

1. Modbus TCP Nedir?

📌 Tanım

Modbus TCP, endüstriyel cihazlar arasında veri alışverişi sağlamak için kullanılan, TCP/IP tabanlı bir haberleşme protokolüdür.

Basitçe:

PC ↔ SCADA ↔ HMI ↔ PLC ↔ SÜRÜCÜ ↔ AKTÜATÖR
cihazlarının birbiriyle veri okuma ve yazma yapmasını sağlar.

NikCom HMI, Modbus TCP ile hem PLC üzerinden hem de doğrudan cihazlara bağlanarak esnek bir haberleşme yapısı sunar.

🔧 Nerelerde Kullanılır?

Modbus TCP yaygın olarak:

- PLC sistemlerinde
- HMI panellerde
- SCADA yazılımlarında
- Endüstriyel otomasyon sistemlerinde kullanılır.

⚙️ Çalışma Mantığı

Modbus TCP, **Client / Server (Master / Slave)** mantığı ile çalışır.

- **Client (İstemci):** Veri isteyen taraf
- **Server (Sunucu):** Veriyi sağlayan taraf

🔑 **NikCom HMI bu yapıda:** Client (İstemci) olarak çalışır.

🔑 **PLC ise:** Server (Sunucu) rolündedir.

🌐 TCP/IP Üzerinden Çalışma

Modbus TCP, Ethernet üzerinden çalışır ve standart olarak Port: 502 kullanır. Bu sayede cihazlar aynı ağ üzerinden kolayca haberleşebilir.

💡 Önemli Not

Modbus TCP:

- Basit
- Hızlı
- Yaygın

bir protokoldür, ancak doğru adresleme ve veri tipi kullanımı çok önemlidir.

⚡ Kısa Özet

Modbus TCP = Endüstriyel cihazlar arasında veri alışverişi sağlayan TCP/IP tabanlı protokol

📖 2. Modbus TCP Nasıl Çalışır?

🔗 Temel Mantık

Modbus TCP, **istemci (Client) ve sunucu (Server)** arasında çalışan bir haberleşme sistemidir.

Client → İstek gönderir

Server → Cevap verir

📦 Veri Paket Yapısı (Basit Anlatım)

Her Modbus TCP isteği:

- adres bilgisi
- veri uzunluğu
- işlem türü (okuma / yazma)

içerir

👉 Bu işlemler kullanıcıya gösterilmez

👉 **NikCom HMI arka planda otomatik yapar**

🔑 Function Code (Kısa Bilgi)

Modbus'ta her işlem bir kod ile yapılır:

- 0x01 → Coil okuma
- 0x03 → Holding Register okuma
- 0x05 → Coil yazma
- 0x06 → Register yazma

👉 Kullanıcı bunları manuel yazmaz

👉 **NikCom HMI arka planda otomatik yapar**

⚠️ Kritik Bilgi

Eğer bağlantı yoksa:

- IP yanlış olabilir
- Port yanlış olabilir
- PLC Modbus açık olmayabilir

3. Modbus Veri Tipleri

Modbus TCP'de veriler farklı türlerde tutulur.

Bu veri türleri, okunacak veya yazılacak bilginin yapısını belirler.

Desteklenen Veri Tipleri (NikCom HMI)

NikCom HMI içerisinde kullanılan temel veri tipleri:

C → Coil

DI → Discrete Input

HR → Holding Register

1. Coil (C)

1 bitlik (true / false) veri

- Açık / Kapalı
- 1 / 0
- On / Off

Kullanım alanı:

- Start / Stop
- Motor çalışıyor mu
- Röle durumu

2. Discrete Input (DI)

1 bitlik sadece okunabilir veri

- Sadece PLC'den okunur
- Yazma yapılamaz

Kullanım alanı:

- Sensör bilgileri
- Limit switch
- Dijital girişler

3. Holding Register (HR)

16 bit veri (en çok kullanılan)

- Okunabilir ✓
- Yazılabilir ✓

Kullanım alanı:

- Sayısal değerler
- Analog değerler
- Set değerleri
- Parametreler

Veri Tipi Detayı

Tip Boyut Okuma Yazma

C	1 bit	✓	✓
DI	1 bit	✓	✗
HR	16 bit	✓	✓

Önemli Not

Holding Register (HR):

Tüm sayısal işlemlerin temelidir

☞ Float, Int32 gibi veri tipleri

☞ birden fazla register kullanılarak oluşturulur

⚠ Dikkat Edilmesi Gerekenler

- Coil ve DI → **bit bazlıdır**
- HR → **word (16 bit) bazlıdır**

☞ Yanlış veri tipi seçilirse:
yanlış veri okunur

⚡ Kısa Özet

C = Yazılabilir bit

DI = Okunabilir bit

HR = 16 bit veri (en önemli)

3.1 NikCom HMI Adresleme Yapısı (Universal ve PLC Özel)

NikCom HMI, Modbus TCP haberleşmesinde iki farklı adresleme yöntemini destekler:

◆ 1. Universal Modbus Adresleme

C → Coil

DI → Discrete Input

HR → Holding Register

☞ Bu adresler:

- Tüm Modbus TCP cihazları için ortaktır
- PLC markasından bağımsız çalışır

✦ Bu nedenle:

HR, C, DI adresleri universal (genel) adresleme olarak kullanılır

☞ Farklı marka PLC veya cihazlara bağlanırken bu yapı kullanılır

◆ 2. PLC Özel Adresleme (Delta / Fatek)

NikCom HMI, bazı PLC markaları için **doğrudan kendi doğal adres formatlarını** destekler.

Desteklenen formatlar:

X → Input

Y → Output

M → Internal bit

D → Data register

☞ Bu yapı:

- Delta PLC
- Fatek PLC

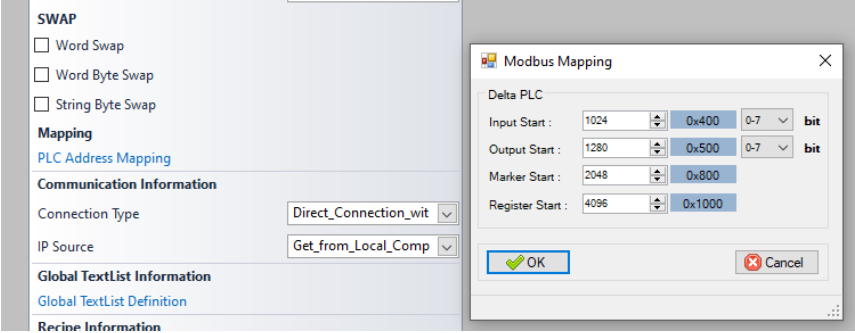
için doğrudan kullanılabilir

☞ Adres Mapping (Eşleme Mantığı)

Bu PLC'lerde kullanılan X, Y, M, D adresleri aslında:

Modbus adreslerine karşılık gelir

NikCom HMI, bu dönüşümü kullanıcı yerine otomatik yapar.



☞ Bu ayarlar ile:

- X → hangi Modbus adresine karşılık gelir
- Y → hangi adres aralığında
- M ve D → hangi register'lara denk gelir

belirlenebilir

💡 Önemli Bilgi

☞ Eğer:

Delta / Fatek PLC kullanıyorsanız

✓ X, Y, M, D adreslerini direkt kullanabilirsiniz

☞ Eğer:

Farklı marka PLC veya cihaz kullanıyorsanız

✓ Universal Modbus adresleri kullanılmalıdır:

HR / C / DI

◆ Bit ve Word Farkı

Bit (C / DI)

1 bit = 0 veya 1

☞ Örnek:

C10 → ON / OFF

Word (HR)

1 register = 16 bit

☞ Örnek:

HR100 → 0 – 65535 arası değer

🔢 Veri Tipi Oluşturma

Bazı veri tipleri birden fazla register kullanır:

- Int32 → 2 register
- Float → 2 register

☞ Örnek:

HR100 + HR101 → Float değer

💡 Önemli Nokta

Modbus adresleri:

PLC'nin gerçek hafızasını değil, haberleşme tablosunu temsil eder.

☞ cihaz içindeki veri tablosunun bir temsilidir

⚠ Dikkat

- PLC özel adresleme sadece desteklenen cihazlarda geçerlidir
- Universal adresleme Modbus TCP destekleyen tüm cihazlarda çalışır

↩ Kısa Özet

X Y M D → PLC özel (Delta / Fatek)

HR C DI → Modbus TCP destekleyen tüm cihazlar (Universal)

4. NikCom HMI Adresleme Yaklaşımı

Modbus TCP cihazlarında adresleme 0 tabanlı veya 1 tabanlı olabilir.

Bu durum farklı cihazlarda adres karmaşasına neden olabilir.

