

NikCom HMI ile WebView2 Kullanımı

HTML / JavaScript ekranlardan NikCom HMI TagList erişimi

1. Amaç

Bu kılavuzda, Visual Studio WinForms projesi içinde Microsoft WebView2 componenti kullanılarak HTML, CSS ve JavaScript tabanlı ekranlardan NikCom HMI taglerine nasıl erişileceği anlatılmaktadır.

Amaç, web teknolojilerini bilen kullanıcıların C# tarafında detaylı kod yazmadan, HTML / JavaScript üzerinden PLC verisi okuyup yazabilmesini sağlamaktır.

2. WebView2 Nedir?

WebView2, Microsoft'un Chromium tabanlı modern browser componentidir. WinForms projeleri içine HTML, CSS, JavaScript, React gibi web teknolojileriyle hazırlanmış ekranları gömmek için kullanılabilir.

Bu yapı ile klasik WinForms HMI ekranları ve modern web dashboard ekranları aynı proje içinde birlikte çalışabilir.

3. Kurulum

NuGet üzerinden aşağıdaki paket yüklenmelidir:

Microsoft.Web.WebView2

Toolbox'a eklemek için:

Toolbox -> Sağ Tık -> Choose Items -> Browse

WinForms için seçilecek DLL:

packages\Microsoft.Web.WebView2.xxx\lib\net462\Microsoft.Web.WebView2.WinForms.dll

Not: lib_manual\netcoreapp3.0 klasörü WinForms .NET Framework projesi için kullanılmamalıdır.

4. Framework ve Referanslar

Proje için önerilen Framework sürümü .NET Framework 4.7 veya üzeridir.

Referanslarda aşağıdaki kütüphaneler bulunmalıdır:

- Microsoft.Web.WebView2.Core

- Microsoft.Web.WebView2.WinForms

5. Önemli Kurulum Notları

WebView2 NuGet paketi yüklendikten sonra mutlaka:

Clean Solution

Rebuild Solution

işlemleri yapılmalıdır.

Bazı durumlarda WebView2 componenti ve referanslar ancak Rebuild işleminden sonra düzgün algılanabilir.

6. Gerekli Using Satırları

Kod tarafında aşağıdaki using satırları eklenmelidir:

```
using Microsoft.Web.WebView2.Core;
using Microsoft.Web.WebView2.WinForms;
using System.Runtime.InteropServices;
```

7. Async Kullanımı

WebView2 başlatma işlemi asenkron çalışır. Bu nedenle WebView2'nin yükleneceği form olayında async kullanılmalıdır.

Örnek olarak Form Load olayı aşağıdaki gibi tanımlanmalıdır:

```
private async void Form1_Form_Load(object sender, EventArgs e)
{
    await webView21.EnsureCoreWebView2Async();
}
```

Bu satır çalışmadan WebView2 içindeki CoreWebView2 nesnesi hazır olmayabilir. Bu yüzden AddHostObjectToScript gibi işlemlerden önce EnsureCoreWebView2Async çağrılmalıdır.

8. HMI_Connect Nesnesine Erişim

Örnekte PLC bağlantısı, form üzerindeki bir NikCom HMI kontrolünün Connect bilgisinden alınmaktadır:

```
HMI_Connect hmi_Connect_PLC = hmi_Label1.Connect;
```

Burada hmi_Label1 üzerinde tasarım zamanında bir Connect seçimi yapıldığında, bu Connect bilgisi kod tarafında hmi_Connect_PLC değişkenine aktarılır.

Bu yöntem sayesinde MasterPage üzerinde tanımlanan HMI_Connect nesnesine başka formlardan rahatlıkla erişilebilir. Kullanıcı, ilgili form üzerindeki herhangi bir NikCom HMI kontrolü üzerinden seçilmiş Connect bilgisini alabilir.

9. JavaScript Tarafından Kullanım Mantığı

HTML tarafında hedeflenen kullanım şudur:

```
<button onclick="hmi_Connect_PLC.TagList_SetValue(25,true)">
    Motor Start
</button>
```

Buradaki 25 değeri, NikCom HMI Tag List içerisindeki Tag ID bilgisini temsil etmektedir.

Veri okuma için:

```
let veri = await hmi_Connect_PLC.TagList_GetValue(25);
```

Buradaki 25 değeri yine NikCom HMI içerisindeki ilgili tagin Tag ID değeridir.

10. Karşılaşılan Problem ve Çözüm

İlk testte şu hata alınabilir:

ReferenceError: hmi_Connect_PLC is not defined

Sebebi: WebView2 host object doğrudan global değişken olarak gelmeyebilir.

Çözüm olarak HTML / JavaScript tarafında aşağıdaki alias tanımlanmalıdır:

```
const hmi_Connect_PLC =
    chrome.webview.hostObjects.hmi_Connect_PLC;
```

11. Full Çalışan Örnek

Aşağıdaki örnek; WebView2 başlatma, HMI_Connect nesnesini JavaScript tarafına aktarma, HTML yükleme, PLC'ye veri yazma ve PLC'den veri okuma işlemlerini birlikte göstermektedir.

