

NIKCOM HMI
KULLANIM KLAVUZU

Giriş ve Amaç :

Nikcom HMI ; Visual Studio ve .Net Teknolojisini kullanarak , fazla kod bilgisi gerektirmeden kolay ve hızlıca kullanıcı arayüzü hazırlamanız için geliştirilmiş bir componentler kümesidir.

Sürükle bırak ve de Componente ait kolay kullanılabilir Özellik Formları İle Visual Studioda rahatlıkla kullanıcı arayüzü tasarlayabileceksiniz ayrıca isterseniz C# ve .Net teknolojisinin esnekliği de elinizin altında olacak.

- Kolay kullanım
- Fazla bilgi gerektirmeden , Profesyonel işlemler yapabilme
- 10 dk da HMI yazılımı hazırlayabilme potansiyeli
- Rahatlıkla PLC İletişimi , Veri Tabanı SQL işlemlerinin yapılabilmesi
- Geniş C# ve .Net Teknolojisini kullanabilme İmkanı

Kimler İçin : Scada Profesyonelleri veya Yeni Başlayanlar , sistem kod bilginiz olmasa bile rahatlıkla Kullanıcı arayüzü geliştirebilmenizi amaçlamaktadır.

Kurulum

Nikom HMI Temelde bir frameworktür.

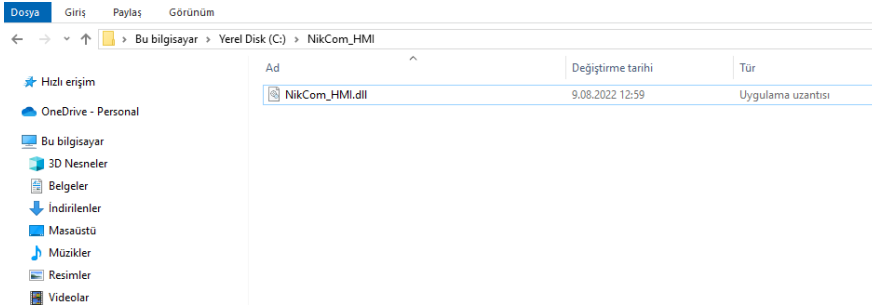
- Nikcom_HMI .dll

Nikom_HMI .dll : Bu bir frameworktür ve kullanacağınız Componentler Bu Framework ün içinde bulunur. Bu framework : yeni sürüm ile birlikte NikCom HMI direkt PLC ile haberleşir

Bizim , kullanıcı ara yüzünü hazırlayacağımız program , NikCom_HMI.dll kütüphanesidir.

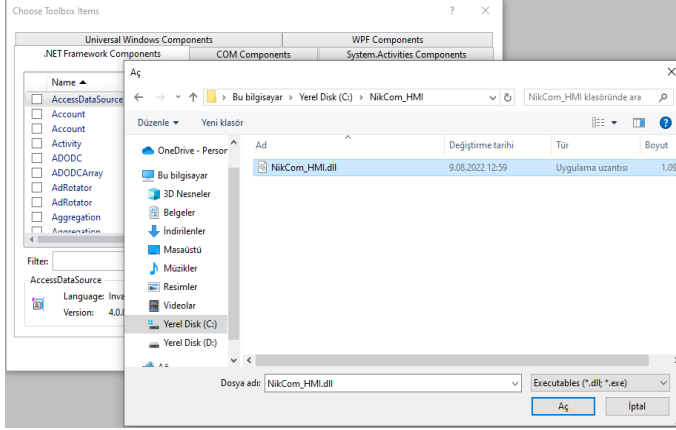
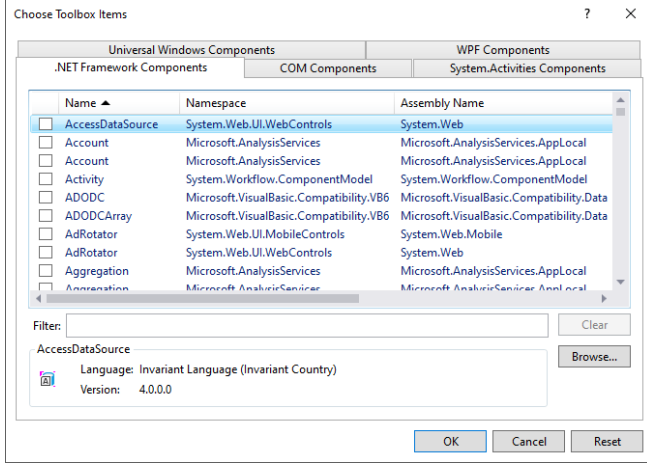
Nikom_HMI .dll Projenize Nasıl Yüklenir.

- Öncelikle C veya D dizininde NikCom_HMI diye bir klasör açınız ve NikCom_HMI.dll dosyasını bu klasörün içerisine atınız.

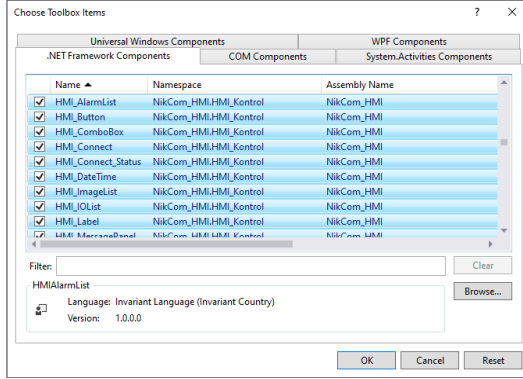


- Visual Studio da ToolBox da Altta General sekmesine sağ tıklayınız ve Choose Items i seçiniz.

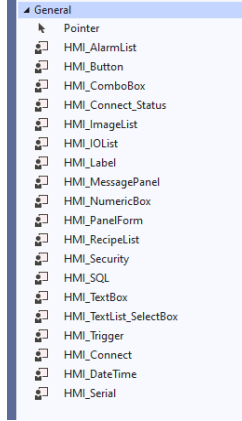
- Daha Sonra Gelen Ekranda .Net FrameWork Components Sekmesinden Browse Butonuna Tıklayınız ve NikCom_HMI.dll Dosyasını Seçiniz



- Yeni Componentler seçili olarak gelecektir.



- OK Deyiniz ve NikCom_HMI Componentleri artık general sekmesi altında görüntülenecektir.



NikCom_HMI.dll	<p>HMI sisteminizi yaparken kullanacağınız Komponentlerin bulunduğu kütüphane , bu kütüphaneyi bilgisayarınıza yükleyip , Visual studio a kurabilirsiniz. Kurulumu önceki konularda anlatıldı. NikCom_HMI kurulumu için herhangi bir lisans ücreti ödemiyorsunuz. Visual studioda rahatlıkla kullanıp , projelerinizi geliştirebilirsiniz.</p> <p>Lisanslama ise programı çalıştırdığınızda yapılır , Full versiyonu çalıştırdığınızda , belirli periyotlarla Sizden Lisans ister. www.ekanik.com dan rahatlıkla full lisans satın alabilirsiniz.</p>
NikCom_HMI Free Versiyon	<p>NikCom_HMI in ücretsiz ve Kısıtlı versiyonudur.</p> <p>Maks Tag Sayısı :50 Maks DB sayısı : 3 Komponentler , kısıtlı HMI _Connect (Kısıtlı , Reçete Sistemi vb. sistemler pasif) HMI _Button HMI _Label HMI _Panel HMI _Connect_Status HMI _Security</p> <p>Bu kütüphane özellikle karmaşık kod bilgisi gerektiren kütüphaneleri (S7.net vb) kullanmadan rahatlıkla PLC haberleşmesi yapıp , projeler geliştirilmesi için geliştirildi.</p>

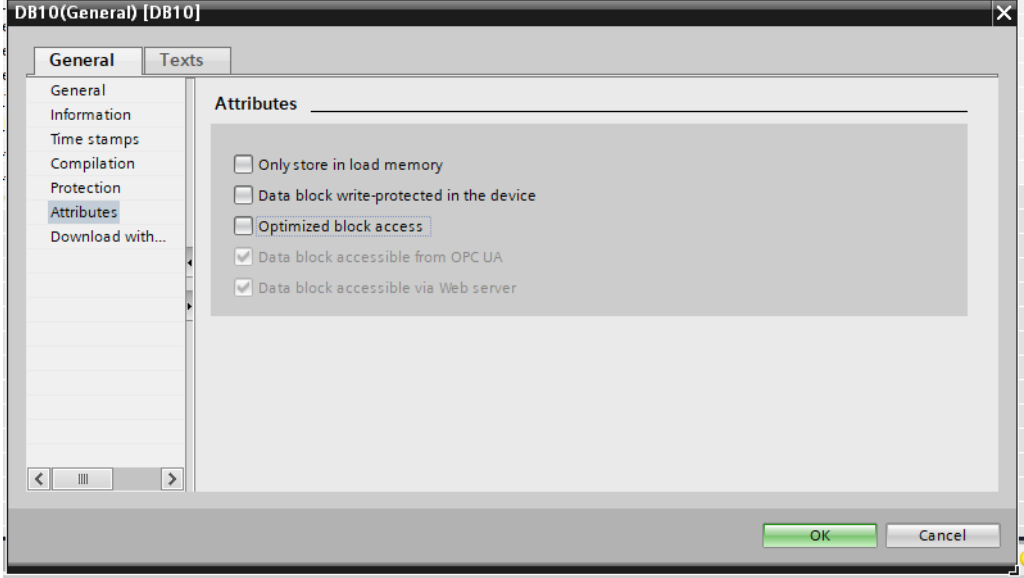
Component Listesi ve Kısa Açıklamalar :

General	
Pointer	
HMI_AlarmList	
HMI_Button	
HMI_ComboBox	
HMI_Connect_Status	
HMI_ImageList	
HMI_IOList	
HMI_Label	
HMI_MessagePanel	
HMI_NumericBox	
HMI_PanelForm	
HMI_RecipeList	
HMI_Security	
HMI_SQL	
HMI_TextBox	
HMI_TextList_SelectBox	
HMI_Trigger	
HMI_Connect	
HMI_DateTime	
HMI_Serial	
	<p>*HMI_Connect : En önemli component dir , PLC iletişimi adreslerin , tagların ve genel anlamda projenin kontrolünün yapıldığı yerdir.</p> <p>*HMI_AlarmList : Alarmların Listelenmesi</p> <p>*HMI_Button : Tıklandığında içerisindeki olay ve komut gruplarının çalıştığı (plc deki bir alanın setlenmesi , veri tabanı işlemleri vb.)</p> <p>*HMI_ComboBox : Bir PLC Tag adresine bağlı olarak seçimin yapılması ve PLC e gönderilmesi , PLC den alınması</p> <p>*HMI_Connect_Status : PLC Bağlantı durumunun görsel olarak sunulması</p> <p>*HMI_ImageList : Resimlerin Bir PLC Tag adresine göre Gösterilmesi</p> <p>*HMI_IOList : Fiziksel IO ların durumlarının gösterilmesi</p> <p>*HMI_Label : Sadece düz yazı ama görünümü rahatça değiştirilebilir ve PLC deki Taga bağlanabilir</p> <p>*HMI_MessagePanel : PLC Tag adresine göre bir metin listesinin gösterilmesi , PLC deki taga göre Renk , flaşör gibi özelliklerin gösterilmesi</p> <p>*HMI_NumericBox : PLC deki bir sayısal adresin değerinin gösterilmesi ve tıklandığında bu adresin değerinin değiştirilebilmesini sağlar</p> <p>*HMI_PanelForm : İçerisine Form alır ve Olay Komutu ile içerisine Form yüklenebilir.</p> <p>HMI_RecipeList : Bir reçete sistemi oluşturmanızı sağlar</p> <p>*HMI_Security : Yetkili Kullanıcı Oluşturmanızı sağlar</p> <p>HMI_SQL : Geniş bir SQL Erişim Sisteminin parçasıdır , Olay Komut ları ile geniş bir kullanım alanı sunar</p> <p>*HMI_TextBox : PLC deki bir string adresin değerinin gösterilmesi ve tıklandığında bu adresin değerinin değiştirilebilmesini sağlar</p> <p>*HMI_TextList_SelectBox : Bir Taga adresine bağlı olarak seçimin yapılması ve PLC e gönderilmesi , PLC den alınması , HMI_Combobox dan farkı açılır kutu değil ayrı bir Formda seçim yapılır</p> <p>*HMI_Trigger : Bir PLC Tag adresine bağlı olarak Olay komut tetiklenmesini sağlar</p> <p>*HMI_DateTime : PLC deki bir tarih adresinin değerinin gösterilmesi ve tıklandığında bu adresin değerinin değiştirilebilmesini sağlar</p> <p>*HMI_Serial : Seri portun okuma ve yazmasını PLC adreslerine bağlı olarak yapar ve Seri portdan alınan veriyi PLC e gönderir veya PLC den aldığı veriyi Seri Porta yazar</p>

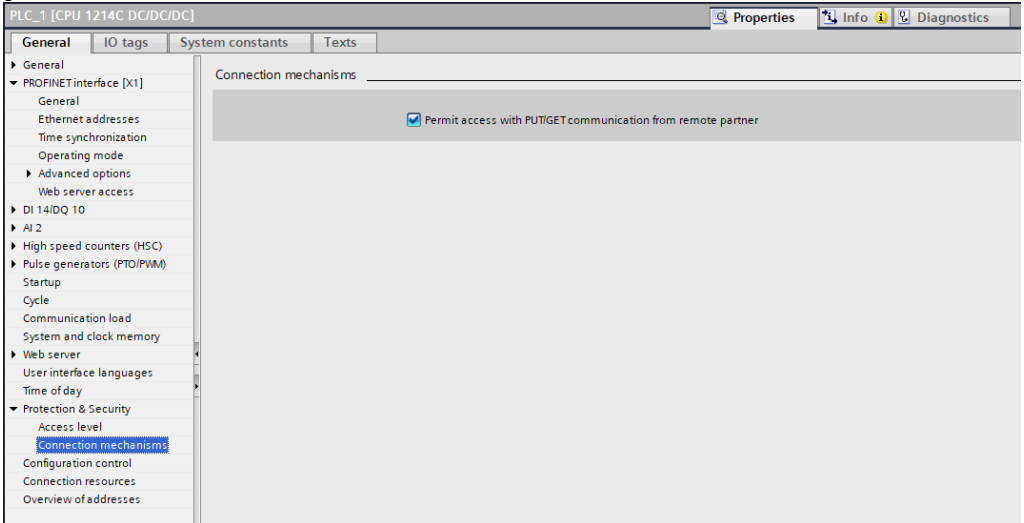
PLC ve DataBlock Ayarları

Öncelikle iletişim için PLC ve DataBlock da 2 ayarı kontrol etmemiz gerekmektedir. Bu ayarlar aşağıdaki gibi değilse iletişim problemleri yaşanır.