NikCom HMI bu karışıklığı azaltmak için belirli bir standart kullanır, ancak bu standart **tüm adresleme türleri için aynı değildir**.

◆ Modbus Adresleri Nasıl Okunur?

Modbus kullanan cihazların dokümanlarında kullanıcı genellikle şu tip adreslerle karşılaşır:

0, 1, 100

40001, 40010

00001

Bu adresler ilk bakışta karışık görünebilir.

NikCom HMI içerisinde ise bu adresler daha anlaşılır bir yapıya çevrilmiştir.

🔗 Universal Modbus Adresleme (HR / C / DI)

NikCom HMI içerisinde **Universal Modbus adresleri** için:

HR / C / DI adresleri 1'den başlar

☞ Yani:

HR1 → ilk register

HR2 → ikinci register

🔧 Arka Plan Mantığı

Kullanıcı:

HR1

yazdığında sistem:

HR0

adresine erişir

☞ Yani:

Sistem girilen değerden 1 çıkarır

🔗 Modbus Adreslerinin Karşılığı

NikCom HMI'da kullanılan adresler:

HR → Holding Register (Sayısal veri)

C → Coil (Yazılabilir bit)

DI → Discrete Input (Okunabilir bit)

Dokümandaki Adres → NikCom Karşılığı

Dokümanda NikCom HMI


40001	HR1
40010	HR10
00001	C1

 Yani:

40001 → HR1

Ne Nerede Kullanılır?


◆ HR (Holding Register)

 Sayısal değerler için kullanılır

- Sıcaklık
- Basınç
- Hız
- Set değerleri

Örnek: HR10 → sıcaklık değeri


◆ C (Coil)

 Yazılabilir bit (On/Off)

- Start / Stop
- Motor çalıştırma

Örnek: C1 → Start komutu

◆ DI (Discrete Input)

 Sadece okunur (sensör verisi)


- Sensör
- Limit switch
- Dijital giriş

Örnek: DI5 → sensör aktif mi?

PLC Tarafındaki Karşılıklar (Delta Örneği)

NikCom HMI PLC

HR	D
C	M / Y
DI	X

 Örnek: (Bu değerler mapping ayarına göre değişir)

D100 → HR4196

X0 → DI1024

M10 → C2058

💡 Önemli Not

Cihaz dokümanlarında görülen:

40001, 0, 1

gibi adresler aslında aynı veriyi farklı şekilde ifade edebilir.

☞ NikCom HMI:

HR / C / DI

kullanarak bu yapıyı sadeleştirir.

40001 → HR1

HR → sayısal veri

C → yazılabilir bit

DI → sensör (okunur bit)

🔗 2. PLC Özel Adresleme (X / Y / M / D)

PLC özel adresleme (Delta / Fatek):

X / Y / M / D

☞ Bu adreslerde:

1 tabanlı standart ZORUNLU DEĞİLDİR

☞ Çünkü:

- Bu adresler doğrudan PLC mantığına bağlıdır
- Mapping ayarına göre değişebilir

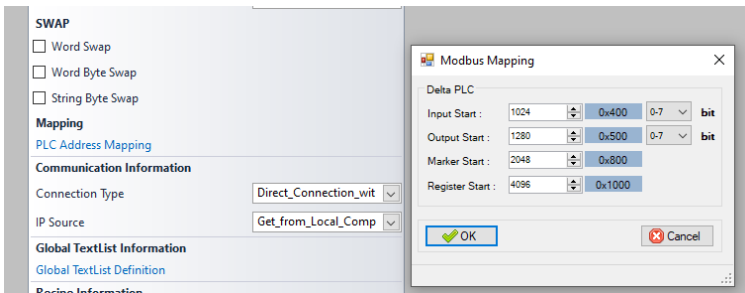
🔄 Mapping Kullanımı

Eğer Address Mapping kullanılıyorsa:

- X, Y, M, D adresleri
- Modbus adreslerine çevrilir

☞ Bu durumda:

0 / 1 farkı mapping üzerinden etkilenebilir



Delta PLC için PLC Address Mapping e Tıklandığında , X Y M ve D için Adreslerin Modbus TCP Başlangıç karşılıkları girilebilir veya değiştirilebilir.

⚠ En Kritik Nokta

Universal adresleme ile PLC adresleme **aynı değildir**

HR / C / DI → her zaman 1 tabanlı kullanılır

X / Y / M / D → PLC'ye bağlıdır

🔍 Doğru Adres Nasıl Bulunur?

Eğer veri okunamıyorsa veya yanlış geliyorsa:

👉 ilk yapılması gereken:

Adresi +1 veya -1 değiştirerek test etmek

★ Örnek

Beklenen veri gelmiyorsa:

HR100 yerine HR101 dene

veya

HR99 dene

Aynı şekilde:

D100 yerine D101 veya D99 dene

💡 Önemli Kural

Modbus'ta veri gelmiyorsa ilk şüphe adres ofsetidir (0 / 1 farkı)

📦 Kullanıcıya Tavsiye

- Cihaz dokümanını kontrol et
- İlk adresin 0 mı 1 mi olduğunu tespit et
- Mapping varsa dikkate al

⚡ Kısa Özet

HR / C / DI → 1 tabanlı (NikCom standardı)

X / Y / M / D → PLC'ye bağlı

👉 Veri gelmiyorsa:

Adresi ± 1 değiştir

5. 40001 / 00001 Nedir?

Modbus sistemlerinde bazı cihazlar adresleri farklı formatlarda gösterir.

En yaygın olan gösterim şekilleri:

40001, 40002, 40003 ...

00001, 00002, 00003 ...

Bu Sayılar Ne Anlama Gelir?

Bu gösterimler aslında:

Veri türünü belirtmek için kullanılan eski Modbus gösterimidir

Yaygın Gösterimler

Gösterim	Anlamı
00001	Coil (C)
40001	Holding Register (HR)


 Yani:

40001 = HR adres alanı

Kritik Nokta

Bu sayılar:

gerçek adres değildir

 Sadece:

adres tipini gösterir

Gerçek Adrese Dönüşüm

Örnek:

40001 → HR1

40002 → HR2


 Yani:

40001 = HR1 (ilk register)

NikCom HMI Kullanımı

NikCom HMI içerisinde:

40001 gibi adresler kullanılmaz

 Bunun yerine:

HR1, HR2, HR3 ...

kullanılır

🔍 Örnek

Cihaz dokümanında:

40010

yazıyorsa:

👉 NikCom HMI'da:

HR10

kullanılır

⚠ Dikkat Edilmesi Gerekenler

Bazı cihazlarda:

- 40001 → HR0 olabilir
- bazı cihazlarda → HR1 olabilir

👉 Bu durum:

0 / 1 adresleme farkından kaynaklanır

★ Kullanıcıya Tavsiye

Eğer veri doğru gelmiyorsa:

HR değerini ± 1 değiştirerek test et

↩ Kısa Özet

40001 = HR1 (NikCom içinde)

👉 Ama:

0 / 1 farkına dikkat et

6. Veri Tipleri ve Swap Ayarları

Modbus TCP’de veriler register yapısında tutulur.

Bazı cihazlar veriyi farklı sırada gönderdiği için, okunan değer doğru adreste olsa bile yanlış görünebilir.

Bu durumda **Swap ayarları** kullanılır.

◆ **Swap Nedir?**

Swap, verinin cihazdan farklı sırada gelmesi durumunda, sıralamayı düzeltmek için kullanılan işlemdir.

NikCom HMI içerisinde HMI_Connect nesnesi üzerinden şu seçenekler bulunur:

- **Word Swap**
- **Word Byte Swap**
- **String Byte Swap**

Bu ayarlar sayesinde veri doğru sıraya getirilir.

◆ **Word Yapısı**

Bir Holding Register:

1 HR = 16 bit = 2 byte

Bu iki byte:

- High Byte
- Low Byte

olarak düşünülebilir.

◆ **Word Byte Swap**

Word Byte Swap, tek word içindeki byte’ların yerini değiştirir.

✦ Yani:

AB → BA

🕒 **Ne zaman kullanılır?**

Küçük sayılar büyük görünüyorsa , yada tam tersi durumlarda.

✦ **Örnek**

- Beklenen: 1
- Gelen: 256

👉 **Word Byte Swap açılmalıdır**

◆ Word Swap

Word Swap, 32 bit verilerde iki register'ın yerini deęiřtirir.

✦ Örnek:

HR100 + HR101

yerine sistem:

HR101 + HR100

olarak yorumlar.

🕒 Ne zaman kullanılır?

Float deęer anlamsız geliyorsa.

✦ Örnek

- Beklenen: 25.5
- Gelen: 3.4E+38

👉 **Word Swap açılmalıdır**

◆ String Byte Swap

String Byte Swap, string verilerde karakter çiftlerinin sırasını deęiřtirir.

