontrol etmeyi ve veri okumayı öğreneceğiz.

### Seri Haberleşme Nedir?

Dijital verinin tek bir hat üzerinden taşınmasına seri haberleşme denmektedir. Seri haberleşmede bir bit yalnızca bir kerede iletilmektedir. Paralel iletişimde ise tüm bitlerin aynı anda iletilmesi söz konusudur. Bu sebepten dolayı, seri haberleşmede yalnızca tek bir kablo kullanılmaktadır. Jumper kablo maliyeti konusunda kesinlikle seri haberleşme daha avantajlıdır.

Arduino üzerinden seri haberleşme konusuna ayrı bir paragrafta değinmeliyiz. Üzerinde bulunan seri iletişim birimi aracıyla seri haberleşme yapılmaktadır. Projelerin kodlarını yazarken Arduino'da seri haberleşme en az bir kere kullanılmaktadır. Arduino ile bilgisayarlar arasındaki seri haberleşmeyi sağlamak adına `serial.begin(9600)` komutu kullanılmaktadır. `Serial.print ()`komutuyla da bilgisayara veri gönderimi yapılmaktadır.

### Arduino Seri Haberleşmesi ile Led Yakma

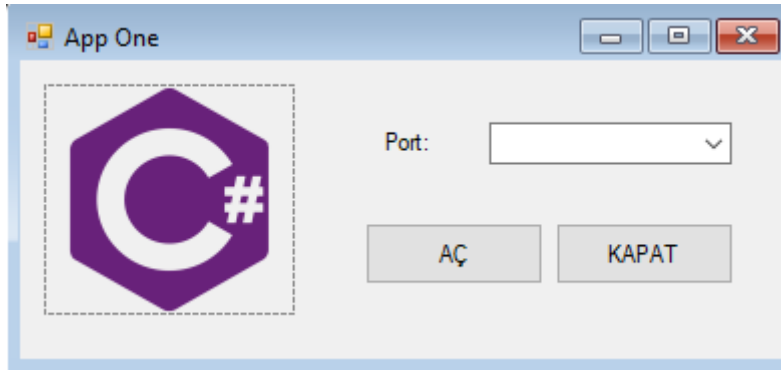
Seri haberleşme ile led yakmak için öncelikli olarak devre hazırlanmalıdır. Gereken kod yüklenmelidir. Ardından, seri port ekranı açılmalıdır. Bu sayede, ledin kontrol edilmesi söz konusu olmaktadır. Bu noktada, projenin bir gereği olarak C# ile oluşturulan arayüz sayesinde ledin kontrolü daha kolay hale gelecektir. Ancak, bunun için öncelikli olarak Visual Studio kurulumu gerekmektedir. Visual Studio bilgisi şarttır. Çünkü, bu sayede, C# form uygulaması oluşturulacaktır.

C# sayesinde led yakmak için öncelikli olarak proje oluşturmak gereklidir. Sürükle bırak tekniği kullanılmaktadır. İşte bu noktada, araç kutusundan eklenmesi gereken nesneler bulunmaktadır. Bu nesneler arasında ComboBox, iki button, label, serialport vardır. Bu nesneler; seri iletişimi sağlamak ve Arduino bağlantısını tamamlamak için gereklidir.

Sonuç olarak, her bir nesne için ayrı ayrı kod yazılması gerekmektedir. Bu sayede, ledin özgün tasarım arayüz üzerinden kontrolü mümkün olmaktadır.



## C# Form Ekranı



## C# Kod Blokları

```
using System;
using System.Windows.Forms;
using System.IO.Ports;

namespace AppOne
{
    public partial class formAppOne : Form
    {
        public formAppOne()
        {
            InitializeComponent();
        }
        SerialPort serialPort = new SerialPort(); // Instance
        private void formAppOne_Load(object sender, EventArgs e)
        {
            string[] ports = SerialPort.GetPortNames();
            foreach (string port in ports)
            {
                cbPorts.Items.Add(port);
            }
        }
        private void cbPorts_SelectedIndexChanged(object sender, EventArgs e)
        {
            serialPort.Close();
            serialPort.PortName = cbPorts.SelectedItem.ToString();
            serialPort.Open();
        }
        private void btnOpen_Click(object sender, EventArgs e)
        {
            serialPort.Write("1");
        }
        private void btnClose_Click(object sender, EventArgs e)
        {
            serialPort.Write("0");
        }
    }
}
```



## Arduino Kod Blokları

```
#define ledPin 7
```

```
byte val;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(ledPin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    if (Serial.available()) {
```

```
        val = Serial.read();
```

```
        Serial.println(val);
```

```
        if (val == '1') {
```

```
            digitalWrite(ledPin, HIGH);
```

```
        }
```

```
        if (val == '2') {
```

```
            digitalWrite(ledPin, LOW);
```

```
        }
```

```
    }
```

```
}
```