NikCom_HMI in haberleşeceği her datablock un Optimized block Access özelliğinin pasif olması gerekir.



PLC nin General->Protection sekmesindeki Permit Access with PUTGET communication özelliğinin seçili olması gerekmektedir.



COMPONENTLER

HMI_Connect : En önemli component dir , PLC iletişimi adreslerin , tagların ve genel anlamda projenin kontrolünün yapıldığı yerdir.

HMI_Connect Tasks

Tag List Özellikleri

Tag Tanımlama (Tag List)

Her Bağlandığında TagListi Yolla

PLC Bilgileri

PLC Tipi: Siemens_s7_1200

Calisma Tipi: Offline

PLC IP Adresi: 192.168.0.10

PLC Rack: 0

PLC Slot: 0

PLC Veri Alma Hızı: 100

İletişim Bilgileri

Bağlantı Tipi: HMI_ile_Direkt_Baglant

IP Kaynağı: Local_Bilgisayardan_AI

İletişim Portu: 5000

Reçete Bilgileri

[Reçete Tanımlama](#)

Dil Listesi

[Dil Listesi Tanımlama](#)

Seçili Dil ID (Dil Seç): 1

Seçili Dil Adı: Türkçe

Security (Yetki)

[Kullanıcı Tanımlama](#)

[Yetki Dil Verileri](#)

Login Zamanı (Sn): 60

Login Dosya Yolu:

Genel Özellikler

Yüklenen Formları Açık Tut

[Dock in Parent Container](#)



Toolbox dan kontrolü form üzerine sürükleyiniz. Soldaki kontrolümüzün simgesidir. Üst sağdaki Ok a tıkladığınızda soldaki resimdeki Menü önünüze gelecektir.

Kısa Açıklamalar

Tag Tanımlama : PLC Taglistin , adreslerin tanımlandığı yerdir.

PLC Tipi : PLC Seçimi buradan yapılır.

Calisma Tipi : Offline : NikCom HMI bir PLC gibi Davranır , PLC Olmadan yazdığınız projeyi test edebilirsiniz. Online : Direkt PLC Bağlantısı aranır.

PLC IP Adresi : PLC nin IP Adresi Buraya Girilir.

PLC Veri Alma Hızı: NikCom HMI ın PLC ile iletişiminde minimum veri alma hızı tanımlanır , çok hızlı işlem yapmanız gerektiğinde bu hızı düşürebilirsiniz. Çok fazla tag sayısı varsada bu sayıyı arttırabilirsiniz.

Reçete Tanımlama : Buradan Makine Ürün Reçetesi tanımlayabilirsiniz. Daha sonra ise HMI_RecipeList i bu reçeteye bağlayabilirsiniz.

Dil Listesi Tanımlama : Sistem 8 adet dil tanımlama imkan tanır daha sonra projenizde bu dilleri aktif bir şekilde değiştirebilirsiniz.

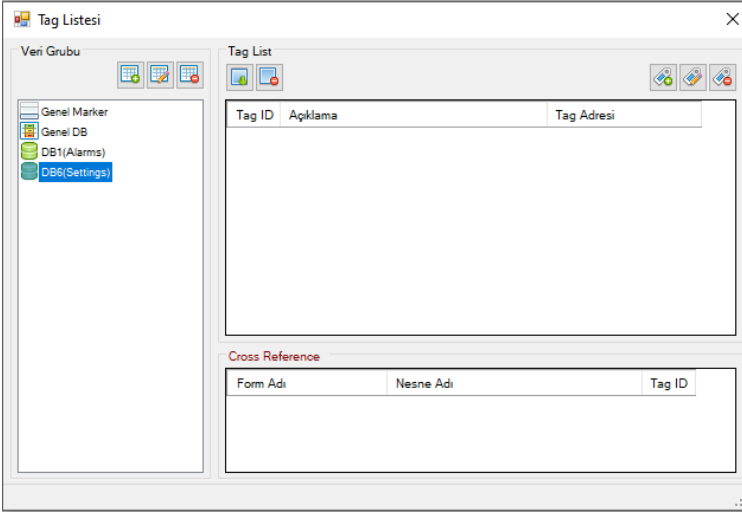
Security : Bu bölümden sisteminize kullanıcı tanımlı yapabilirsiniz.

Login Zamanı : Kullanıcı girişi yapıldığında buraya girilen saniye kadar sistem Login durumunda kalır.

Login Dosya Yolu : Birden fazla Kullanıcı arayüzü programını tek bir merkezden kullanıcı bilgilerini alabilmesi için tanımlandı.

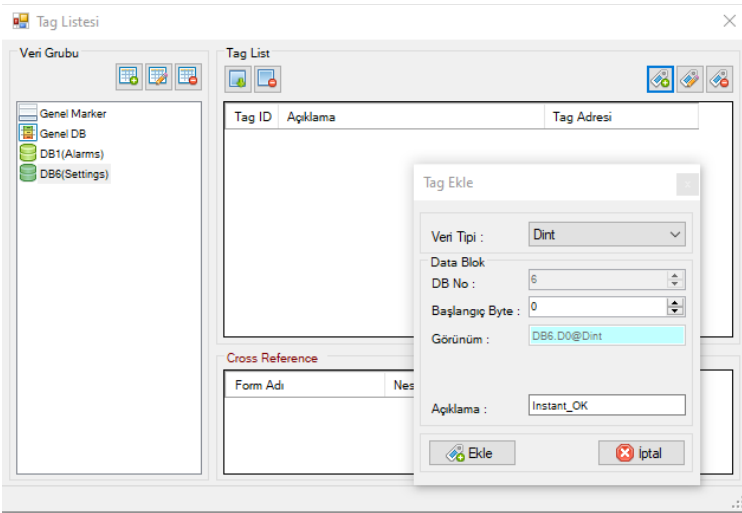
Yüklenen Formları Açık Tut : Daha hızlı form geçişleri istiyorsanız , Formları sürekli açık tutabilirsiniz , dezavantajı fazla bilgisayar kaynağı kullanması.

Tag Tanımlama : PLC Taglistin , adreslerin tanımlandığı yerdir.



Veri Grubu alanında Data Block tanımlaması yaparsınız. Genel DB de Data Block ismi vermeden kayıt yapabilirsiniz. Ayrıca Ekleme Butonuna Tıklayıp Data Block ismi verip de DB tanımlaması yapabilirsiniz.

Tag List alanında tag tanımlaması yapabilirsiniz , isterseniz sol taraftan aşağıdaki gibi manuel tag tanımlaması yapabilirsiniz.



Ayrıca toplu TagList tanımlaması yapabilirsiniz , Bunun için Data Block alanından 3 bölümü aşağıdaki gibi seçiniz ve Copy yapınız . “Name , Data Type , Offset”

Name	Data type	Offset	Start value	Retain	Accessible f...	Write...	Visible in ...	Setpoint
Static								
Instant_LOK	Dint	0.0						
Instant_NOK	Dint	4.0						
Instant_Counter	Dint	8.0						
Total_Counter	Dint	12.0						
Operator_ID	Dint	16.0						
Reference_From_Harling	Bool	20.0						
Buzzer_On_Alarm	Bool	20.1						
Selected_Reference_No	Byte	21.0						
Selected_Reference_String	String[20]	22.0						
Printer_Wait_Time	Time	44.0						
Printer_Pulse_Time	Time	48.0						
Printer_Output_Count	Int	52.0						
FanPresence_1_Sensor_Status	Byte	54.0	16#0					
FanPresence_2_Sensor_Status	Byte	55.0	16#0					
FanPresence_3_Sensor_Status	Byte	56.0	16#0					
FanPresence_4_Sensor_Status	Byte	57.0	16#0					
FanPresence_5_Sensor_Status	Byte	58.0	16#0					

Daha sonra , doğru seçim yaptıysanız , aşağıdaki gibi TagList alanında Kırmızı flaşörü görürsünüz.

Tag Listesi
✕

Veri Grubu

- Genel Marker
- Genel DB
- DB1(Alarms)
- DB6(Settings)

Tag List
✕

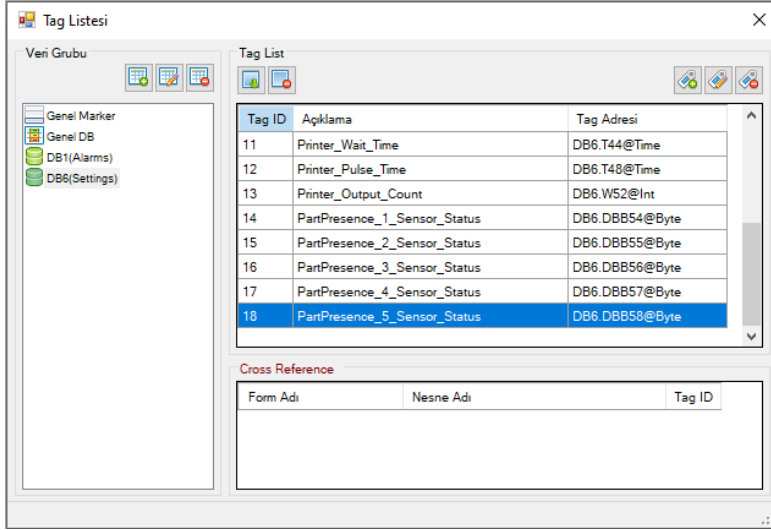
Tag ID	Açıklama	Tag Adresi

Cross Reference

Form Adı	Nesne Adı	Tag ID

Flaşörle yanıp sönen butona tıkladığınızda , Tagların toplu halde TagList e eklendiğini göreceksiniz.

Topluca ekledikten sonra mutlaka kontrol ediniz.



The screenshot displays the 'Tag Listesi' application window. On the left, there is a 'Veri Grubu' (Data Group) sidebar with a tree view containing 'Genel Marker', 'Genel DB', 'DB1(Alarms)', and 'DB6(Settings)'. The main area is titled 'Tag List' and contains a table with the following data:

Tag ID	Açıklama	Tag Adresi
11	Printer_Wait_Time	DB6.T44@Time
12	Printer_Pulse_Time	DB6.T48@Time
13	Printer_Output_Count	DB6.W52@Int
14	PartPresence_1_Sensor_Status	DB6.DBB54@Byte
15	PartPresence_2_Sensor_Status	DB6.DBB55@Byte
16	PartPresence_3_Sensor_Status	DB6.DBB56@Byte
17	PartPresence_4_Sensor_Status	DB6.DBB57@Byte
18	PartPresence_5_Sensor_Status	DB6.DBB58@Byte

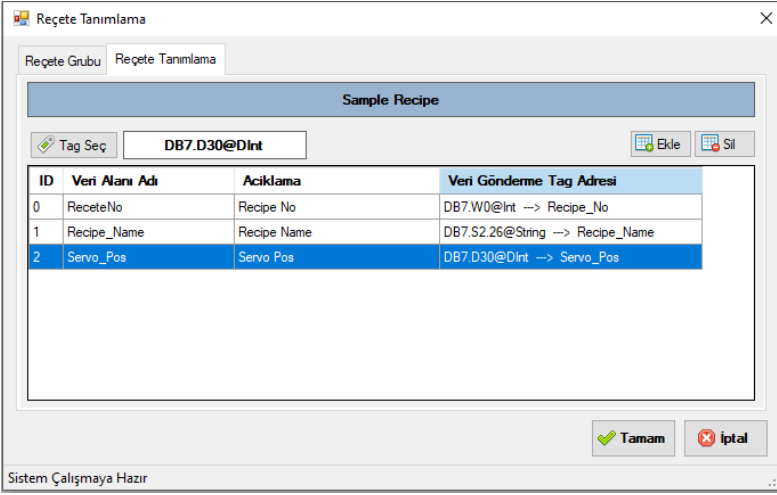
Below the table is a 'Cross Reference' section with a table structure:

Form Adı	Nesne Adı	Tag ID
----------	-----------	--------

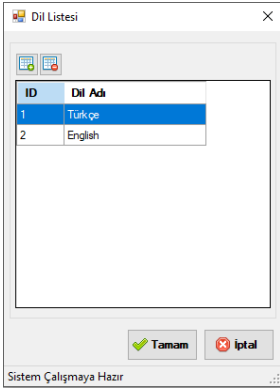
Reçete Tanımlama : Buradan Makine Ürün Reçetesi tanımlayabilirsiniz. Daha sonra ise HMI_RecipeList i bu reçeteye bağlayabilirsiniz. Reçete Grubu ve Reçete Tanımlama alanı diye iki bölüme ayrılır , Reçete grubu ile , birden fazla reçete sistemi oluşturabilirsiniz. Reçete tanımlama ile de bu reçete grubuna ait veri alanlarını tanımlayabilirsiniz.

Aşağıda , Reçete tanımlama ekranı görünmektedir , bu bölümde ilgili reçete gruplarının veri alanları tanımlanır

Aşağıda veri alanları tanımlanmış bir reçete örneğini görüyorsunuz , Bu reçete sistemi HMI_Recipe ve Olay Komutları ile rahatlıkla yönetilebilir ve çok verimli reçeteler tanımlayabilirsiniz. Detayları sonraki konularda anlatacağız.

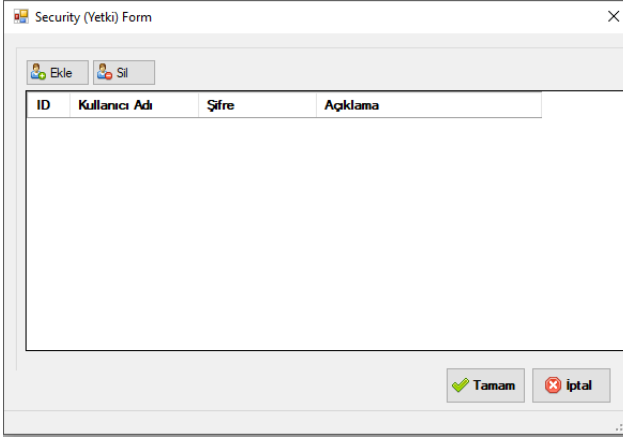


Dil Listesi Tanımlama : Sistem 8 adet dil tanımlama imkanı sunar daha sonra projenizde bu dilleri aktif bir şekilde değiştirebilirsiniz.



Yanda rahatlıkla dil tanımlaması yapabilirsiniz ve Projenizin Dil tercihini Olay Komut Sistemi (HMI_Button ve HMI_Trigger konusunda daha detaylı anlatılacaktır) aracılığı ile rahatlıkla değiştirebilirsiniz.

Security : Bu bölümden sisteminize kullanıcı tanımlı yapabilirsiniz.