✦ Örnek

- Beklenen: abcd
- Gelen: badc

👉 **String Byte Swap açılmalıdır**

◆ NikCom HMI'da Swap

NikCom HMI içinde HMI_Connect nesnesinde řu seçenekler bulunur:

- Word Swap
- Word Byte Swap
- String Byte Swap

Kullanıcı, cihazdan gelen veriye göre uygun swap seçeneęini açmalıdır.

◆ Kısa Özet

Float yanlış geliyorsa → Word Swap

1 deęeri 256 geliyorsa → Word Byte Swap

abcd yerine badc geliyorsa → String Byte Swap

7. Modbus TCP’de En Sık Yapılan Hatalar

Modbus TCP haberleşmesinde karşılaşılan sorunların büyük çoğunluğu aşağıdaki temel hatalardan kaynaklanır.

◆ 7.1 Adres Hatası (0 / 1 Farkı)

Veri gelmiyorsa veya yanlış geliyorsa:

Adres ± 1 kontrol edilmelidir

✦ Sebep:

Cihazın 0 tabanlı veya 1 tabanlı adresleme kullanması

◆ 7.2 Yanlış Veri Tipi Seçimi

Yanlış veri tipi seçildiğinde:

Okunan değer hatalı olur

✦ Örnek:

- Float yerine Int seçilirse
- 32 bit veri 16 bit okunursa

◆ 7.3 Swap Ayarlarının Yanlış Olması

Veri doğru adreste olmasına rağmen yanlış görünüyor:

Swap ayarları kontrol edilmelidir

✦ Örnek:

- Float değer anlamsız geliyorsa
- 1 değeri 256 geliyorsa
- String bozuk görünüyor

◆ 7.4 Yanlış Register Türü

Yanlış adres tipi seçildiğinde veri okunamaz:

C / DI / HR doğru seçilmelidir

✦ Örnek:

- HR yerine C okunmaya çalışılırsa veri gelmez

◆ 7.5 Mapping Ayarlarının Yanlış Olması

PLC özel adresleme kullanılıyorsa:

Mapping ayarları kontrol edilmelidir

✦ Özellikle:

- X / Y / M / D adreslerinde

◆ 7.6 Cihaz Dokümanının Dikkate Alınmaması

Her cihazın adres yapısı farklı olabilir:

Doğru adres için cihaz dokümanı incelenmelidir

◆ 7.7 Kısa Özet

Adres yanlış → ±1 kontrol et

Veri yanlış → Swap kontrol et

Veri yok → Register tipi kontrol et

8. Bağlantı Sorunlarında Kontrol Listesi

Modbus TCP bağlantısında sorun yaşandığında aşağıdaki adımlar kontrol edilmelidir.

◆ 8.1 Ağ Ayarları

- IP adresi doğru mu?
- PLC ile aynı ağda mı?
- Ping atılabiliyor mu?

◆ 8.2 Port Kontrolü

- Port 502 açık mı?
- Firewall engelliyor olabilir mi?

◆ 8.3 PLC Ayarları

- PLC’de Modbus TCP aktif mi?
- Doğru port kullanılıyor mu?

◆ 8.4 Adres Kontrolü

- Doğru adres girildi mi?
- 0 / 1 farkı kontrol edildi mi?

◆ 8.5 Veri Tipi ve Swap

- Veri tipi doğru mu?
- Swap ayarları doğru mu?

◆ 8.6 Register Türü

- C / DI / HR doğru seçildi mi?

◆ 8.7 Kısa Özet

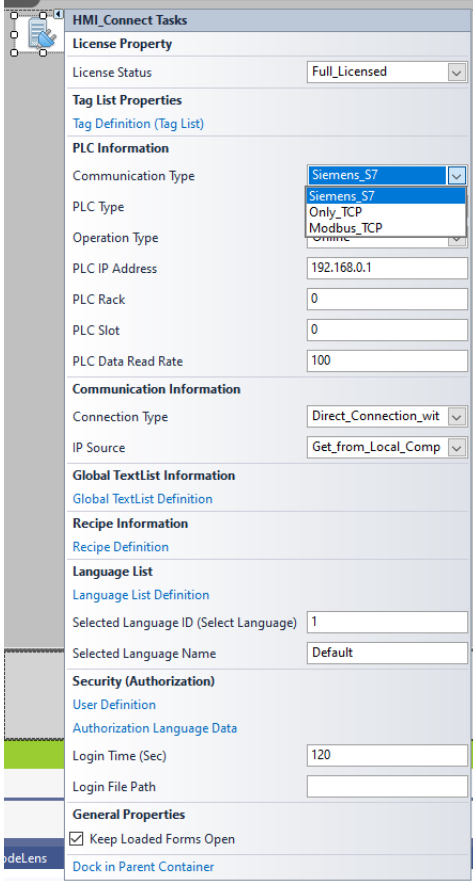
Bağlantı yok → IP / Port kontrol et

Veri yok → adres kontrol et

Veri yanlış → Swap ve veri tipi kontrol et

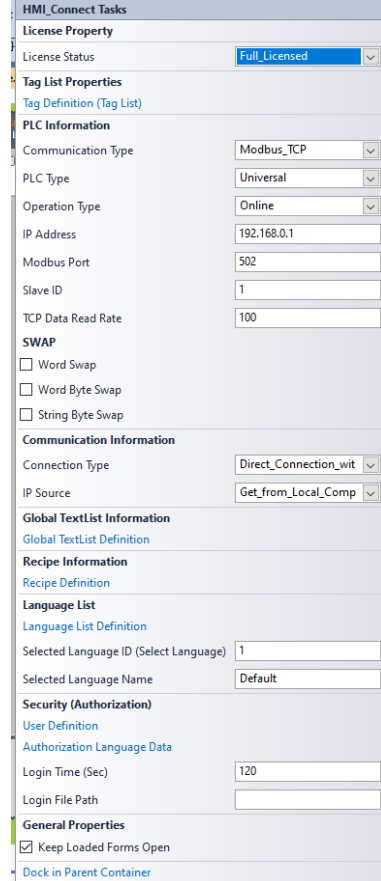
9. NikCom HMI ile Modbus TCP Kullanımı

NikCom HMI da Modbus TCP , HMI Connect nesnesinin bir iletişim özelliğidir. Modbus TCP için gerekli parametreler ,HMI Connect nesnesi üzerinden yapılacaktır.



The screenshot shows the 'HMI_Connect Tasks' configuration window. The 'PLC Information' section is expanded, and the 'Communication Type' dropdown menu is open, showing 'Modbus_TCP' selected. Other visible settings include 'License Status' set to 'Full_Licensed', 'PLC IP Address' as '192.168.0.1', and 'PLC Data Read Rate' as '100'.

Öncelikle Communication Type den Modbus TCP i Seçiniz



The screenshot shows the 'HMI_Connect Tasks' configuration window. The 'PLC Information' section is expanded, and the 'PLC Type' dropdown menu is open, showing 'Universal' selected. Other visible settings include 'Communication Type' set to 'Modbus_TCP', 'IP Address' as '192.168.0.1', and 'Modbus Port' as '502'.

Modbus TCP için gerekli PLC Type ini PLC Type den Seçebilirsiniz

HMI_Connect Tasks

License Property

License Status: Full_Licensed

Tag List Properties

Tag Definition (Tag List)

PLC Information

Communication Type: Modbus_TCP

PLC Type: Universal

Operation Type: Delta

IP Address:

Modbus Port: 502

Slave ID: 1

TCP Data Read Rate: 100

SWAP

Word Swap

Word Byte Swap

String Byte Swap

Communication Information

Connection Type: Direct_Connection_wit

IP Source: Get_from_Local_Comp

Global TextList Information

Global TextList Definition

Recipe Information

Recipe Definition

Language List

Language List Definition

Selected Language ID (Select Language): 1

Selected Language Name: Default

Security (Authorization)

User Definition

Authorization Language Data

Login Time (Sec): 120

Login File Path:

General Properties

Keep Loaded Forms Open

[Dock in Parent Container](#)

Delta / Fatek PLC kullanıyorsanız

✓ X, Y, M, D adreslerini direkt kullanabilirsiniz

☞ Eğer:

Farklı marka PLC veya cihaz kullanıyorsanız

✓ Universal Modbus adresleri kullanılmalıdır:

HR / C / DI adresleme ile kullanabilirsiniz

Modbus TCP'de veriler register yapısında tutulur.

Bazı cihazlar veriyi farklı sırada gönderdiği için, okunan değer doğru adreste olsa bile yanlış görünebilir.

Bu durumda **Swap ayarları** kullanılır.

Swap Nedir ? ☞ [Tıklayınız](#) (Ctrl + Tıklat)

Aşağıdaki ekranda Universal Modbus adresleri ile Tag Tanımlamayı anlatacağız.

