

# C# ile NJ Simulatöre Bağlanmak

- Genel Bilgi
- Sysmac Studio'da Değişkenlerin Oluşturulması
- Windows Form Oluşturulması
- Gerekli Dosyaların Referanslara Eklenmesi
- Form1.cs Dosya Yapısı

## Genel Bilgi

---

Bu dökümanda C# ile yazılmış bir projenin NJ Simulatör ile nasıl haberleşeceği anlatılmıştır.

Bu örnekte iki değişkenin değeri Windows Form ile okunup, değiştirilecektir.

Bu sayede yazılan C# kodları gerçek bir donanıma ihtiyaç duyulmadan denenebilecektir.

Not: Aynı kodda yapılacak bir kaç küçük değişiklik ile gerçek donanıma da bağlantı yapılabilir.

Bu dökümanda anlatılan işlemleri yapabilmek için aşağıda yer alan linkteki CNC Operator dosyası indirilmelidir.

Not: Simulatör haberleşmesi için gereken dll dosyaları bu dosyalar içerisinde olduğu için indirmek gerekiyor. Yapılacak olan işlemlerin CNC ile herhangi bir ilgisi yok.

**Önemli Not:** İlgili dosyaları indirmek için Sysmac Lisansı gerekmektedir.

<http://www.ia.omron.com/product/tool/cnc-operator/>

## Sysmac Studio'da Değişkenlerin Oluşturulması

C# ile NJ simülâtör haberleştirilirken Sysmac Studio'da herhangi bir ayar yapılmasına gerek yoktur.

Sadece Şekil 1'de gösterilen ilgili değişkenleri oluşturmak yeterli olacaktır.

Bu örnekte test ve test2 olarak 2 adet integer değişken oluşturulmuştur.

| Name  | Data Type | Initial Value | AT | Retain                   | Constant                 | Network Publish | Comment |
|-------|-----------|---------------|----|--------------------------|--------------------------|-----------------|---------|
| test  | INT       |               |    | <input type="checkbox"/> | <input type="checkbox"/> | Publish Only    |         |
| test2 | INT       |               |    | <input type="checkbox"/> | <input type="checkbox"/> | Publish Only    |         |

Şekil 1 Global Variables

Değişken oluştururken değişkenlere verilen isimlere ve data tiplerine dikkat edilmesi gerekir.

C# projesi yazarken değişken isimleri ve data tiplerine göre kod yazılır.

## Windows Form Oluşturulması

C# projesi için öncelikle Windows Form kullanılarak aşağıdaki gibi bir form oluşturulur.

Şekil 2 Windows Form Yapısı

Simulatöre Bağlan: Simulatör bağlantısını kurmak için kullanılır.

Bağlantıyı Kes: Simulatör bağlantısını kesmek için kullanılır.

Bağlantı Durumu: Simulatöre bağlantı gerçekleşirse "**Simulatöre bağlantı kuruldu.**" Bağlantı kesilirse de "**Bağlantı kesildi.**" Bilgisini göstermek için kullanılır.

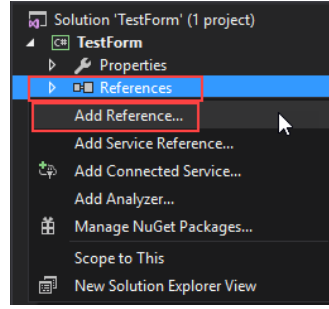
Oku: test ve test2 değişkenlerinin değerini simulatörden okuyup ilgili kısımlarda göstermek için kullanılır.

Yaz: test ve test2 değişkenlerinin değerini simulatöre yazmak için kullanılır.

## Gerekli Dosyaları Referanslara Eklenmesi

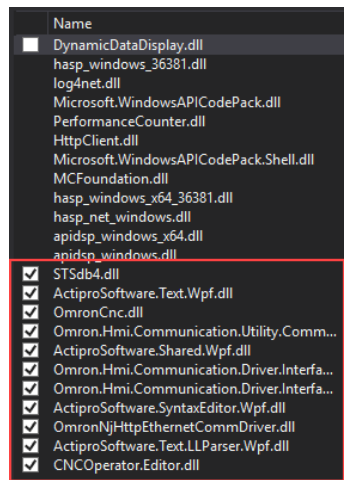
NJ Simulatör ile haberleşmek için gereken dll dosyalarının indirildikten sonra Solution References'a eklenmesi gerekir.

Referansları eklemek için References > Add Reference... yolu izlenir.



Şekil 3 Referans Ekleme

Referans ekleme sayfasında aşağıdaki referansların eklenmiş olması gerekir.



Şekil 4 Eklenmesi Gereken Referanslar

## Form1.cs Dosyasının Yapısı

Form1.cs dosyasının yapısı Şekil 5’de görüldüğü gibidir.

```
namespace TestForm
{
    3 references
    public partial class Form1 : Form
    {
        private readonly IController controller;
        private Hashtable myVar;

        1 reference
        public Form1()
        {
            InitializeComponent();
            var log = new NullLog();
            this.controller = ControllerFactory.CreateController(true, log, new Omron.Cnc.Range(17000,17005));
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)...

        1 reference
        private void button2_Click(object sender, EventArgs e)...

        1 reference
        private void button3_Click(object sender, EventArgs e)...

        1 reference
        private void button4_Click(object sender, EventArgs e)...
    }
}
```

Şekil 5 Form1.cs Yapısı

IController interface yapısı ile controller değişkeni oluşturulur.