ID	Kullanıcı Adı	Şifre	Açıklama
----	---------------	-------	----------

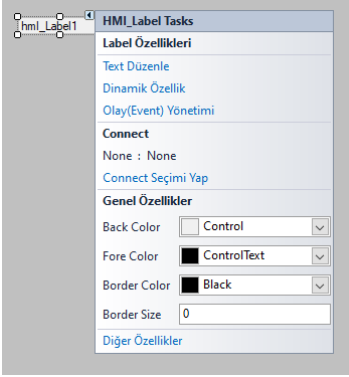
Olay Komut Sistemi (HMI_Button ve HMI_Trigger konusunda daha detaylı anlatılacaktır) sayesinde projenizin dilini rahatlıkla değiştirebilirsiniz.

Yetki Dil Verileri : HMI_Security ile bu tanımlama ekranını projenize ekleyebilirsiniz , yani yetki dil verileri sayesinde bu tanımlama ekranı (Formu) üzerindeki nesnelerin metinlerine dil tanımlaması yapıp seçili dile göre metinlerin değişmesini sağlayabilirsiniz.

HMI_Label : Sadece düz yazı ama görünümü rahatça değiştirilebilir ve PLC deki Taga bağlanabilir



Sağ üstteki kulakçığı tıklayınız ve HMI_Label in Özelliklerine Ulaşınız



Açılan Menüde HMI_Labelin özellikleri bulunmaktadır.

Text Düzenle : HMI_Label de görünecek metin buradan girilmektedir. Burada dil Parametresi ile giriş yapılmaktadır. Yani hangi dilleri tanımladıysanız dillerin metinlerini text düzenleden yapabilirsiniz.

Dinamik Özellik : HMI_Label i Dinamik bir şekilde tag adresine bağlayıp rengini , konum , visibility ve Enable özelliklerini değiştirmenize imkan tanır.

Connect Seçimi : HMI_Label mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Back Color : Arka Plan rengi buradan değiştirilebilir.

Fore Color : Yazı rengi buradan değiştirilebilir.

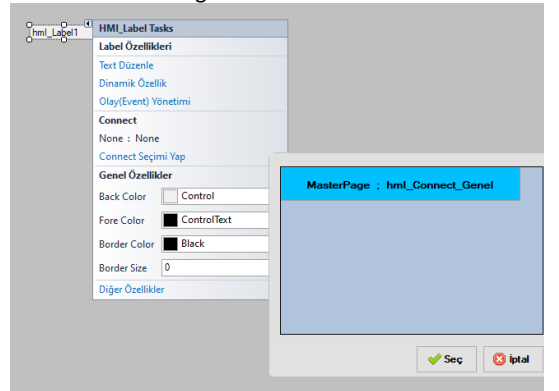
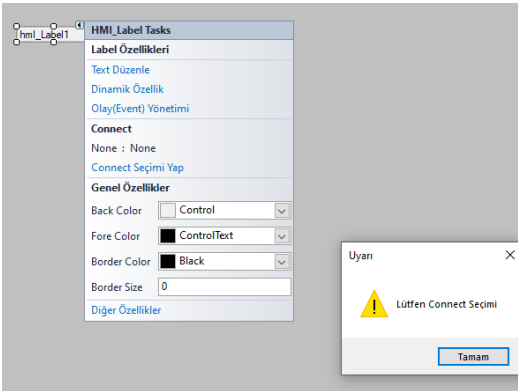
Border Color - Border Size : Çerçeve rengi ve çerçeve kalınlığı buradan belirlenebilir.

Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir.

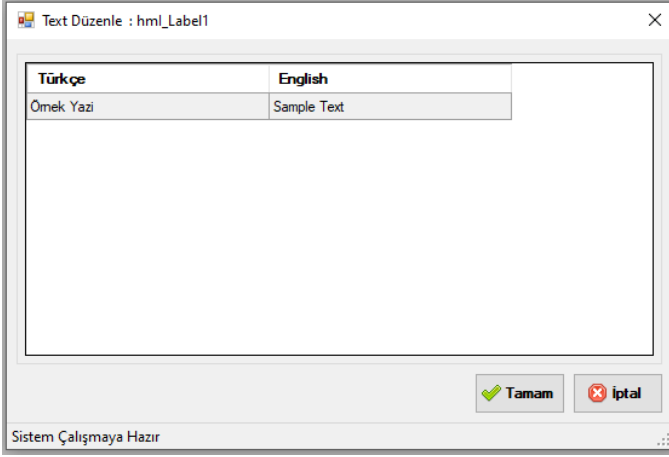
Connect Seçimi : HMI_Label mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Eğer connect seçimi yapılmazsa sistem "Lütfen Connect Seçiniz" diye uyarır.

Connect Seçimi Yap özelliğine tıkladığınızda Connect Seçim Ekranı Gelir , Ekranda gördüğünüz gibi Form adı ve Connect adı görünür.



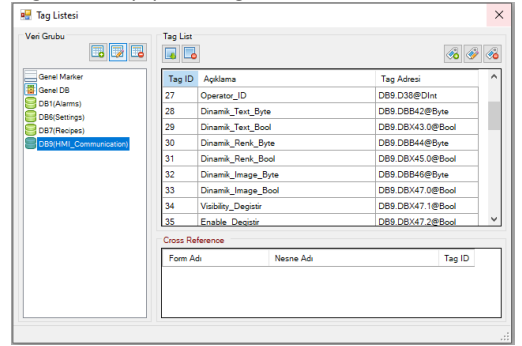
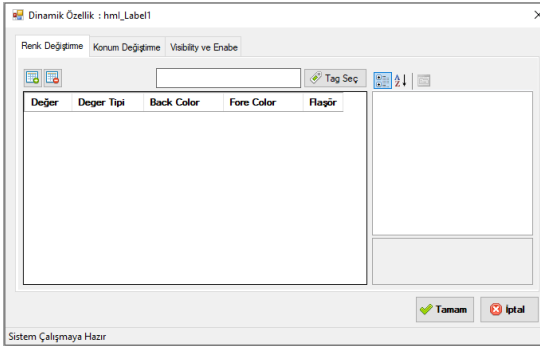
Text Düzenle : HMI_Label de görünecek metin buradan girilmektedir. Burada dil Parametresi ile giriş yapılmaktadır. Yani hangi dilleri tanımladıysanız dillerin metinlerini text düzenleden yapabilirsiniz.



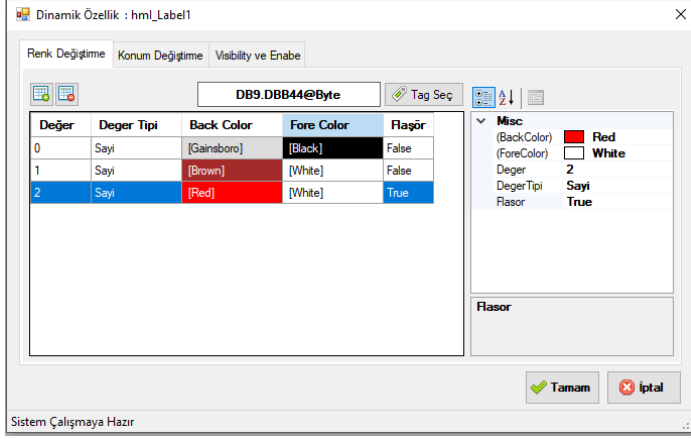
Dinamik Özellik :

HMI_Label i Dinamik bir şekilde tag adresine bağlayıp rengini , konum , visibility ve Enable özelliklerini değiştirmenize imkan tanır.

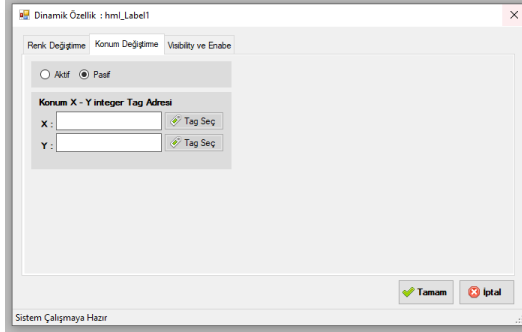
HMI_Connect de TagList de HMI_Communication DataBlock unu ekledik ve aşağıdaki gibide Tag ları ekledik. Bu taglar sayesinde dinamik özelliklere tag ataması yapabileceğiz.



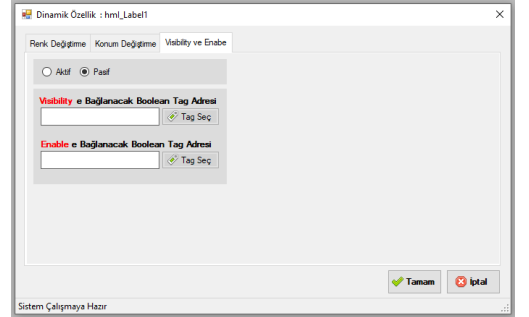
Aşağıdaki gibi bir tagın değerine bağlı olarak dinamik bir şekilde arkaplan ve yazı rengini değiştirebilirsiniz. Ayrıca flaşör bağlayabilirsiniz.

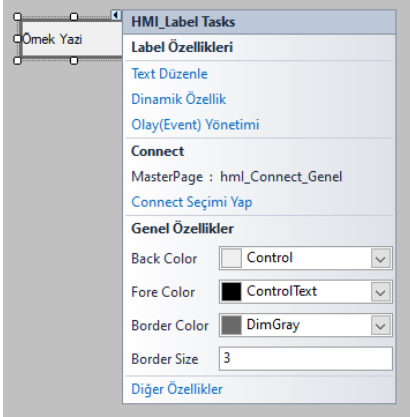


Bir taga bağlayıp dinamik bir şekilde konumunu değiştirebilirsiniz



Bir taga bağlayıp dinamik bir şekilde visibility ve enable değerini değiştirebilirsiniz

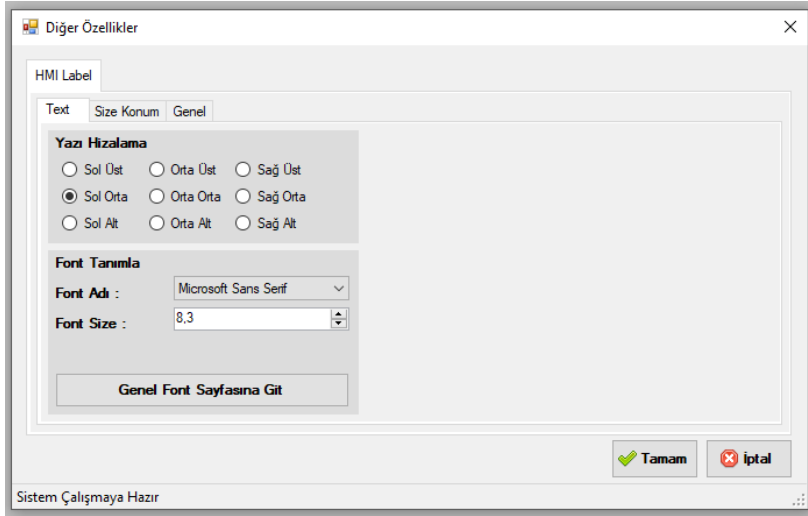




Yandaki gibi Border Sizesini (çerçeve kalınlığı) ve Border Color u , çerçeve rengini değiştirebilirsiniz.

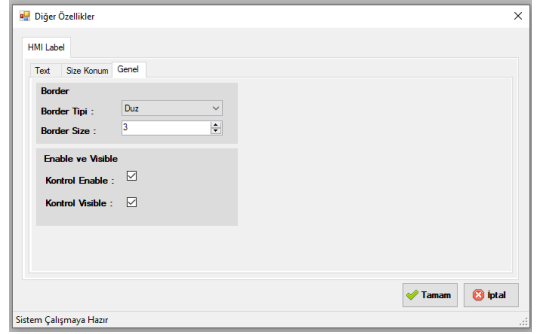
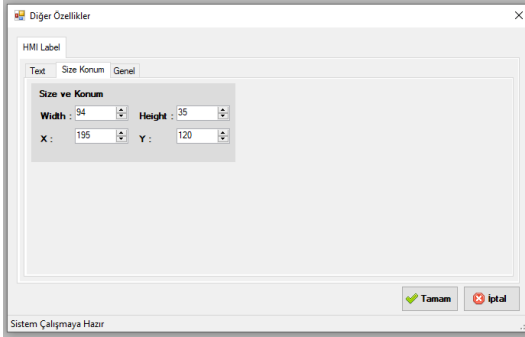
Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir.

Text özelliği ile yazının konumunu belirleyip fontunu değiştirebilirsiniz.

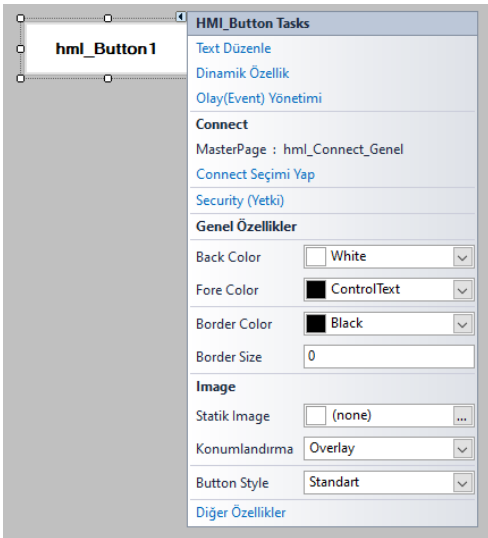


Size ve konum ile hassas Size ve konumlama yapabilirsiniz

Genel de ise Hassas Border girişi yapıp , Static bir şekilde Enable ve Visiblesini değiştirebilirsiniz.



HMI_Button : Tıklandığında içerisindeki olay ve komut gruplarının çalıştığı (plc deki bir alanın setlenmesi , veri tabanı işlemleri vb.) temel componentlerden biridir.



Text Düzenle : Button üzerindeki metnin statik olarak yazılmasını sağlar , ayrıca dinamik özelliklerden de metnin bir tag adresine göre değişmesini sağlayabilirsiniz.

Dinamik Özellik : Bir Tag adresine göre , button üzerindeki metin , button üzerindeki resim , buttonun rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz.

Olay (Event) Yönetimi : Button tıklandığında buttonun hangi işleri yapacağı buradan tanımlanır.

Connect Seçimi : HMI_Button mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Security : Buttonu kullanabilecek kullanıcılar burdan seçilmelidir.

Statik İmage : İstenirse butona sabit resim eklenebilir. Konumlandırma : Eklenecek resmin button üzerindeki konumu belirlenir.