◆ Universal Modbus Nedir?

Universal Modbus, tüm Modbus TCP cihazlarında ortak olarak kullanılan **standart adresleme yapısıdır**.

HR → Holding Register (Sayısal veri) Örn. Deltada D

C → Coil (Yazılabilir bit) Örn. Deltada Y,M

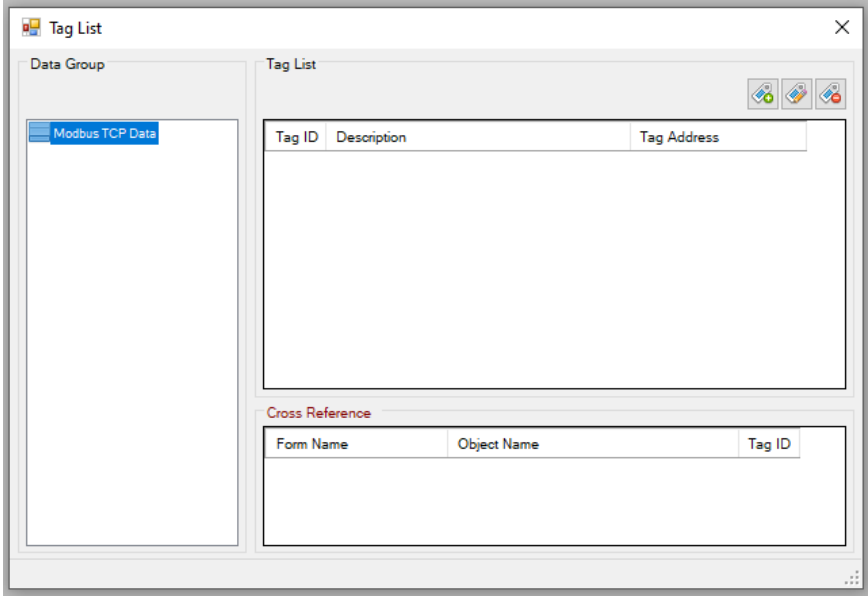
DI → Discrete Input (Okunabilir bit) Örn. Deltada X

Bu adresleme yöntemi:

- PLC markasından bağımsızdır
- Tüm Modbus uyumlu cihazlarda çalışır

☞ Bu nedenle farklı marka PLC veya cihazlara bağlanırken **Universal Modbus adresleri** kullanılır.

Detaylı Bilgi İçin ☞ [Tıklayınız](#) (Ctrl + Tıklat)



HMI Connect Nesnesinde Tag Tanımlamak için tıkladığınız Tag List Defination Özelliğine Tıkladığınızda artık Modbus TCP için düzenlenmiş Tag List Tanımlama ekranı gelir

Öncelikle Servonun Pozisyon Değerini Alan Adresi Tanımlayalım. Bu adresin 40001 ile İlk Register adresi olduğunu varsayıyoruz , Universal Modbus Karşılığı İse HR1 dir (Adreslemede anlatılan 0 veya 1 den Başlama konusunu dikkate alınız)

Add Tag İle Tag Ekleme ekranını çağırınız

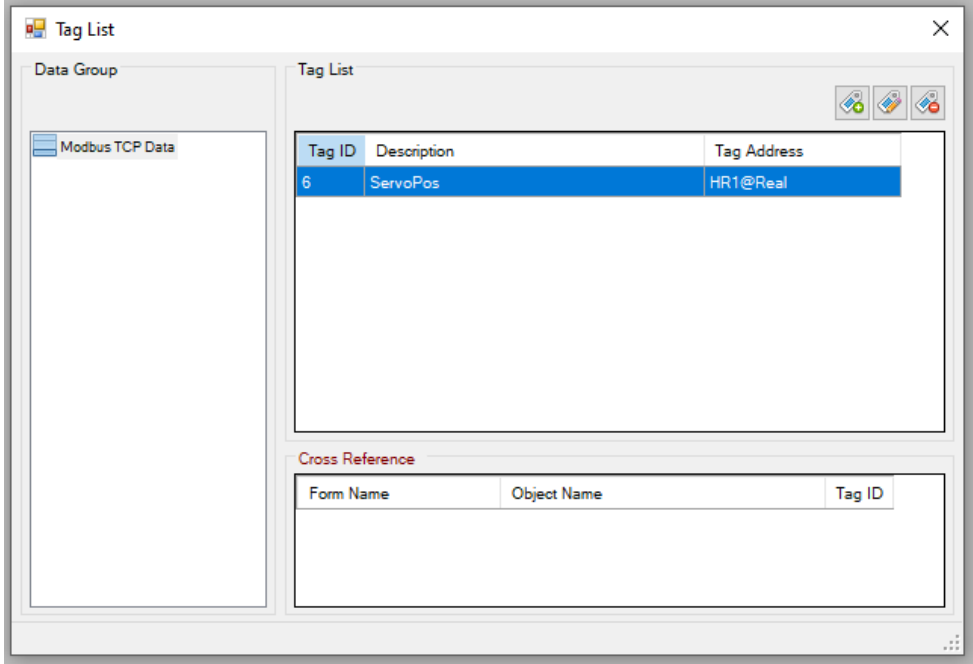
Ekleyeceğimiz Adres İçin Data Type Seçiniz
Bu adres için Real seçimi yapınız

Sayısal bir değer alacağımız için HR seçimi yapınız

Data Type Area ve No Girdikten Sonra Açıklama yada ServoPos giriniz , HR1@Real , Klasik adreslemede 40001 ve 40002 i alır bu Wordleri Float a çevirir. Önemli uyarı: Adreslemede 0 / 1 farkına dikkat et , Eğer veri doğru gelmiyorsa: HR değerini ± 1 değiştirerek test et

Burada **Number of Tags Özelliği** ile Ard arda tag tanımlamasıda yapabiliriz. Örneğin 5 Yazarsak HR1 den başlayıp sırası ile Real tipinde 5 adet Tag Tanımlar

ServoPos Tag List Görüntüsü aşağıdaki gibidir



Çıkış için Aşağıdaki Gibi Tag Ekleyebilirsiniz

The "Add Tag" dialog box shows the following configuration:

- Data Type: Bool
- Modbus Area: C
- No: 1
- View: C1@Bool
- Number of Tags: 1
- Description: Piston Forward

Buttons: Add, Cancel

Piston İleri Çıkışı

Giriş için Aşağıdaki Gibi Tag Ekleyebilirsiniz

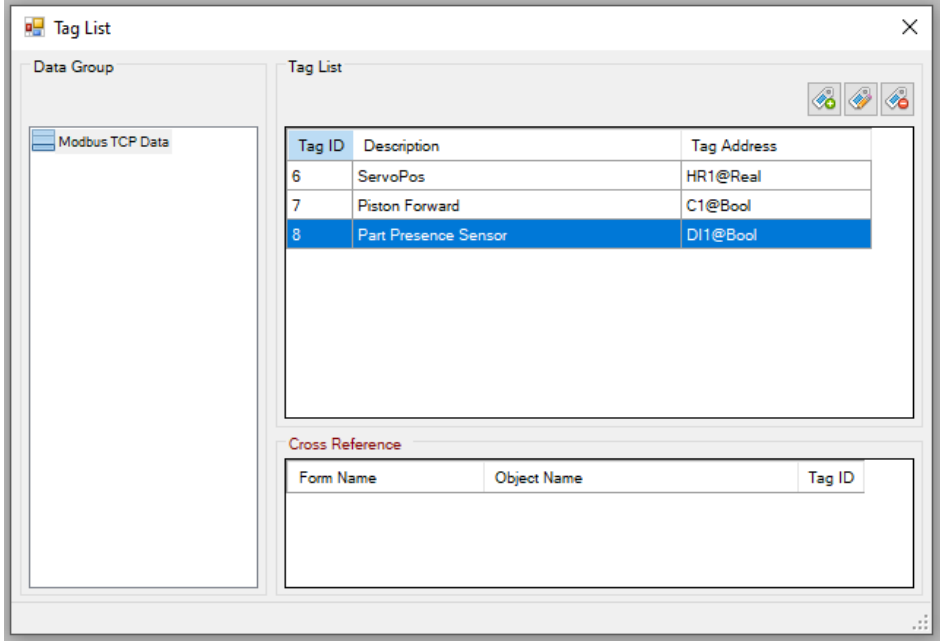
The "Add Tag" dialog box shows the following configuration:

- Data Type: Bool
- Modbus Area: DI
- No: 1
- View: DI1@Bool
- Number of Tags: 1
- Description: Part Presence Sensor