Daha sonra bu controller değişkeninin karşılığı CreateController metodu ile oluşturulur.

ControllerFactroy.CreateController() metodu içerisindeki ilk parametre true yapılırsa bu programla simulatöre bağlantı yapılabilir. Eğer bu parametre false yapılırsa gerçek bir cihaza bağlantı yapılabilir.

Not: Sysmac Studio Simulatör çalıştırıldığında 17000. Portu kullanır. Birden fazla Sysmac simulatör açık olması durumunda kullanılan port farklı olabilir o yüzden port numarasına bir aralık vermek gerekir.

Button1, Simulatöre Bağlan butonudur. Button1'e basıldığında Button1\_Click event'i çağrılır.

```
private void button1_Click(object sender, EventArgs e)
{
    this.controller.Connect("127.0.0.1");
    if (this.controller.IsConnected)
    {
        textBox2.Text = "Simulatöre bağlantı kuruldu.";
        return;
    }

    if (!this.controller.IsConnected)
    {
        DialogResult result = MessageBox.Show("Bağlantı kurulamadı.");
        return;
    }
}
```

Şekil 6 Button1\_Click Event'i

Burada simülator lokal bilgisayarda çalıştığından dolayı 127.0.0.1 IP adresi verilir. Burada eğer gerçek PLC'ye bağlanılacaksa buraya PLC IP adresi verilmelidir. Bağlantı kurulursa Bağlantı Durumu kısmına "Simulatöre bağlantı kuruldu" bilgisi yazdırılır.

Bağlantı kurulamazsa ekrana bir mesaj kutusu gelir ve "Bağlantı kurulamadı" uyarısı yazdırılır.

Button2, Yaz butonudur. Button2'ye basıldığında Button2\_Click event'i çağrılır.

```
private void button2_Click(object sender, EventArgs e)
{
    int test_int = Int32.Parse(textBox5.Text);
    this.controller.Write("test", test_int);
    int test2_int = Int32.Parse(textBox4.Text);
    this.controller.Write("test2", test2_int);
}
```

Şekil 7 Button2\_Click Event'i

Bu butona basıldığında textBox'larda yer alan değerler alınır ve integer'a dönüştürülür. Daha sonra controller.Write() metodu ile istenen değişkene ilgili değer yazdırılır.

Burada textbox'lardan alınan değerlerin Sysmac Studio'da oluşturulmuş değişkenin data tipine dönüştürülmesi gerekmektedir.

Sysmac Studio'da oluşturulan test değişkeni Integer olduğundan dolayı textbox'dan alınan değer de Integer'a dönüştürülerek yazdırılır. Aksi takdirde bu butona basıldığında hata alınacaktır.

Button3, Oku butonudur. Button3'e basıldığında Button3\_Click eventi çağrılır.

```
private void button3_Click(object sender, EventArgs e)
{
    string[] readvar = new string[] { "test", "test2" };
    this.controller.Read(readvar, out myVar);
    textBox5.Text = myVar["test"].ToString();
    textBox4.Text = myVar["test2"].ToString();
}
```

Şekil 8 Button3\_Click Eventi

Okuma işlemi yapmak için öncelikle okuma yapılacak olan değişkenlerden oluşan bir string dizisi oluşturulur.

Daha sonra controller.Read() metoduna okuma yapılacak olan string dizisi birinci parametre olarak verilir. İkinci parametre olarak ise daha önceden oluşturulmuş olan Hashtable data tipine sahip değişken verilir. Bu örnek için oluşturulan değişken: myVar.

Gelen değerler textBox'larda string olarak gösterileceğinden dolayı okunan Integer değerler stringe dönüştürülerek ilgili textBox'lara yazdırılır.

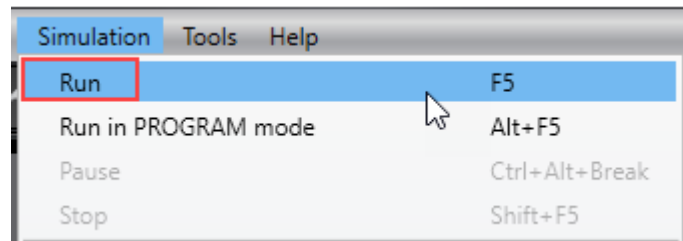
Button4 Bağlantıyı Kes butonudur. Button4'e basıldığında Button4\_Click eventi çağrılır.

```
private void button4_Click(object sender, EventArgs e)
{
    this.controller.Disconnect();
    textBox2.Text = "Bağlantı kesildi.";
}
```

Şekil 9 Button4\_Click Eventi

Bağlantıyı Kes butonuna basıldığında simulatör ile kurulmuş olan bağlantı kesilir. Bağlantı Durumu 'na da "Bağlantı kesildi" bilgisi yazdırılır.

C# projesi ile simulatöre bağlanmak için NJ simulatörün çalışıyor olması gerekmektedir. Simulatörü çalıştırmak için **Simulation > Run** yolu izlenir.



Şekil 10 Simulatör Başlatma