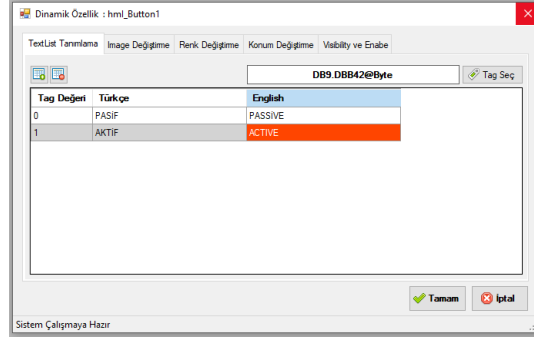
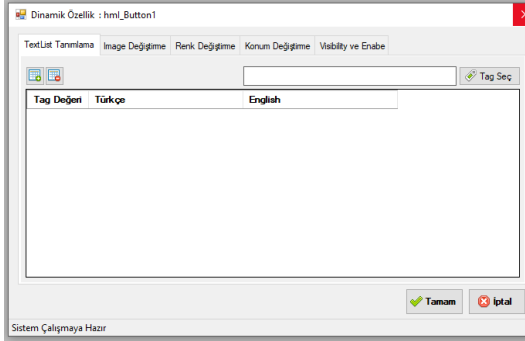
Button Style : Buttonun görüntü stili belirlenebilir.

Connect Seçimi : HMI_Button mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Dinamik Özellik :

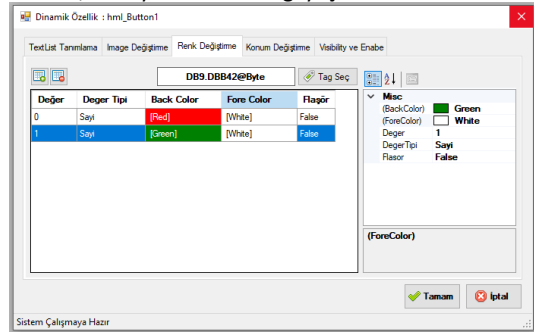
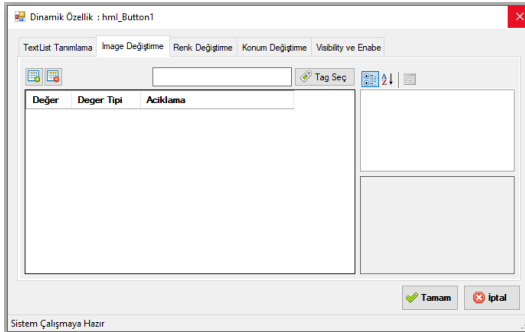
Bir Tag adresine göre , Button üzerindeki metni değiştirebilirsiniz

Örnekte tag adresine göre Metin Pasif veya Aktif yapıyor.

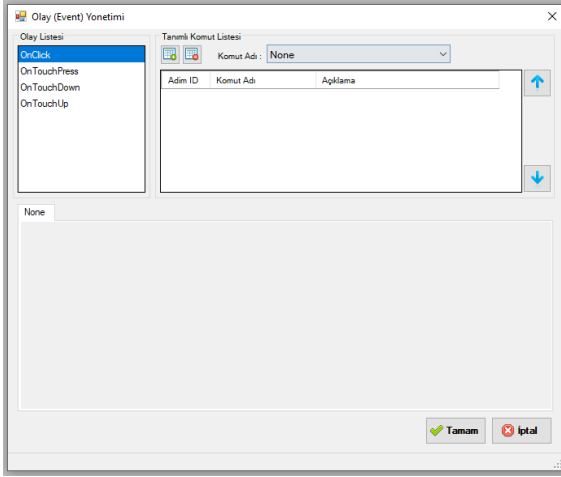


Bir tag adresinin değerine göre button üzerindeki image yi dinamik bir şekilde değiştirebilirsiniz

Bir tag adresinin değerine göre button üzerindeki renkleri dinamik bir şekilde değiştirebilirsiniz , burada , textList deki tag adresi ile buradaki tag adresi aynı , Yani buttonda pasif yazarsa rengi kırmızı , aktif yazarsa ise rengi yeşil olmaktadır



Olay (Event) Yönetimi : Button tıklandığında buttonun hangi işleri yapacağı buradan tanımlanır.



Olay Listesi , komutların hangi durumlarda çalışacağını belirler.

OnClick : Tıklandığında bir kez

OnToucPress : Basılı tutulduğunda – süreklilik gereken işlemler , örneğin jog işlemleri

OnTouchDown: Button Tıklandığında basılı iken bir kez çalışır.

OnTouchUp: Button Tıklanıp bırakıldığında bir kez

Tanımlı Komut Listesi

Komut Adından , komut seçiniz ve Ekle ye tıklayınız , Komut eklendiğinde , aşağıdaki satırdan seçerseniz aşağıda o komutla ilgili parametreler görünecektir.



Set Bit : Olay çalıştığında Seçilen Tagın değeri Setlenir

Reset Bit : Seçili tagın değeri resetlenir

Set Reset Bit : Tagın değeri önce setlenir daha sonra resetlenir

Set Bit When Touch Press: Tagın değeri Button basılı olduğunda settedir , bırakıldığında resetlenir , özellikle jog işlemlerinde kullanılır , bu komut **OnTouchDown** olayında kullanılır.

Toggle Bit : Tagın değeri set ise resetler , reset ise setler

Dil Değiştir : Projenizi , parametrede tanımladığınız dile çevirir

Login : Kullanıcı giriş ekranı gelir ve kullanıcı girişi yaparsınız.

Logout : Logout yapılır.

Veri Gönder : Belirlediğiniz değeri tag adresine yollar

Form Aç : Seçtiğiniz Formu açar ve seçtiğiniz HMI_Panel nesnesi içerisinde görüntüler. İsterseniz formu ayrı pencerede modal olarakta görüntüleyebilir.

Bekleme : Girdiğiniz komutların arasına bekleme eklemek isterseniz bu komutu kullanınız , ard arda bit değer değişimi veya veri gönderme yapacaksanız mutlaka bekleme veriniz.

Mesaj Görüntüle : Ekranda mesaj görüntüleyebilirsiniz

Makro Çalıştır : Kendi fonksiyonlarınızı yazıp , bu komutlar arasında çalıştırabilirsiniz , Web Servis ile veri gönderme vs. Ayrıca Makro kullanmadan design aşamasında butonu çift tıklayıp altında C# kodları yazabilirsiniz

İf Koşul : İf komutunu kullanarak komutlarda farklı satırlara gidebilirsiniz , hatta Adıma Git komutu ile birlikte While benzeri döngü kurabilirsiniz.

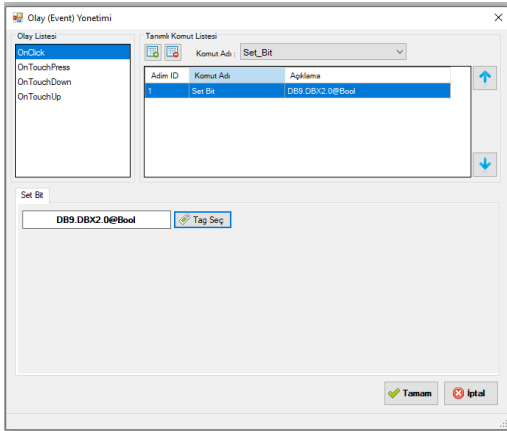
Adıma Git : Belirlediğiniz Adıma geçiş yapar , Adım önceki veya sonraki adımda olabilir , İf_Kosul ile birlikte kullanıldığında While benzeri döngü kurulabilir.

Adım Sonu : Bu komut ile Adım sonuna Gidersiniz

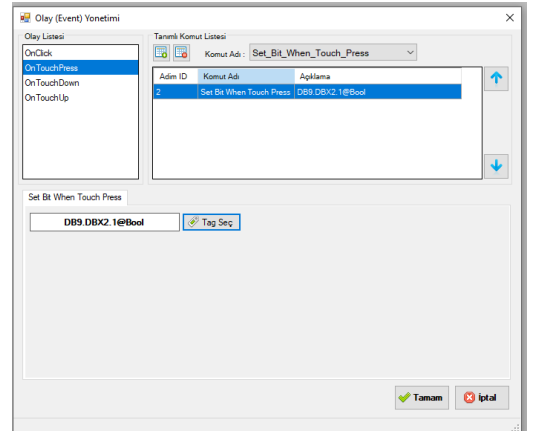
Reçete – Kaydet – Yeni - Sil : Reçete işlemleri bu komut ile yapılır

SQL Sorgu Yolla : Geniş bir SQL Erişim sistemini bu komut ve parametreleri ile kullanabilirsiniz.

Aşağıda OnClik Olayında Set Bit Komutu Örneği Görüntülenmektedir.

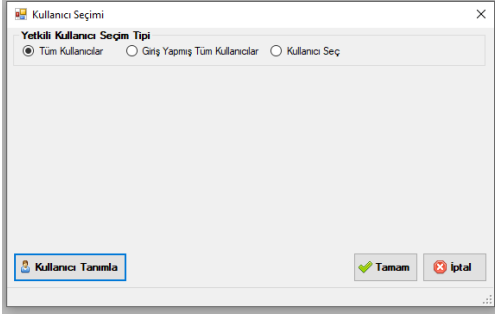


Aşağıda ise OnTouchPress Olayında Set Bit **Set Bit When Touch Press Komutu** Örneği Görüntülenmektedir.

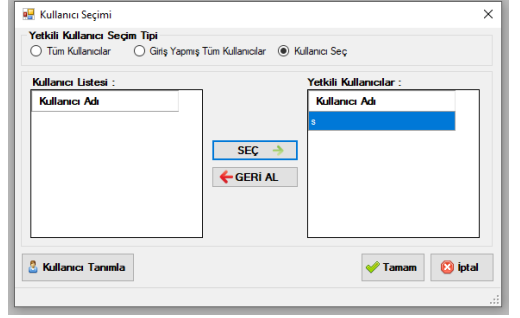


Security : Butonu kullanabilecek kullanıcılar burdan seçilmelidir. Olay ve Komutlar sadece yetkili kullanıcı Butonu kullandığında çalışır. İsterseniz burada kullanıcı tanımlayabilirsiniz , isterseniz HMI_Connect nesnesinde de kullanıcı tanımlayabilirsiniz. Buradaki kullanıcı listesi HMI_Connect nesnesindeki kullanıcı yapısıdır.

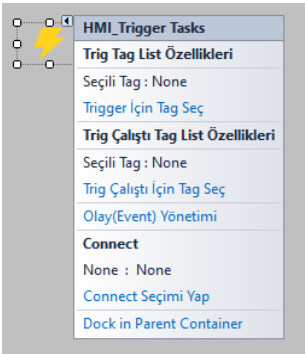
Varsayılan olarak tüm kullanıcılar Buttondaki Olay ve komutları çalıştırabilir.



İsterseniz kullanıcı listesinden seçim yaparak , kullanıcıya yetki verebilirsiniz.



HMI_Trigger : Bir PLC Tag adresine bağlı olarak Olay komut tetiklenmesini sağlar

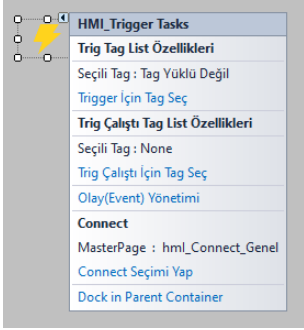


Trigger İçin Tag Seç : Seçilen Tagın yükselen kenarında Olay Komut Sistemi Çalışır , Trig Çalıştı sinyalinin PLC e yollar.

Trig Çalıştı İçin Tag Seç : Trig Sinyali Geldikten Sonra PLC e geri bildirim yapmak için seçilen tagdır. Bu adresten PLC e veri geldikten sonra PLC tarafında bu adresin false yapılmasını unutmayınız , ideal i veri geldikten 2-3sn sonra false yapılmasıdır.

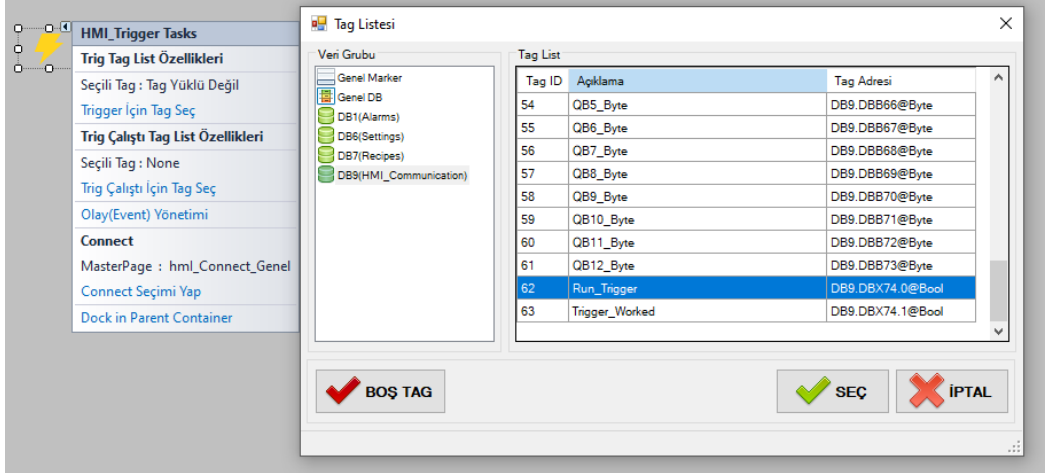
Olay (Event) Yönetimi : Trig sinyali geldiğinde hangi işlerin yapılacağı buradan tanımlanır.