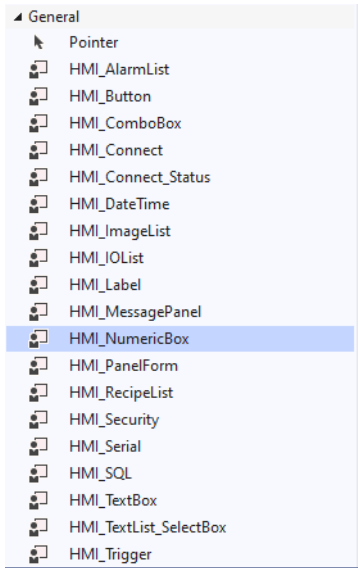
Buttons: Add, Cancel

Parça Varlık Sensörü Girişi

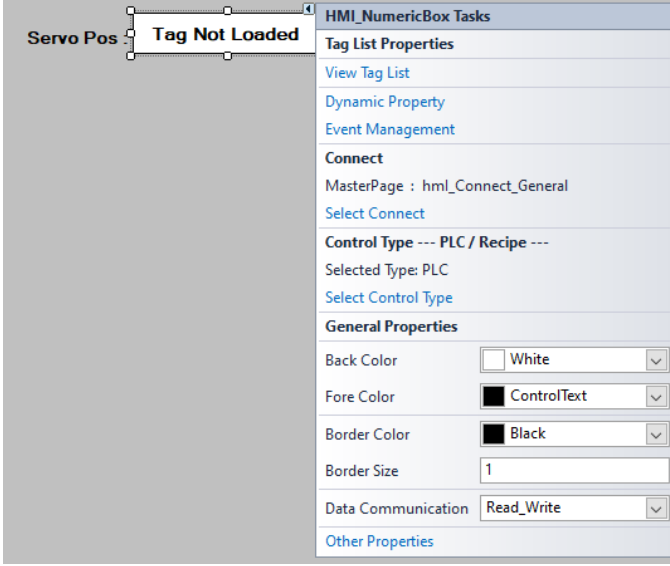
Eklediğimiz 3 Tag ın Tag List Görüntüsü Aşağıdaki Gibidir



Artık Bu Tagları Sırası İle HMI_NumericBox , HMI_Button ve HMI_Label a bağlayabiliriz



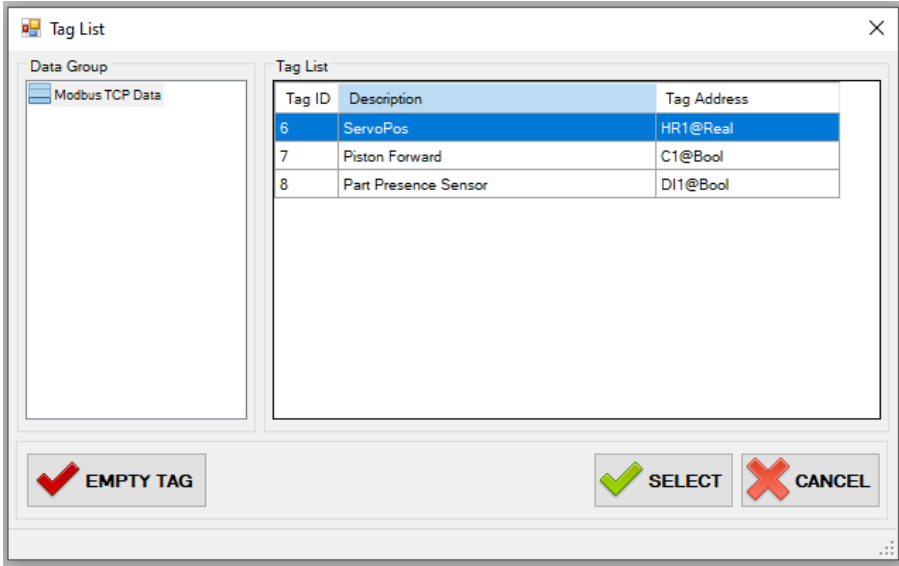
ToolBox dan **HMI_NumericBox** u Sürükleyip Bırakarak İle Formumuza Taşıyalım



HMI_NumericBox da öncelikle Connect Seçimi Yapalım , Select Connect ile Modbus TCP Tanımlaması yaptığımız HMI_Connect Nesnesini Seçelim

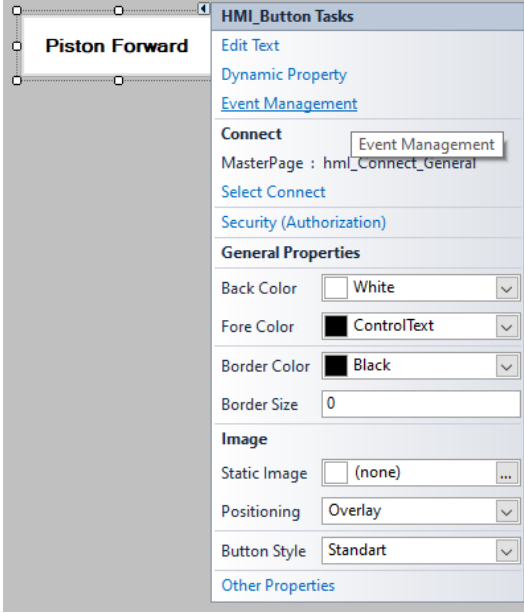
Daha Sonra İse **View Tag List** ile ServoPos Tagını HMI_NumericBox a bağlayalım

ServoPos için Tanımladığımız HR1@Real Tagını HMI_NumericBox için Seçiyoruz



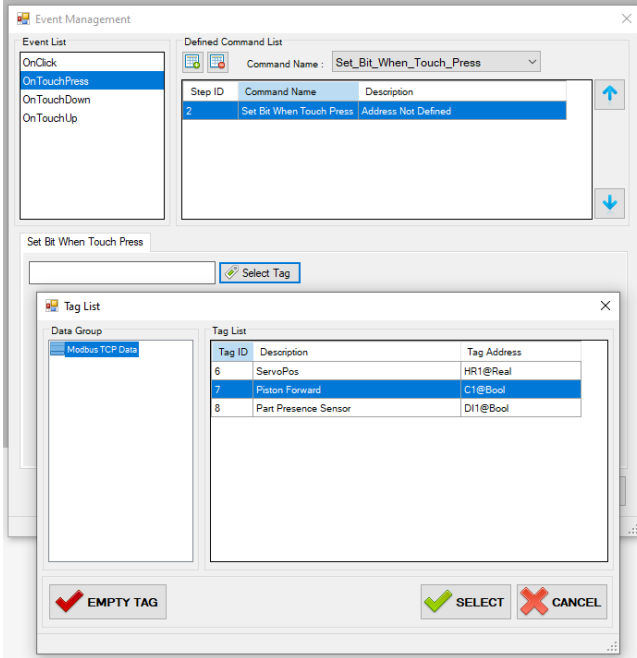
Artık HMI_NumericBox HR1@Real Adresine Bağlandı

Formumuza 1 Adet **HMI_Button** Ekleyim Piston İleri Komutunu Bu Button İle PLC e gönderebiliriz



HMI_Button da öncelikle Connect Seçimi Yapalım , Select Connect İle Modbus TCP Tanımlaması yaptığımız HMI_Connect Nesnesini Seçelim

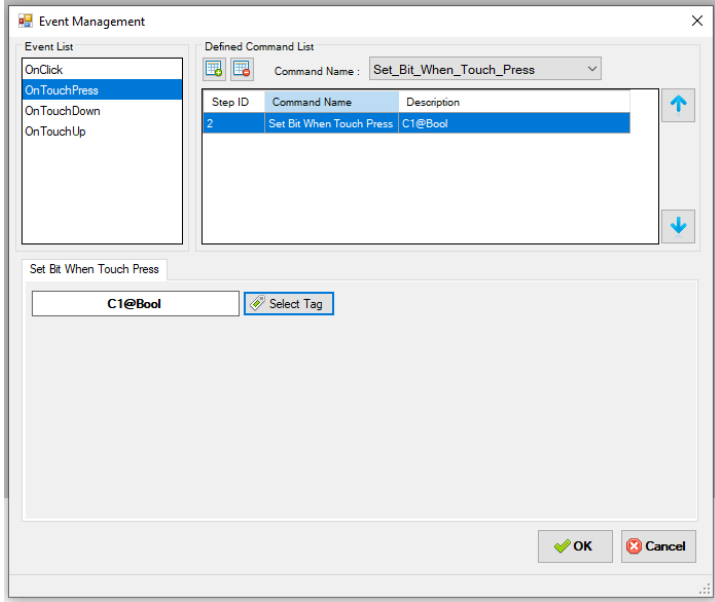
Daha Sonra ise **Event Managment** ile gerekli Komutu yazalım