Örneği yeni bir formda kullanabilir veya mevcut formlarınızdan birinin Load olayına ekleyebilirsiniz. Aşağıdaki olay adı örnek amaçlı Form1_Form_Load olarak verilmiştir.

11.1 Form Load Örneği

```
private async void Form1_Form_Load(object sender, EventArgs e)
{
    // hmi_Label1 üzerinde seçilmiş olan Connect bilgisi alınıyor.
    // Bu Sayede MasterPage üzerinde tanımlı HMI_Connect nesnesine
    // başka bir formdan da erişilebilir.
    HMI_Connect hmi_Connect_PLC = hmi_Label1.Connect;

    // WebView2 başlatılıyor.
    await webView21.EnsureCoreWebView2Async();

    // JavaScript tarafına hmi_Connect_PLC adıyla obje aktarılıyor.
    webView21.CoreWebView2.AddHostObjectToScript(
        "hmi_Connect_PLC",
        new NikComWebBridge(hmi_Connect_PLC)
    );

    string html = @"
<!DOCTYPE html>
<html>
<head>
<meta charset='utf-8'>
<title>NikCom HMI WebView2 Test</title>
<style>
    body {
        font-family: Arial;
        background: #f2f2f2;
        padding: 30px;
    }
    h2 { color: #003b73; }
    button {
        font-size: 18px;
        padding: 12px 22px;
        margin: 6px;
        cursor: pointer;
        border: none;
        border-radius: 6px;
        background: #005bbb;
        color: white;
    }
    button.stop { background: #b00020; }
    button.read { background: #00843d; }
    #sonuc {
        margin-top: 20px;
        padding: 15px;
        background: white;
        border: 1px solid #ccc;
        font-size: 24px;
        color: #222;
        width: 350px;
    }
</style>
</head>
```

11.2 HTML ve JavaScript Bölümü

```
<body>
  <h2>NikCom HMI - WebView2 Test</h2>
  <button onclick="hmi_Connect_PLC.TagList_SetValue(25, true)">
    Motor Start
  </button>
  <button class='stop' onclick="hmi_Connect_PLC.TagList_SetValue(25, false)">
    Motor Stop
  </button>
  <button class='read' onclick="TagOku()">
    Tag Oku
  </button>
  <div id='sonuc'>Değer Bekleniyor...</div>
  <script>
    // WebView2 host object alias tanımı
    const hmi_Connect_PLC =
      chrome.webview.hostObjects.hmi_Connect_PLC;

    async function TagOku() {
      try {
        let veri = await hmi_Connect_PLC.TagList_GetValue(25);

        document.getElementById('sonuc').innerHTML =
          'Tag Değeri : ' + veri;
      }
      catch (err) {
        document.getElementById('sonuc').innerHTML =
          'Hata: ' + err;
      }
    }
  </script>
</body>
</html>";

  webView21.NavigateToString(html);
}
```

11.3 Bridge Class

Bu class C# ile JavaScript arasında köprü görevi görür. Kullanıcı HTML tarafında sadece TagList_GetValue ve TagList_SetValue metodlarını kullanır.

```
[ComVisible(true)]
public class NikComWebBridge
{
    private dynamic _connect;

    public NikComWebBridge(dynamic connect)
    {
        _connect = connect;
    }

    public object TagList_GetValue(int tagID)
    {
        try
        {
            return _connect.TagList_GetValue(tagID);
        }
        catch (Exception ex)
        {
            return "Read Error: " + ex.Message;
        }
    }

    public bool TagList_SetValue(int tagID, object value)
    {
        try
        {
            _connect.TagList_SetValue(tagID, value);
            return true;
        }
        catch
        {
            return false;
        }
    }
}
```

12. Harici HTML Dosyası Kullanımı

Örneklerde HTML kodları test amacıyla doğrudan C# kodu içerisinde string olarak yazılmıştır.

İstenirse HTML / JavaScript kodları ayrı bir .html dosyası içerisinde hazırlanabilir ve WebView2 içerisine ilgili fonksiyonlar ile yüklenebilir.

Bu yöntem özelliklerle:

- büyük projelerde
- React / Vue gibi yapılarda
- profesyonel web arayüzlerinde

daha düzenli ve yönetilebilir bir yapı sağlar.

Örneğin:

```
webView21.Source = new Uri(@"C:\\Web\\index.html");
```

veya

```
webView21.CoreWebView2.Navigate("file:///C:/Web/index.html");
```

13. Sonuç

Bu yapı ile WebView2 içinde çalışan HTML / JavaScript ekranlar, NikCom HMI üzerinden PLC taglerine erişebilir. Kullanıcı HTML tarafında basit kodlarla veri okuyup yazabilir.

NikCom HMI, Visual Studio ve WebView2 altyapısı sayesinde klasik WinForms yapıları ile modern web teknolojilerini aynı proje içinde bir araya getirebilir.