Connect Seçimi : HMI_Trigger mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

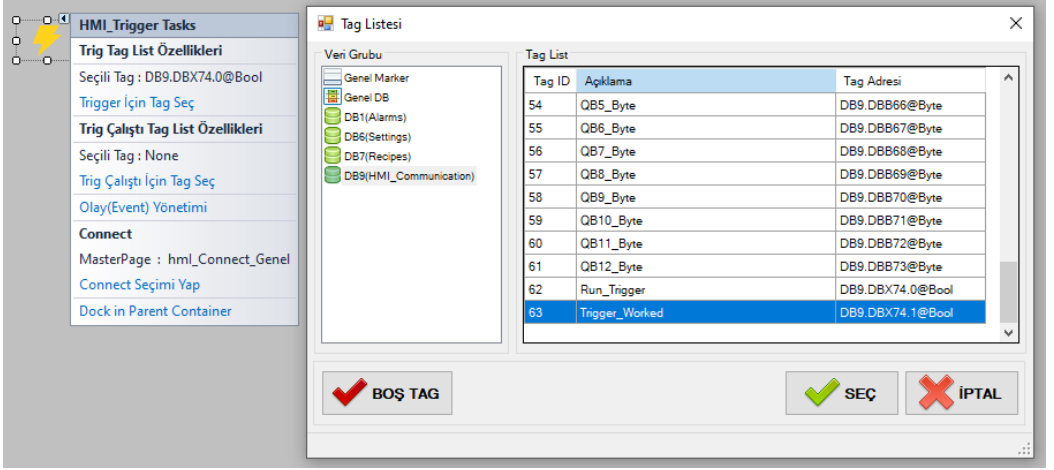


Connect Seçimi : HMI_Trigger mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

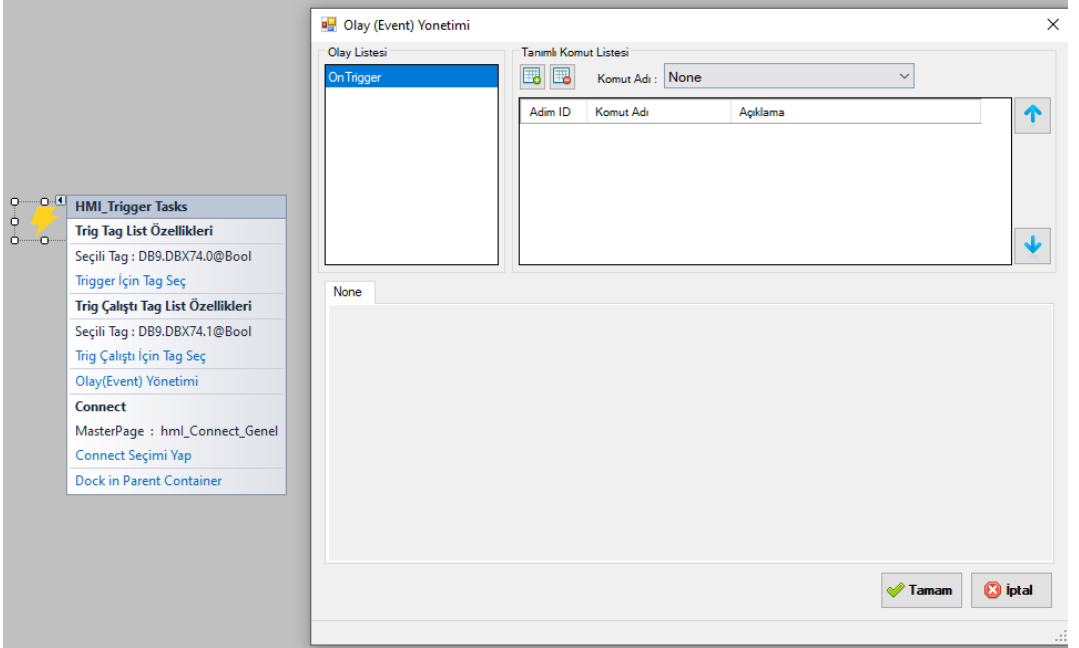
Aşağıda yapıldığı gibi , Triggeri çalıştıracak Bool Tipindeki Tagı seçebilirsiniz , Bu tag değeri true olursa , tag değeri sistem tarafından okunduğunda bu değerin yükselen kenarı tetikleyici olur (yükselen kenar : PLC Terimidir , yani birkez çalışır ve tekrar çalışması için tag değerinin false olması beklenir)



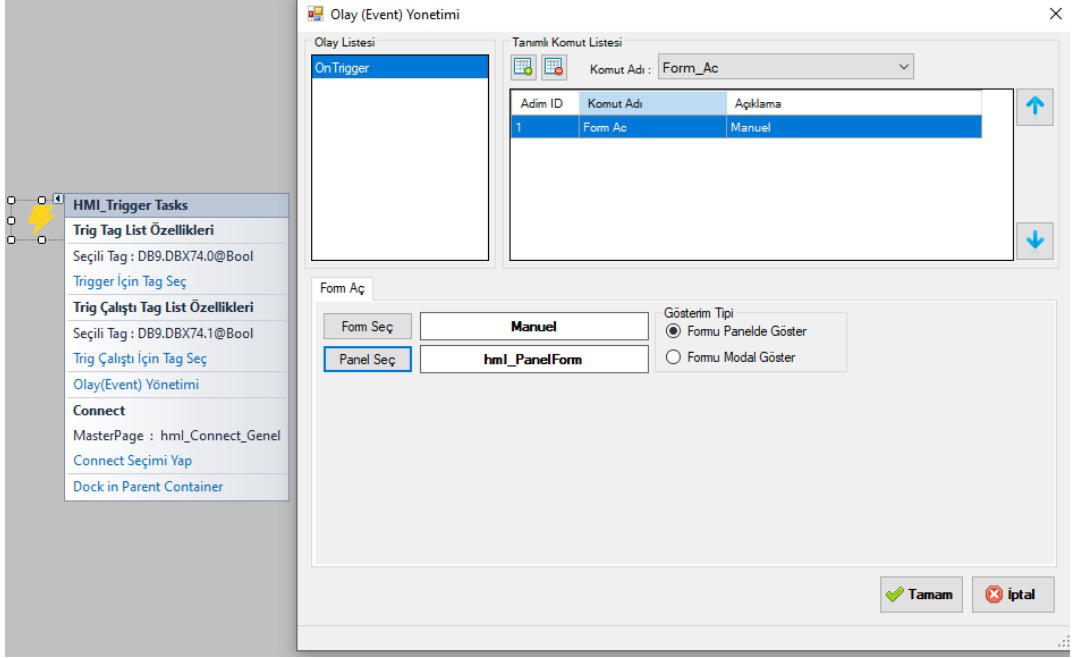
Aşağıda ise Trig Çalıştı sinyalinin PLC e gönderilmesi için Tag Seçimi yapılıyor. Bu Tag sayesinde Trig sinyali geldikten sonra PLC e geri bildirim yapılmaktadır.



Trig Geldiğinde Çalışacak Olay ve Komut Aşağıdaki Gibidir. HMI_Button benzeri bir yapıya sahiptir , rahatlıkla kullanabilirsiniz , kullanım için HMI_Button u inceleyebilirsiniz , Burdaki Olay OnTrigger Olayıdır.



Aşağıda, Formun Açılıp Panel İçerisinde Gösterilmesine Örnek Komut Verilmiştir. Açılacak Form ve İçerisinde Gösterileceği Panel Seçilmelidir.



Olay (Event) Yonetimi

Olay Listesi

On Trigger

Tanımlı Komut Listesi

Komut Adı : Form_Ac

Adım ID	Komut Adı	Açıklama
1	Form Ac	Manuel

Form Aç

Form Seç Manuel

Panel Seç hml_PanelForm

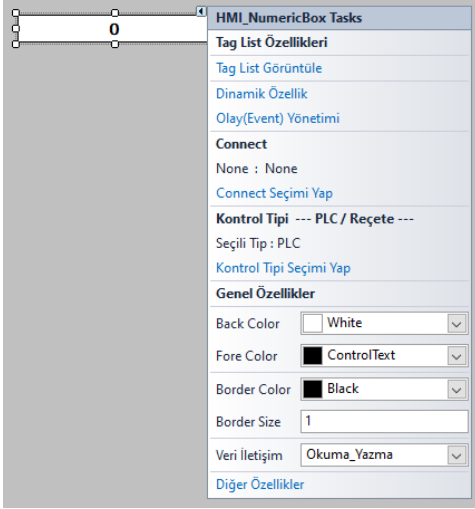
Gösterim Tipi

Formu Panelde Göster

Formu Modal Göster

Tamam İptal

HMI_NumericBox : PLC deki bir sayısal adresin değerinin gösterilmesi ve tıkladığında bu adresin değerinin değiştirilebilmesini sağlar , ayrıca reçete sisteminde de kullanılabilir.



Tag List Görüntüle : HMI_NumericBox un kontrol edeceği Tag adresi buradan seçilir.

Dinamik Özellik : Bir Tag adresine göre , NumericBoxun rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz.

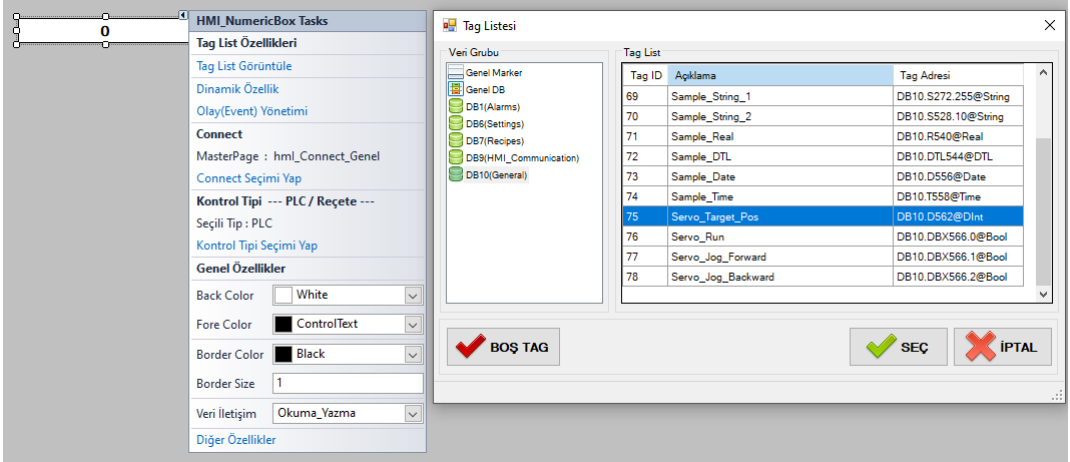
Connect Seçimi : HMI_NumericBox mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Kontrol Tipi Seçimi : HMI_NumericBox un PLC için mi Reçete içinmi kullanılacağını belirler.

Veri İletişimi : Okuma_Yazma veya Sadece Okuma seçilebilir. Sadece Okuma seçilirse PLC Değeri görüntülenir ama HMI_NumericBox üzerinden değiştirilemez.

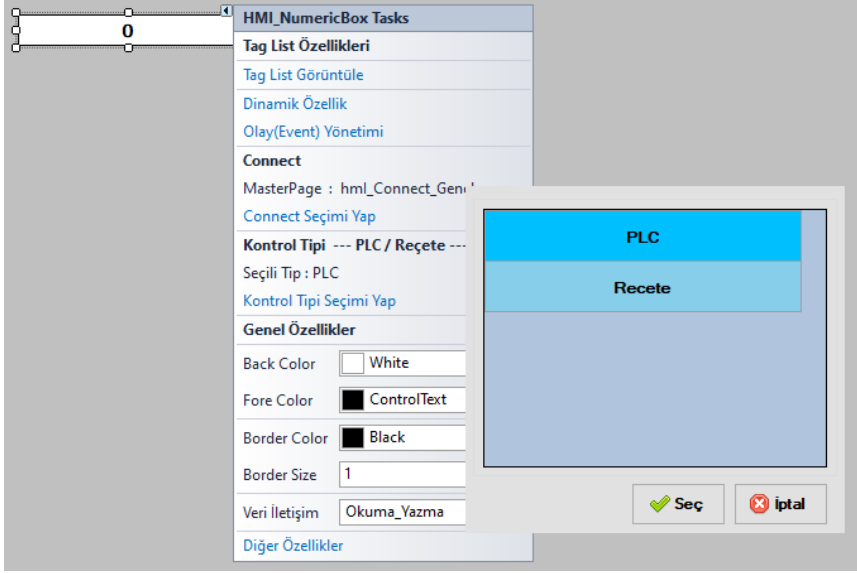
Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir.

Tag List Görüntüle ile HMI_NumericBox un kontrol edeceği Tag adresi seçilir. Projemize DB10 General DataBlock unu ekledik ve DB10 içindeki Servo_Target_Pos Dint adresini NumericBox a Bağladık



Tag Adresini seçtikten sonra yandaki gibi NumericBox üzerinde Tag Adresi görünmektedir

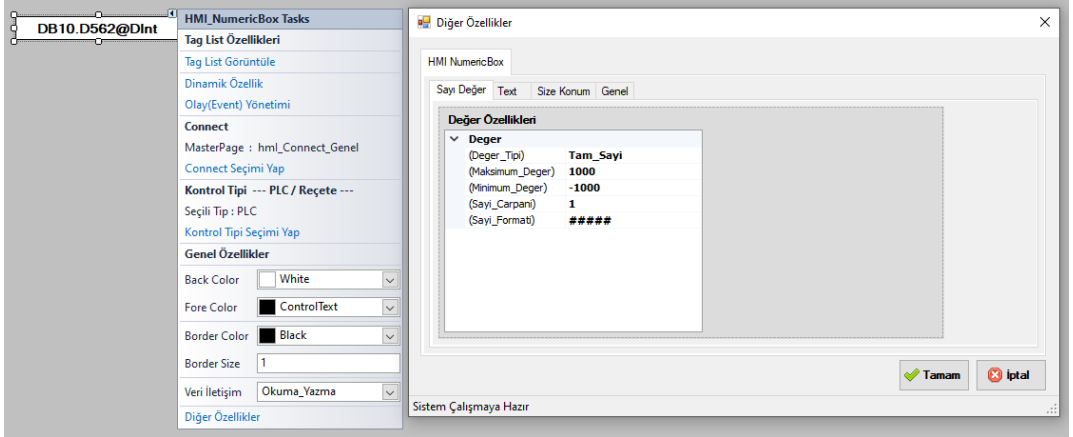
Kontrol Tip Seçimi ile , NumericBox un Veri Bağının PLC Tag Adresimi yoksa Reçete sistemimi olduğunu belirliyoruz. Projenizde Reçete sistemi kurmak istiyorsanız , HMI_Connect üzerinde reçete tanımlayıp , NumericBox ile verileri kontrol edebilirsiniz.



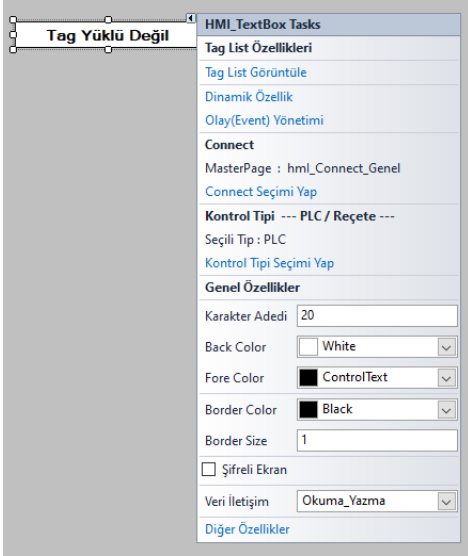
Diğer Özellikler:

NumericBox un sayı değerinin görünümü ile ilgili bilgileri buradan kontrol edebilirsiniz ayrıca Yazı hizalama font size gibi özellikler buradan değiştirilebilir.