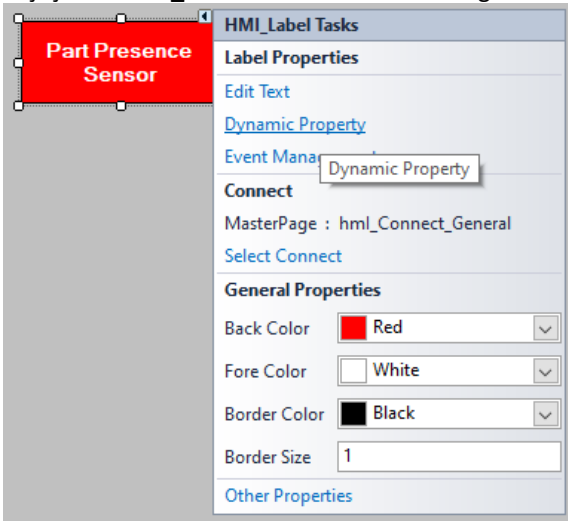
OnTouchPress Olayına SetBitWhenTouchPress Komutunu Ekliyoruz , Böylelikle Buttona tıkladığımız süre boyunca ilgili bit True olacak , tıklamayı bıraktığımızda False olacak

Tag List den Piston ileri C1@Bool Tagını seçiniz

C1@Bool Tagı Seçildi , OK tıklayınız

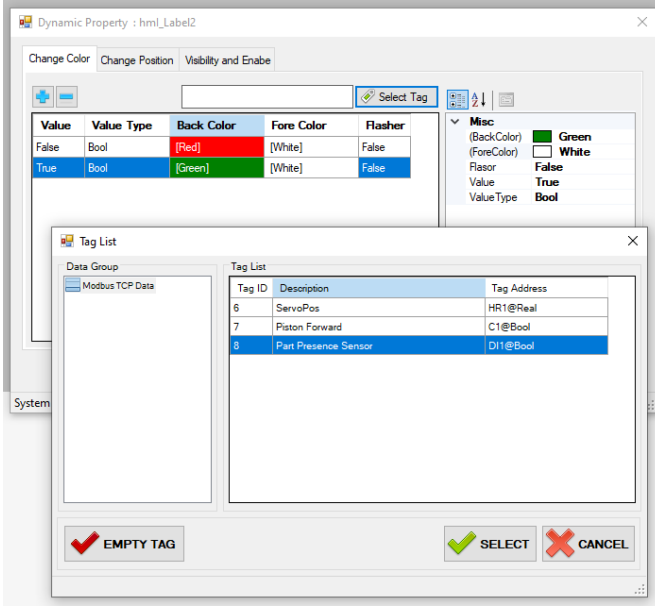


Giriş için ise **HMI_Label** kontrolünü kullanacağız



HMI_Label de öncelikle Connect Seçimi Yapalım , Select Connect ile Modbus TCP Tanımlaması yaptığımız HMI_Connect Nesnesini Seçelim

Daha Sonra ise HMI_Label imizi , **Dynamic Property** ile Giriş Değeri True ise Yeşil , False ise Kırmızı olacak şekilde ayarlayacağız



Change Color Özelliğine iki adet satır ekleyelim , Value Type lerini Bool yapalım ,

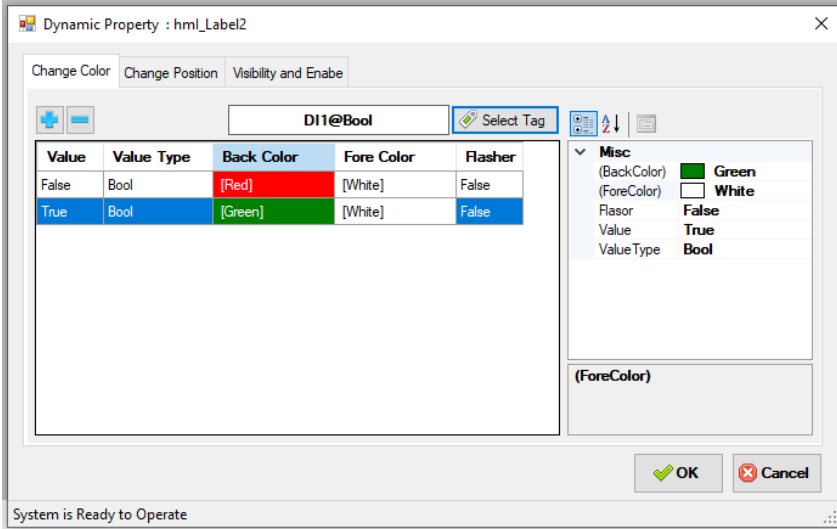
İlk satır ın **Valuesini** False **Back Color** unu Red , **Fore Color** unu White yapalım

İkinci satır ın **Valuesi** True **Back Color** unu Green , **Fore Color** unu White yapalım

Select Tag ile Bu özellikleri , değeri ile kontrol edecek Tagı Seçelim

Parça varlık sensörünün tagı olan DI1@Bool u seçiyoruz

Dynamic Property nin son hali aşağıdaki gibidir



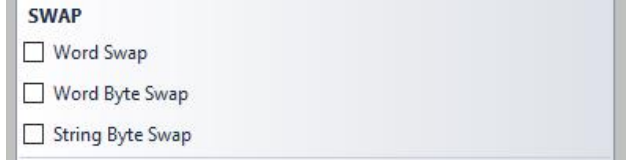
DI1@Bool Değerine Göre HMI_Label imizin Arka Plan ve Yazı Rengi Değişmektedir , bu sayede parça varlık sensörünün durumunu anlayabilirsiniz.

🔄 SWAP Ayarları

Swap, verinin cihazdan farklı sırada gelmesi durumunda, sıralamayı düzeltmek için kullanılan işlemdir.

NikCom HMI içerisinde **HMI_Connect** nesnesi üzerinden şu seçenekler bulunur:

- **Word Swap**
- **Word Byte Swap**
- **String Byte Swap**



SWAP

Word Swap

Word Byte Swap

String Byte Swap

Bu ayarlar sayesinde veri doğru sıraya getirilir.

Veriler aşağıdaki gibi geliyorsa mutlaka Swap yapmalısınız

Float yanlış geliyorsa → Word Swap

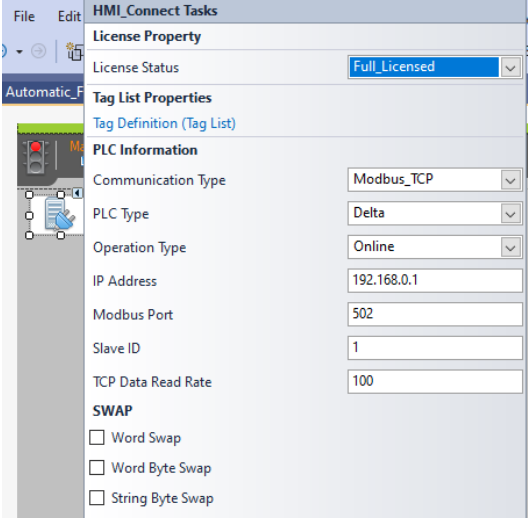
1 değeri 256 geliyorsa → Word Byte Swap

abcd yerine badc geliyorsa → String Byte Swap

Daya detaylı bilgi için [👉 Tıklayınız](#) (Ctrl + Tıklat)

İşlemlerimizi **Universal Modbus** adresleri ile yaptık , **Artık PLC Seçimi ile Direkt olarak PLC** nin adreslerini bağlayabiliriz.

🔄 PLC Seçimi ile Direkt Bağlanma (Delta Örneği)



Öncelikle **HMI_Connect** Nesnemizde **PLC Type** Özelliğini **Delta** olarak değiştiriniz

Universal Modbus Adresleme Delta PLC Adresleme

HR	D
C	M / Y
DI	X

🔗 Örnek: Delta Mapping e Göre

D100 → HR4196

X0 → DI1024

M10 → C2058

PLC Address Mapping ile ilgili Adres gruplarının başlangıç adreslerini yazabilirsiniz , Adreslerin (0-7) bit aralığında mı , yoksa (0-10) bit aralığında mı olduğunu seçebilirsiniz ,