- **Değer Tipi** : Bağlı olduğu veri merkezinin sayısal tipidir. Tam sayı veya virgüllü sayı
- **Maksimum değer** : NumericBox un maksimum değeri
- **Minimum Değer** : NumericBox un minimum değeri
- **Sayı Çarpanı** : Çok önemli bir özelliktir , özellikle servolarda pozisyon bilgisi gönderilirken ihtiyaç duyulabilir. NumericBox a girilen değer sayı çarpanı ile çarpılır ve PLC e yolların ayrıca PLC den gelen değer de sayı çarpanına bölünerek NumericBox da gösterilir. Örn NumericBox a 100 değeri ve sayı çarpanı 10 diyelim PLC e 100x10 yani 1000 değeri gönderilir , benzer şekilde PLC deki adrese 1200 değeri girilirse bu sayı 10 a bölünür ve NumericBox da 120 değeri gösterilir.
- **Sayı Format** : Sayının gösterileceği format belirlenir , örneğin virgülden sonraki 1 hanenin görünmesini istiyorsunuz ve virgülden önce de 5 hanenin görünmesini istiyorsunuz , Format : #####,# yapılır Maksimum ve minimum değeri ve Değer_Tipi ile uyumlu olmalıdır.



HMI_TextBox : PLC deki bir string adresin deęerinin gsterilmesi ve tıklanđında bu adresin deęerinin deęiřtirilebilmesini saęlar , ayrıca reęete sisteminde de kullanılabilir.



Tag List Grntle : HMI_TextBox un kontrol edeceęi Tag adresi buradan seęilir.

Dinamik zellik : Bir Tag adresine gre , HMI_TextBox rengini , konumunu , visibility ve enable zelliklerini deęiřtirebilirsiniz.

Connect Seęimi : HMI_TextBox mutlaka bir HMI_Connect nesnesine baęlanmalıdır.

Kontrol Tipi Seęimi : HMI_TextBox un PLC iin mi Reęete iinmi kullanılacaęını belirler.

Karakter Adedi : Girilebilecek maksimum karakter sayısıdır.

Veri İletişimi : Okuma_Yazma veya Sadece Okuma seęilebilir. Sadece Okuma seęilirse PLC Deęeri grntlenir ama HMI_TextBox zerinden deęiřtirilemez.

Şifreli Ekran : Girilen karakterlerin gizli olması saęlanır

Dięer zellikler : Yazı hizalama font size gibi zellikler buradan deęiřtirilebilir.

Connect Seęimi : HMI_TextBox mutlaka bir HMI_Connect nesnesine baęlanmalıdır.

Dinamik zellik : Bir Tag adresine gre , HMI_TextBox rengini , konumunu , visibility ve enable zelliklerini deęiřtirebilirsiniz. HMI_NumericBox ile benzer yapıya sahiptir , bundan dolayı tekrar anlatılmayacaktır.

Kontrol Tipi Seęimi : HMI_TextBox un PLC iin mi Reęete iinmi kullanılacaęını belirler. HMI_NumericBox ile benzer yapıya sahiptir , bundan dolayı tekrar anlatılmayacaktır.

Dięer zellikler : Yazı hizalama font size gibi zellikler buradan deęiřtirilebilir. HMI_Label ile benzer yapıya sahiptir , bundan dolayı tekrar anlatılmayacaktır.

HMI_ComboBox : Bir PLC Tag adresine bağılı olarak seçimin yapılması ve PLC e gönderilmesi , PLC den alınması



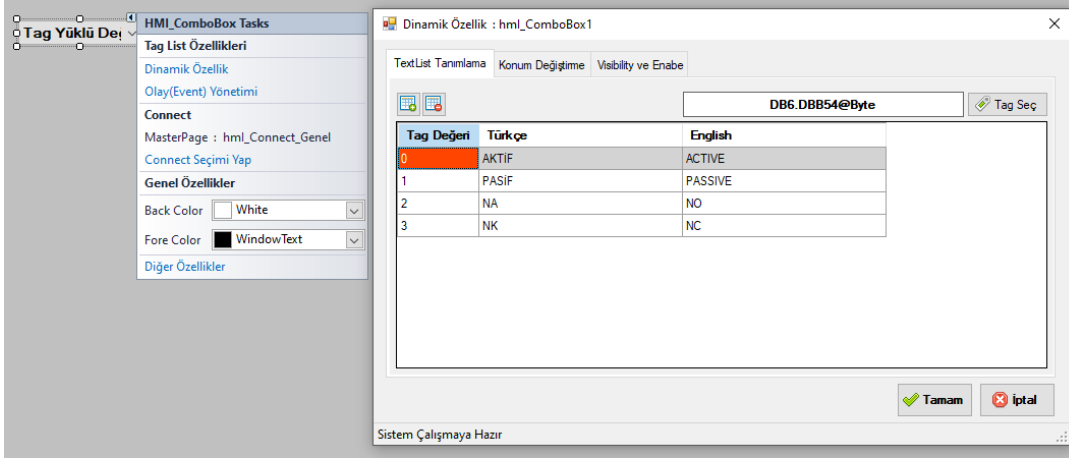
	<p>Dinamik Özellik : Bir Tag adresine göre , HMI_ComboBox un rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_ComboBox un listesinde görüntülenecek seçeneklerde TextList tanımlama bölümünde yazdığınız satırlardır.</p> <p>Connect Seçimi : HMI_ComboBox mutlaka bir HMI_Connect nesnesine bağlanmalıdır.</p> <p>Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir.</p>
--	---

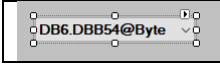
Connect Seçimi : HMI_ComboBox mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Dinamik Özellik : Bir Tag adresine göre , HMI_ComboBox un rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_ComboBox un listesinde görüntülenecek seçeneklerde TextList tanımlama bölümünde yazdığınız satırlardır.

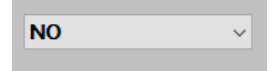
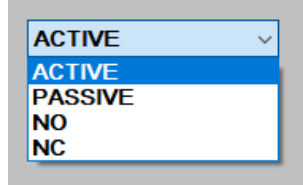
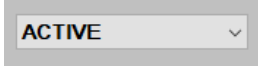
TextList tanımlama bölümüne yazdığınız satırlar , HMI_Combobox un seçenekleridir. Seçilen tag adresinin sayısal değerine göre o seçenek gösterilir , siz HMI_ComboBox tan bir seçenek seçerseniz , O seçeneğin sayısal değeri seçtiğiniz adrese gönderilir

Aşağıdaki Örnekte 4 Seçenek girilmiştir. Ve bu seçenekler [DB6.DBB54@Byte](#) adresine bağlanmıştır. Tag Değeri karşılığındaki metin HMI_Combobox da gösterilecek veya seçilebilecektir.



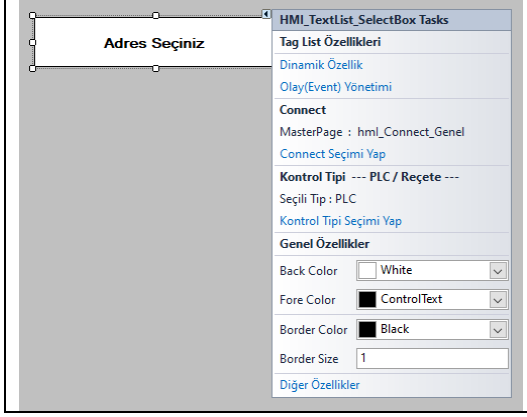
 TextList tanımlamada seçtiğiniz tag adresi, design modda HMI_Combobox un üzerinde görüntülenmektedir.

Aşağıda HMI_Combobox dan Seçim Yapılmıştır , Active Seçimi NO Olarak Değiştirilmiştir , PLC e ise Bu Seçeneklerin Tag Değeri , yani sayısal değer gönderilmiştir.



Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir. HMI_Label ile benzer yapıya sahiptir , bundan dolayı tekrar anlatılmayacaktır.

HMI_TextList_SelectBox : Bir Taga adresine bağlı olarak seçimin yapılması ve PLC e gönderilmesi , PLC den alınması , HMI_Combobox dan farkı açılır kutu değil ayrı bir Formda seçim yapılır.



Dinamik Özellik : Bir Tag adresine göre , HMI_TextList_SelectBox un rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_TextList_SelectBox un listesinde görüntülenecek seçeneklerde TextList tanımlama bölümünde yazdığınız satırlardır.

Connect Seçimi : HMI_TextList_SelectBox mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

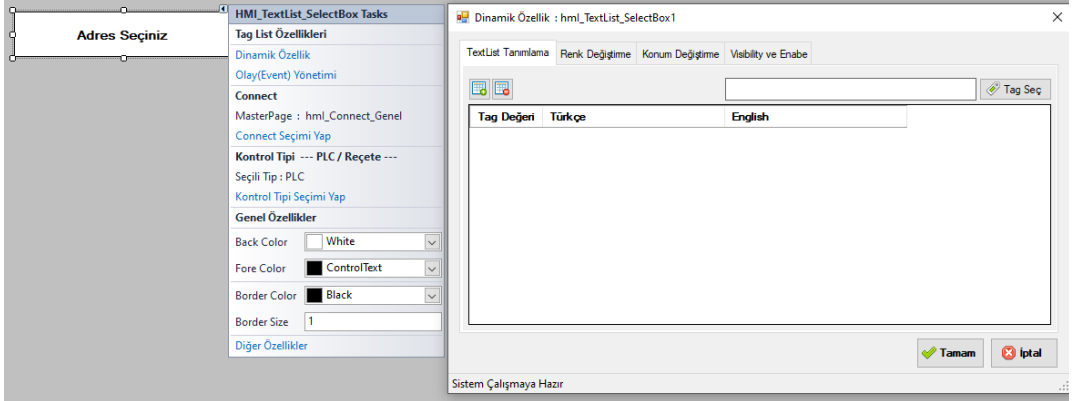
Kontrol Tipi Seçimi : HMI_TextList_SelectBox un PLC için mi Reçete içinmi kullanılacağını belirler.

Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değiştirilebilir.

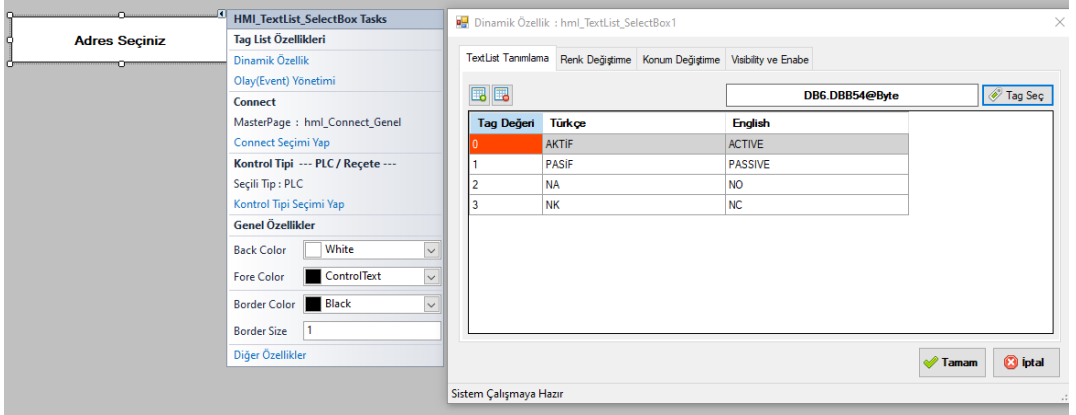
Connect Seçimi : HMI_TextList_SelectBox mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Dinamik Özellik : Bir Tag adresine göre , HMI_TextList_SelectBox un rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_TextList_SelectBox un listesinde görüntülenecek seçeneklerde TextList tanımlama bölümünde yazdığınız satırlardır.

TextList tanımlama bölümüne yazdığınız satırlar , HMI_TextList_SelectBox un seçenekleridir. Seçilen tag adresinin sayısal değerine göre o seçenek gösterilir , siz HMI_TextList_SelectBox dan bir seçenek seçerseniz , O seçeneğin sayısal değeri seçtiğiniz adrese gönderilir. Seçenekler , HMI_TextList_SelectBox tıkladığında ayrı bir Formda gösterilir.

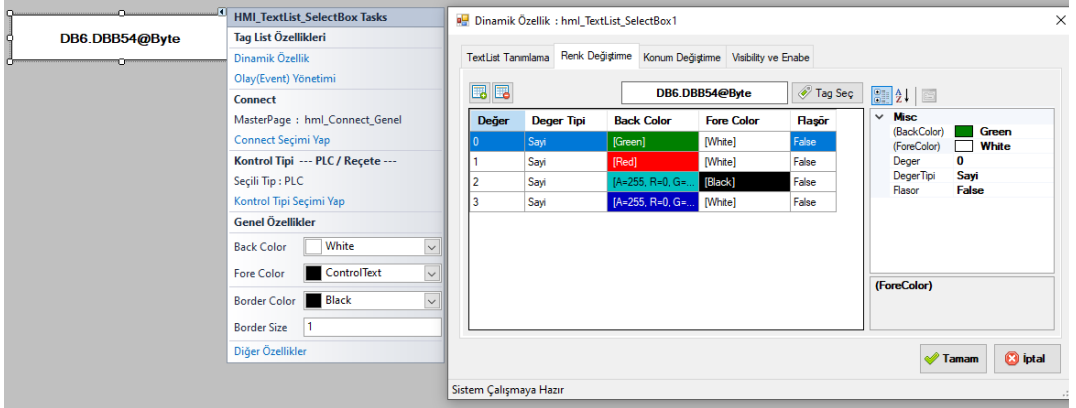


Aşağıdaki Örnekte 4 Seçenek girilmiştir. Ve bu seçenekler [DB6.DBB54@Byte](#) adresine bağlanmıştır. Tag Değeri karşılığındaki metin HMI_TextList_SelectBox da gösterilecek veya seçilebilecektir.

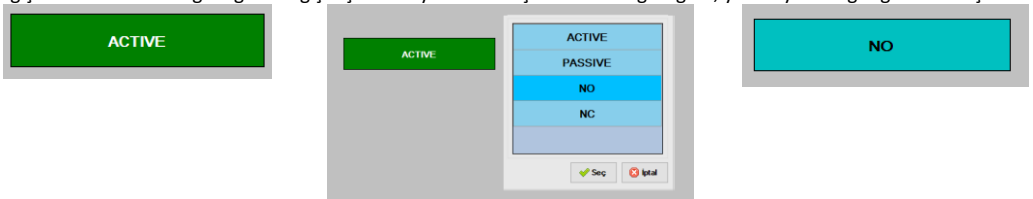


Ayrıca Her seçenek seçildiğinde , o seçeneğe göre arkaplan renginin ve yazı renginin değişmesini sağlayabilirsiniz. Dinamik özelliklerde Renk Değiştirme seçeneği ile bunu yapabilirsiniz.

Aşağıdaki Örnekte TextList ile aynı adrese "[DB6.DBB54@Byte](#)" Renk değiştirme dinamik özelliğini bağladık ve 4 seçenek için ayrı renkler tanımladık.

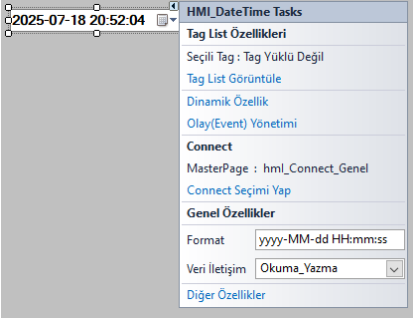


Aşağıda HMI_TextList_SelectBox dan Seçim Yapılmıştır , Active Seçimi NO Olarak Değiştirilmiştir ve Seçimin Rengide renk değıştirme dinamik özelliğine göre değıştirmiştir. PLC ye ise Bu Seçeneklerin Tag Değeri , yani sayısal değıer gönderilmiştir.



Diğer Özellikler : Yazı hizalama font size gibi özellikler buradan değıştirilebilir. HMI_Label ile benzer yapıya sahiptir , bundan dolayı tekrar anlatılmayacaktır.

HMI_DateTime : PLC deki bir tarih adresinin deęerinin gsterilmesi ve tıkladığında bu adresin deęerinin deęiřtirilebilmesini saęlar.

	<p>Tag List Grntle : HMI_DateTime in kontrol edeceęi Tag adresi buradan seilir.</p> <p>Dinamik zellik : Bir Tag adresine gre , konumunu , visibility ve enable zelliklerini deęiřtirebilirsiniz.</p> <p>Connect Seęimi : HMI_DateTime mutlaka bir HMI_Connect nesnesine baęlanmalıdır.</p> <p>Format : Tarihin Grntleme Formatı Belirlenir.</p> <p>Veri İletiřimi : Okuma_Yazma veya Sadece Okuma seilebilir. Sadece Okuma seilirse PLC Deęeri grntlenir ama HMI_DateTime zerinden deęiřtirilemez.</p>
---	---

Connect Seęimi : HMI_DateTime mutlaka bir HMI_Connect nesnesine baęlanmalıdır.

Format : Tarihin Grntleme Formatı Belirlenir. Format byk ve kk harfe duyarlıdır.

y: Yıl
M: Ay
d: Gn
H: Saat , 24 saat
h: Saat , 12 saat
m: dakika
s: saniye

nemli uyarı : Kk m ile Byk M birbirinden farklıdır , M: Ay , m:dakika , Format karakterleri kk ve byk harfe duyarlıdır.

rnek Gsterimler:

yyyy-MM-dd	2025-06-18
yy-M-dd	25-6-18
dd-MM-yyyy	18-06-2025
dd.MM.yyyy	18.06.2025
HH:mm:ss	21:09:55
yyyy-MM-dd HH:mm:ss	2025-06-18 21:09:55
dd.MM.yyyy HH:mm:ss	18.06.2025 21:09:55

Tag List Görüntüle ile HMI_DateTime in kontrol edeceği Tag adresi seçilir. Projemize DB10 General DataBlock unda Sample_DTL adresini HMI_DateTime a bağladık. HMI_DateTime DTL veya Date Tipini desteklemektedir

The screenshot shows the HMI software interface. The main window is titled 'HMI_DateTime Tasks' and 'Tag List Özellikleri'. It contains a 'Tag List Görüntüle' button and a 'Connect' section with 'MasterPage : hml_Connect_Genel' and 'Connect Seçimi Yap' button. The 'Genel Özellikler' section shows 'Format : yyyy-MM-dd HH:mm:ss' and 'Veri İletişim : Okuma_Yazma'. The 'Tag Listesi' window is open, showing a list of tags with columns 'Tag ID', 'Açıklama', and 'Tag Adresi'. The tag 'Sample_DTL' (ID 72) is selected. Below the list are buttons for 'BOŞ TAG', 'SEÇ', and 'İPTAL'.

Tag ID	Açıklama	Tag Adresi
64	System_Time	DB10.DTL0@DTL
65	Printer_Barcode	DB10.S12.255@String
66	Printer_Done	DB10.DBX268.0@Bool
67	Printer_Error	DB10.DBX268.1@Bool
68	Printer_Status	DB10.W270@Int
69	Sample_String_1	DB10.S272.255@String
70	Sample_String_2	DB10.S528.10@String
71	Sample_Real	DB10.R540@Real
72	Sample_DTL	DB10.DTL544@DTL
73	Sample_Date	DB10.D556@Date
74	Sample_Time	DB10.T558@Time

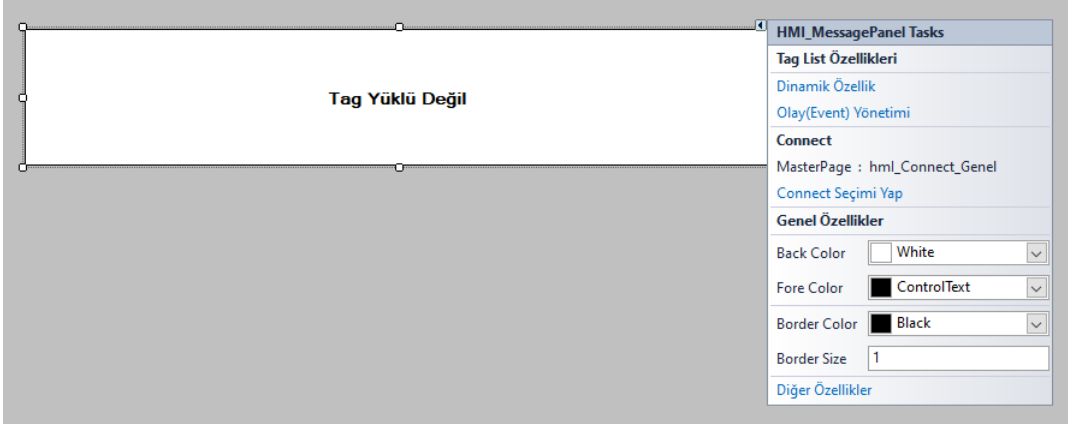
The screenshot shows a calendar widget for the month of July 2025. The date 18 is highlighted. Below the calendar, it says 'Bugün: 18.07.2025'.

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

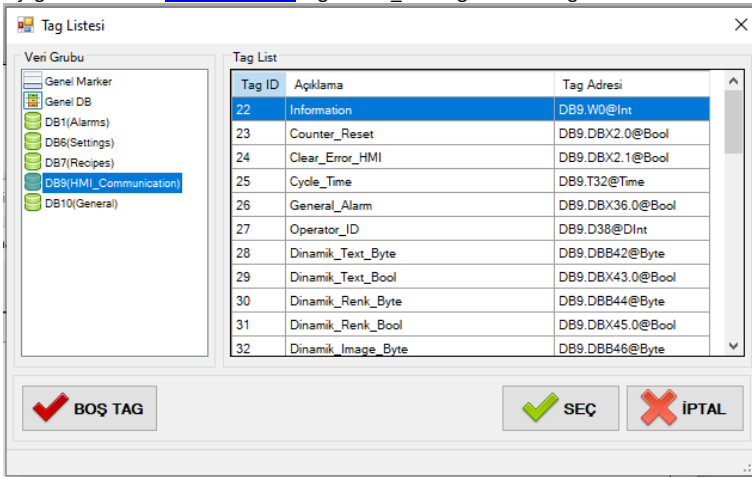
Projenizi çalıştırdığımızda, Veri İletişimi Okuma_Yazma ise , yandaki gibi tarih seçimi yapabilirsiniz.

HMI_MessagePanel : PLC Tag adresine göre bir metin listesinin gösterilmesi , PLC deki taga göre Renk , flaşör gibi özelliklerin gösterilmesini sağlar. Genel makine durumuyla ilgili kullanıcıya bilgiler vermek istiyorsanız , HMI_MessagePanel i kullanabilirsiniz.

- **Connect Seçimi** : HMI_MessagePanel mutlaka bir HMI_Connect nesnesine bağlanmalıdır.
- **Dinamik Özellik** : Bir Tag adresine göre , HMI_MessagePanel ın rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_MessagePanel in listesinde görüntülenecek seçeneklerde TextList tanımlama bölümünde yazdığınız satırlardır.



Aşağıdaki Örnekte [DB9.WO@Dint](#) tagını HMI_MessagePanel e bağladık



5 Adet Mesajı Dinamik Özellik -> TextList Tanımlama Ekranından Girdik

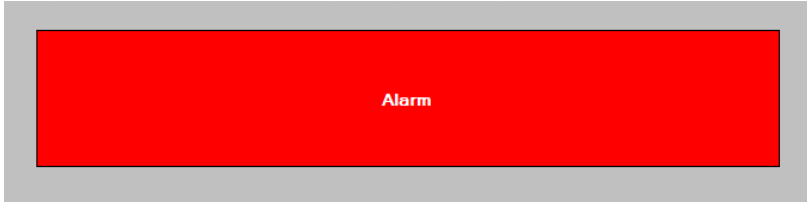
Tag Değeri	Türkçe	English
0	Alarm	Alarm
1	Uzun Reset Bekleniyor	Long Reset Waiting
2	Uzun Reset Yapılıyor	Long Reset In Progress
3	Manuel Hazır	Manual Ready
4	Otomatik Hazır	Automatic Ready

Bu Mesajların ArkaPlan ve Yazı Renklerini de Girdik , 0 nolu Mesaj yani Alarm mesajına Flasör Tanımladık

Değer	Değer Tipi	Back Color	Fore Color	Flaşör
0	Sayı	[Red]	[White]	True
1	Sayı	[Yellow]	[Black]	False
2	Sayı	[Gold]	[Black]	False
3	Sayı	[A=255, R=0, G=...]	[White]	False
4	Sayı	[Green]	[White]	False

Misc
(BackColor) Red
(ForeColor) White
Değer 0
Değer Tipi Sayı
Flaşör True

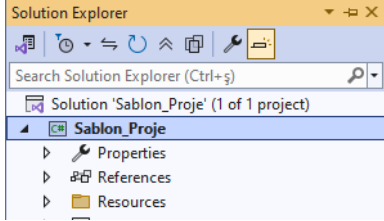
Projeyi Çalıştırdık [DB9.WO@Dint](#) tagının değeri 0 olduğu için Alarm Mesajı Görünmektedir



HMI_Imagelist : Resimlerin Bir PLC Tag adresine göre Gösterilmesi

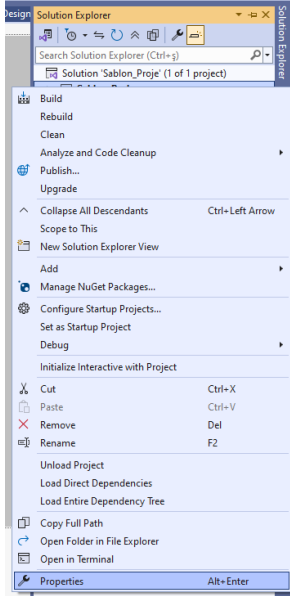
Öncelikle projemize nasıl resim ekleyeceğimizi anlatalım

1.



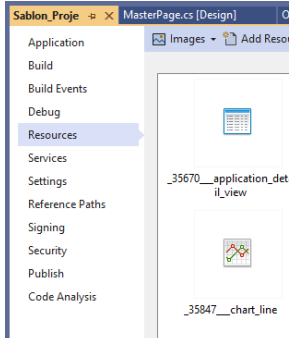
Solution Explorer da C# Simgesi İle Seçili Proje Adına Sağ Tıklayınız

2.



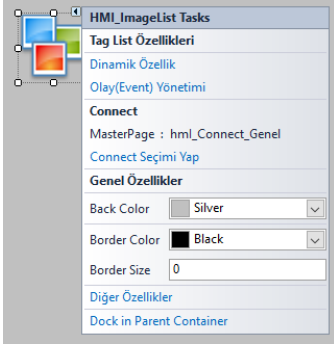
Sağ tıkladıktan sonra aşağıdaki **Properties** seçeneğini tıklayınız.

3.



Gelen ekrandan Resources seçeneğini tıklayınız ve sağdaki alana (Alanda resim değilde yazı olsa bile) resmi yapıştırınız.

Tabi öncelikle , bilgisayarınızdan bu alana yapıştıracağınız resmi kopyalamanız gerekmektedir.



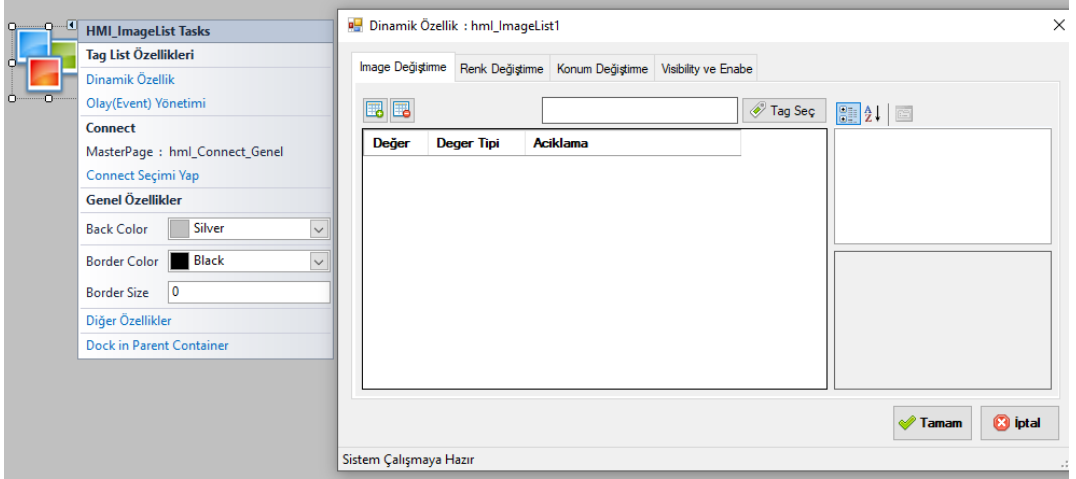
Dinamik Özellik : Bir Tag adresine göre , HMI_ImageList in Barındıracağı resimleri , rengini , konumunu , visibility ve enable özelliklerini değiştirebilirsiniz. Ayrıca HMI_ImageList in listesinde görüntülenecek seçeneklerde Image değiştirme bölümüne yükleyeceğimiz satırlardır.