Adresler (0-7) Bit aralığında ise Aşağıdaki gibi Listelenir

X0...X7

X10...X17

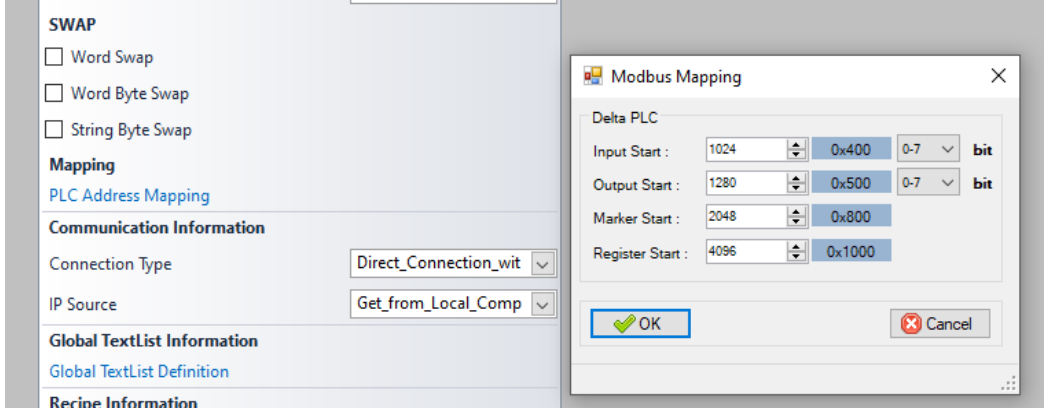
X20...X20

Adresler (0-10) Bit aralığında ise Aşağıdaki gibi Listelenir

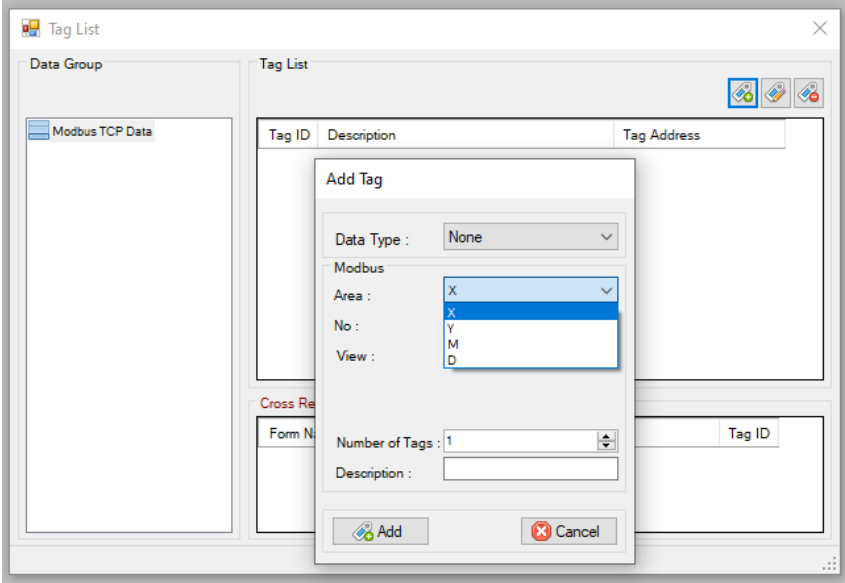
X0...x9

X10...X19
X20... X29

PLC nin adres yapısına göre bit aralığını seçiniz



HMI Connect Nesnesinde Tag Tanımlamak için tıkladığınız **Tag List Definition** Özelliğine Tıkladığınızda artık Modbus TCP için düzenlenmiş **Delta PLC** e göre Tag List Tanımlama ekranı gelir



Universal Adresleme de tanımladığımız tagları , Delta PLC içinde tanımlayalım

Add Tag

Data Type : Real

Modbus Area : D

No : 0

View : D0@Real

Number of Tags : 1

Description : Servo_Pos

Add Cancel

D0 adresini Real veri tipinde Servo Pos için Tanımlıyoruz

Burada **Number of Tags Özelliği** ile Ard arda tag tanımlamasıda yapabiliriz. Örneğin 5 Yazarsak D0 dan başlayıp sırası ile Real tipinde 5 adet Tag Tanımlar

Diğer Tag Adreslerininide (Piston İleri Y0@Bool ve Parça Varlık Sensörü X0@Bool) Delta PLC adreslemesine göre tanımlayınız

Tag List

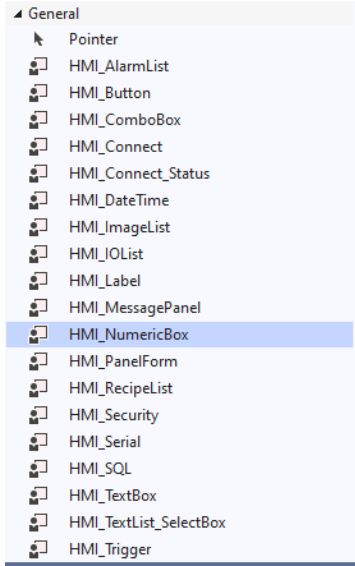
Data Group: Modbus TCP Data

Tag ID	Description	Tag Address
9	Servo_Pos	D0@Real
10	Piston Forward	Y0@Bool
11	Part Presence Sensor	X0@Bool

Cross Reference

Form Name	Object Name	Tag ID
-----------	-------------	--------

Artık Bu Tagları Sırası İle **HMI_NumericBox** , **HMI_Button** ve **HMI_Label** a bağlayabiliriz

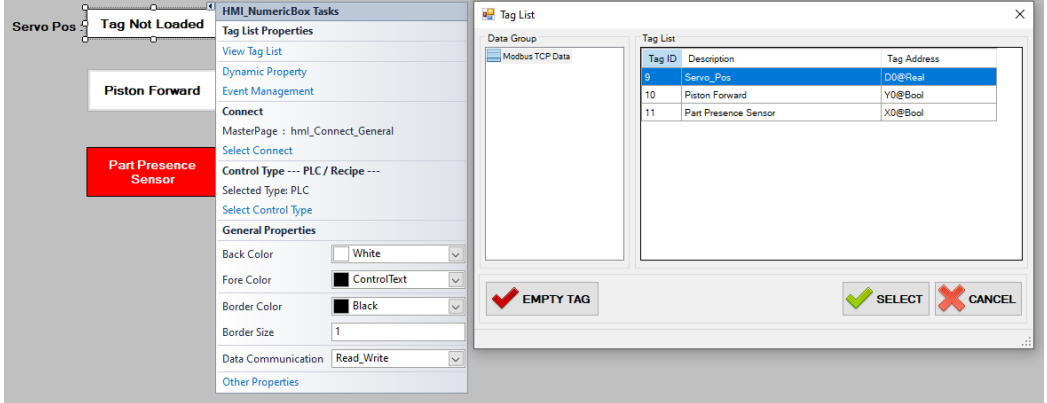


ToolBox dan **HMI_NumericBox** u Sürükle Bırak ile Formumuza Taşıyalım

Ayrıca Formumuza **HMI_Button** ve **HMI_Label** ekleyelim

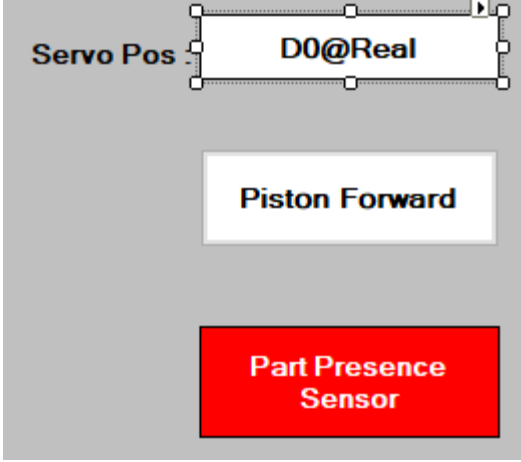
HMI_NumericBox da öncelikle Connect Seçimi Yapalım , Select Connect İle Modbus TCP Tanımlaması yaptığımız **HMI_Connect** Nesnesini Seçelim

Servo Pos (D0@Real) u **HMI_NumericBox** a Tag List View özelliğini kullanarak bağlayınız



Universal Adreslemede anlatıldığı gibi **HMI_Button un Event Management** özelliği ve **HMI_Label in Dynamic Property** özelliği ile ilgili tagları ilgili Kontrollerin özelliklerine bağlayalım.

Artık Nikcom HMI ı , Modbus TCP ile Delta PLC de kullanabilirsiniz



Kontrollerin Form üzerindeki görünümü yandaki gibidir.

10. İki Farklı PLC Arasında Veri Aktarımı (Kod ile)

NikCom HMI ile farklı marka PLC'ler arasında veri alışverişi yapmak oldukça kolaydır. Bu örnekte, **Delta PLC'den okunan veriler Siemens PLC'ye aktarılacaktır.**

Senaryo

- Delta PLC → veri okunur
- Siemens PLC → aynı veri yazılır

Bu işlem **TagList yapısı** kullanılarak gerçekleştirilir.

Yapılan Hazırlıklar

◆ 1. İki ayrı HMI_Connect nesnesi oluşturulur

- hmi_Connect_Delta → Delta PLC bağlantısı
- hmi_Connect_Siemens → Siemens PLC bağlantısı

◆ 2. Her iki PLC için Tag tanımları yapılır

- Delta PLC → 5 adet Word Tag
- Siemens PLC → 5 adet Word Tag

HMI_Connect name sini property ekranından **hmi_Connect_Delta** yapınız , gerekli özellikleri (IP vb .) yazınız

Delta PLC → 5 adet Word Tag Tanımlanıyor , Aktarmalar Tag ID e göre yapılacağı için Tag ID i not alınız.

The screenshot displays the software interface for configuring a Tag List. On the left, the 'Tag List Properties' dialog box is open, showing the 'Tag Definition (Tag List)' section. The 'PLC Information' section is expanded, showing the following settings:

- Communication Type: Modbus_TCP
- PLC Type: Delta
- Operation Type: Online
- IP Address: 192.168.0.1
- Modbus Port: 502
- Slave ID: 1
- TCP Data Read Rate: 50

The 'SWAP' section has three checkboxes: Word Swap, Word Byte Swap, and String Byte Swap, all of which are unchecked. The 'Mapping' section has 'PLC Address Mapping' selected. The 'Communication Information' section has 'Connection Type' set to 'Direct_Connection_wit' and 'IP Source' set to 'Get_from_Local_Comp'.

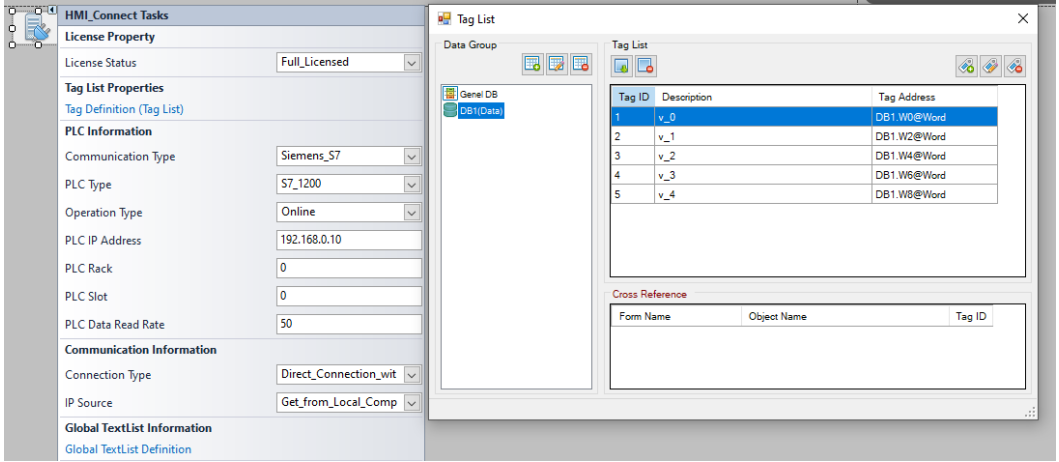
On the right, the 'Tag List' dialog box is open, showing a table with the following data:

Tag ID	Description	Tag Address
1	v_0	D0@Word
2	v_1	D1@Word
3	v_2	D2@Word
4	v_3	D3@Word
5	v_4	D4@Word

Below the table, there is a 'Cross Reference' section with a table that has three columns: Form Name, Object Name, and Tag ID. The table is currently empty.

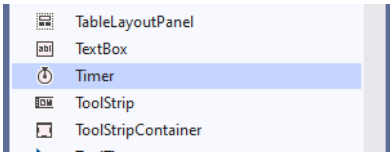
HMI_Connect name sini property ekranından **hmi_Connect_Siemens** yapınız , gerekli özellikleri (IP vb .) yapınız

Siemens PLC → 5 adet Word Tag Tanımlanıyor , Aktarmalar Tag ID e göre yapılacağı için Tag ID i not alınız.

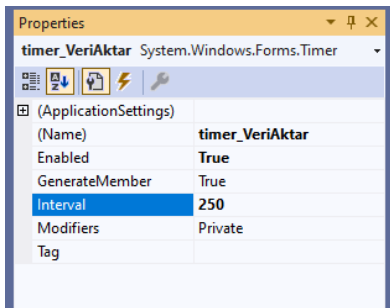


3. Timer ile Veri Aktarımı

Veri aktarımı belirli aralıklarla çalışması için bir Timer kullanılır.



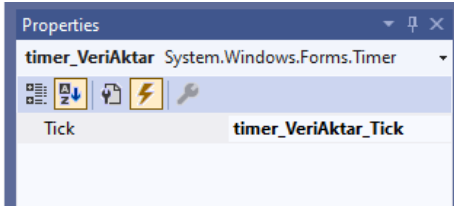
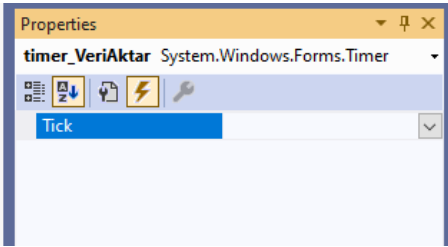
Projenize ToolBox dan bir timer ekleyiniz ,



Propertiesten özelliklerini aşağıdaki gibi yapınız


- Namesini timer_VeriAktar yapınız
- Enabled True (Timer Aktif)
- Interval 250 ms

Timer ile işlemin tekrar etmesini sağlayacağız



Timerin her çalışmada , çalıştıracağı kodu yazalım.

Öncelikle , **Properties** ekranından ,

Events  Simgesine Tıklayınız ve Tick Olayının yanındaki boşluğa çift Tıklayınız

Çift tıkladığınızda **timer_VeriAktor_Tick** olayı tanımlanır ve Aşağıdaki gibi boş bir kod yazma ekranı önünüze gelir

```
1 reference
private void timer_VeriAktor_Tick(object sender, EventArgs e)
{
}
}
```

Bu kod ekranındaki süslü parantezler { } arasına ilgili kodu yazacağız

Aşağıdaki kodları inceleyiniz.

Kod ile Veri Aktarımı


```
private void timer_VeriAktar_Tick(object sender, EventArgs e)
```

```
{  
    try  
    {  
        timer_VeriAktar.Enabled = false;  
        object veri = null;  
  
        veri = hml_Connect_Delta.TagList_GetValue(1);  
        hml_Connect_Siemens.TagList_SetValue(1, veri);  
        hml_Connect_Siemens.Wait(100);  
  
        veri = hml_Connect_Delta.TagList_GetValue(2);  
        hml_Connect_Siemens.TagList_SetValue(2, veri);  
        hml_Connect_Siemens.Wait(100);  
  
        veri = hml_Connect_Delta.TagList_GetValue(3);  
        hml_Connect_Siemens.TagList_SetValue(3, veri);  
        hml_Connect_Siemens.Wait(100);  
  
        veri = hml_Connect_Delta.TagList_GetValue(4);  
        hml_Connect_Siemens.TagList_SetValue(4, veri);  
        hml_Connect_Siemens.Wait(100);  
  
        veri = hml_Connect_Delta.TagList_GetValue(5);  
        hml_Connect_Siemens.TagList_SetValue(5, veri);  
        hml_Connect_Siemens.Wait(100);  
    }  
    catch  
    {  
    }  
    finally  
    {  
        timer_VeriAktar.Enabled = true;  
    }  
}
```

Kod Açıklaması


◆ Veri Okuma

```
veri = hml_Connect_Delta.TagList_GetValue(1);
```

 Delta PLC'den, Tag ID = 1 olan veri okunur , Burdaki 1 , Deltanın 1 nolu Tagıdır, karıştırmayınız

◆ Veri Yazma

```
hml_Connect_Siemens.TagList_SetValue(1, veri);
```

 Okunan veri Siemens PLC'de Tag ID si 1 olan Tag a yazılır , Burdaki 1 , Siemensin 1 nolu Tagıdır, karıştırmayınız

◆ Bekleme (Önerilen)

```
hml_Connect_Siemens.Wait(100);
```

☞ Yazma işleminin stabil olması için kısa bekleme yapılır

◆ Timer Kontrolü

```
timer_VeriAktar.Enabled = false;
```

☞ Aynı anda tekrar tetiklenmesini engeller

```
timer_VeriAktar.Enabled = true;
```

☞ İşlem tamamlandıktan sonra tekrar aktif edilir

⚠ Dikkat Edilmesi Gerekenler

- **Tag ID'ler üzerinden aktarma gerçekleştiği için , HMI_Connect nesnelерinde , aktarılabacak tagların , Tag ID lerini not alınız ve doğru Tag ID leri yazdığınızı kontrol ediniz**
- Veri tipleri (Word / Float vb.) uyumlu olmalıdır
- Delta da Swap ayarları doğru yapılmalıdır

✂ Kısa Özet

- Delta → veri oku
- Siemens → veri yaz
- Timer ile tekrar et

NikCom HMI ile Modbus TCP, karmaşık değil — artık tamamen kontrolünüz altında.