Connect Seçimi : HMI_ImageList mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

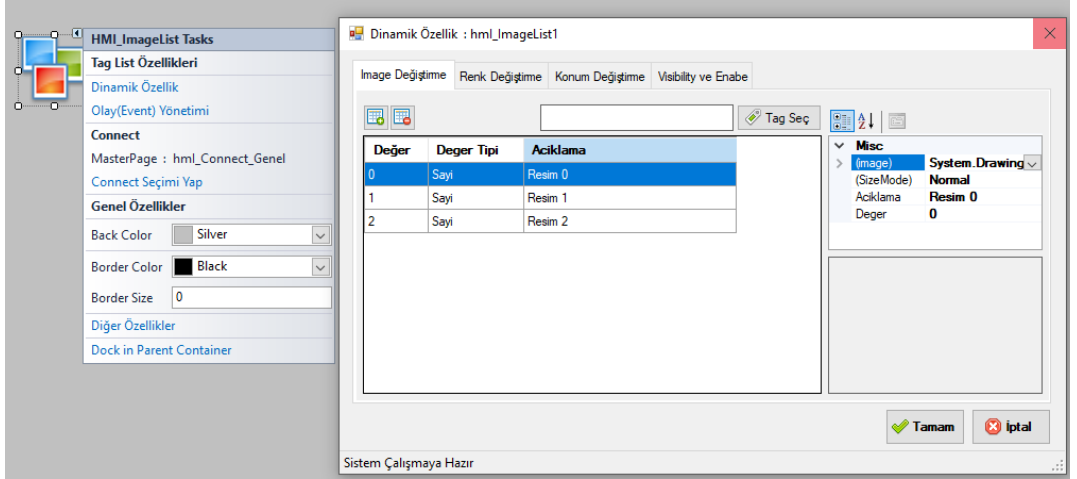
Connect Seçimi : HMI_ImageList mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Image değiştirme alanına , satlar ekleyiniz ve sağdaki özellik değiştirme ekranından Imagerleri seçiniz , seçtiğiniz image özellik ekranının altın görünmektedir , Açıklama ile , Resmin amacını yazabilirsiniz.

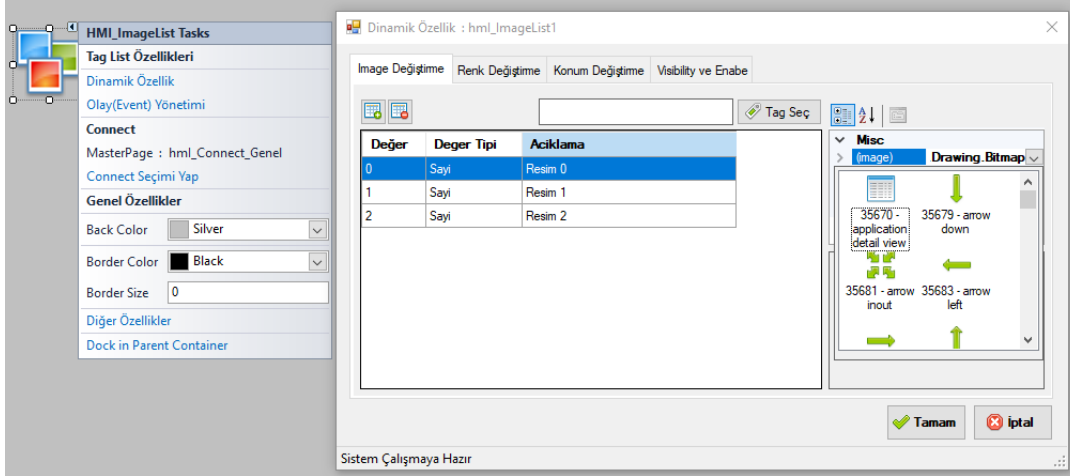
TagSeç ile Resim değişimlerinin hangi adresin değerine göre yapılacağını seçiniz.



Aşağıda 3 satır eklendi

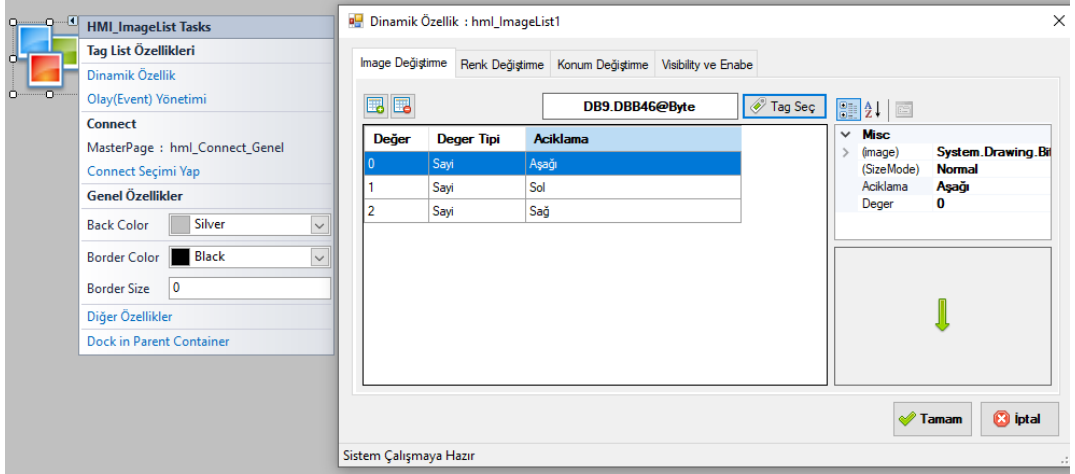


Aşağıda ise , sağdaki özellik ekranından resim seçildi , bu resimler daha önce anlattığımız , Resources e eklediğimiz resimlerdir



Aşağıda ise Imagelerin hangi tag adresinin değerine göre değişeceğini tanımladık , Burada Tag Seç ile [DB9.DBB46@Byte](#) tag adresine bağladık. Bu adresin değeri , her bir satır için tanımladığımız değer ile eşleştğinde o satırın resmi kontrolde gösterilecektir.

Ayrıca her bir satır için açıklama girişide yapıldı.



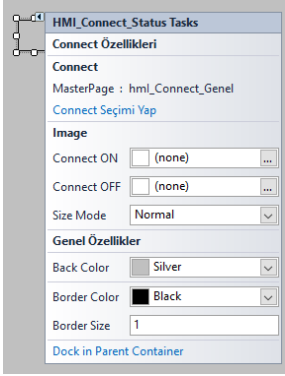
The screenshot displays a software configuration window titled "Dinamik Özellik : hml_ImageList1". The window is divided into several sections:

- Image Değiştirme**: Includes a "Tag Seç" field with the value "DB9.DBB46@Byte".
- Table**: A table with three columns: "Değer", "Deger Tipi", and "Acıklama". The table contains three rows:

Değer	Deger Tipi	Acıklama
0	Sayı	Aşağı
1	Sayı	Sol
2	Sayı	Sağ
- General Properties**: Includes "Back Color" (Silver), "Border Color" (Black), and "Border Size" (0).
- Misc**: A panel on the right showing "System.Drawing.Br" and "Normal" settings.

At the bottom of the window, there are "Tamam" (OK) and "İptal" (Cancel) buttons, and a status bar indicating "Sistem Çalışmaya Hazır".

HMI_Connect_Status : PLC Bağlantı durumunun görsel olarak sunulması

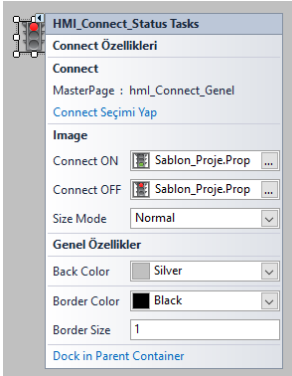


Connect Seçimi : HMI_Connect_Status mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Connect ON : PLC e Bağlanmışsa buraya yüklediğiniz resim görüntülenir.

Connect OFF : PLC e **Bağlanılmamışsa** buraya yüklediğiniz resim görüntülenir.

Size Mode : Resmin kontrol içerisindeki gösterim şeklini belirler.

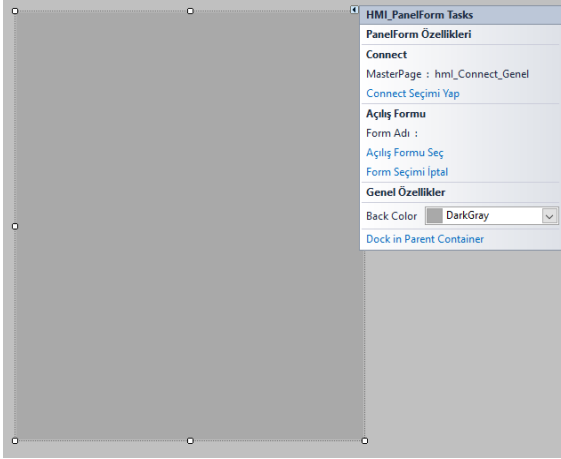


Burdaki örnekte , Connect ON da yeşil bir trafik lambası , Connect OFF da ise kırmızı bir trafik lambası yüklenmiştir.

PLC e bağlantı kurulduğunda yeşil , bağlantı kurulamazsa veya koparsa kırmızı trafik lambası resmi görüntülenir.

HMI_PanelForm : İçerisine Form alır ve Olay Komutu ile içerisine Form yüklenebilir. Çok önemli bir kontroldür. **Projede tanımladığımız Formlar temelde bu kontrol aracılığı ile gösterilir.**

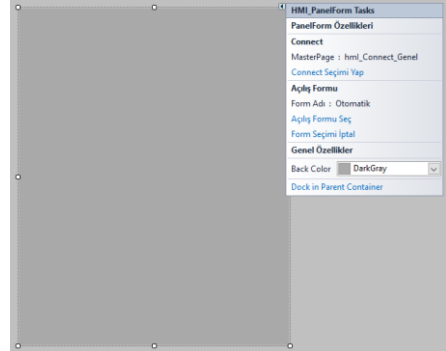
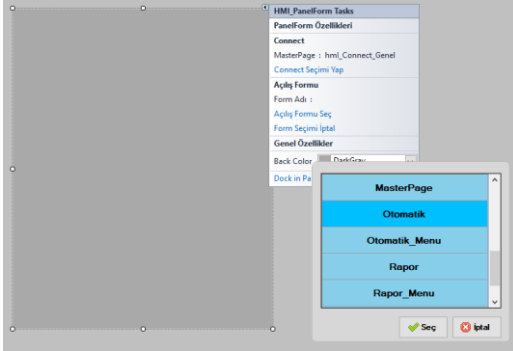
HMI_PanelForm umumuzun amacı , projemizde tanımladığımız Form ların gösterimidir. Form un Size özelliği ile HMI_PanelForm un Size özelliğinin aynı değerde olmasına dikkat ediniz.



Connect Seçimi : HMI_PanelForm mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Açılış Formu : İsterseniz HMI_PanelForm görüntülediğinde içerisinde başlangıç (Açılış) formu yüklemesi yapabilirsiniz. Buraya yükleyeceğimiz form , PanelForm görüntülediğinde direkt yüklenir.

Aşağıda HMI_PanelForm nesnesinin açılış formuna , Otomatik Form Sayfası atanmıştır.

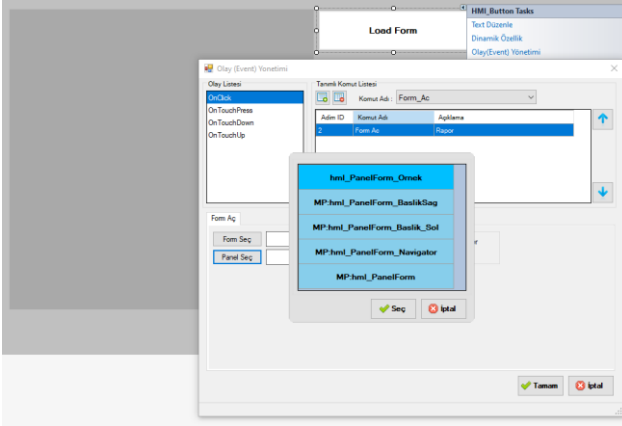


Genel kullanımı HMI_Button nesnesi iledir. HMI_Button nesnesinin Olay Event yönetiminde OnClick Olayına Form_Ac komutu eklenir. Buradan HMI_PanelForm ve HMI_PanelForm içerisine yüklenecek **Form** Seçimi yapılır.

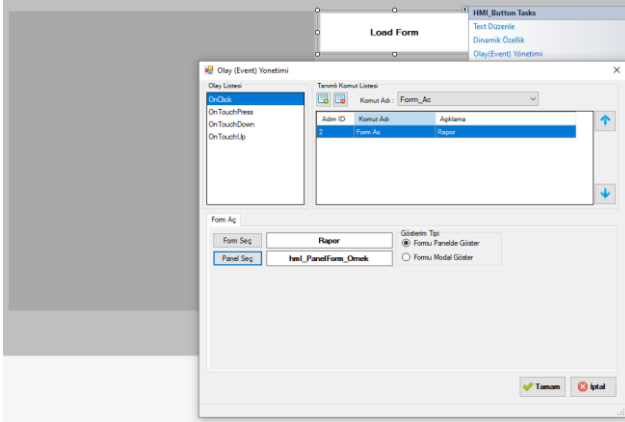


Text Düzenle ile HMI_Button nesnesinin üzerine Load Form Yazınız

HMI_Button nesnesinin Olay Event Yönetimini Açınız



OnClick Olayına Form_Ac Komutunu ekleyiniz , aşağıdaki Form Aç parametrelerinden , Form ve HMI_PanelForm seçimi yapınız



Bu Örneğimizde , HMI_Button nesnesine tıklandığında hmi_PanelForm_Ornek içerisine Rapor formu yüklenip gösterilecektir.



HMI_Button nesnesine (Load Form) tıkladığında hmi_PanelForm_Ornek içerisine Rapor formu yüklenip gösterilecektir.

HMI_Security : Yetkili Kullanıcı Oluşturmanızı sağlar.

Aşağıda Komple HMI_Security nesnesini görmekteyiz. Bu nesneyi HMI_Connect nesnesine bağladığımızda , Kullanıcıların çalışma zamanında kullanıcı tanımlamasını sağlayabilirsiniz. Böylece kullanıcı tanımlamayı dinamik hale getirmiş olursunuz.

Kullanıcı Tanımlama : Design zamanında kullanıcı tanımlamanızı sağlar.

Connect Seçimi : HMI_Security mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Dinamik Özellik : Bir Tag adresine göre , HMI_Security nin konumunu , visibility ve enable özelliklerini değiştirebilirsiniz.

Login Zamanı : Kullanıcının Login kalacağı zamanı belirler.

ID	Kullanıcı Adı	Şifre	Açıklama
----	---------------	-------	----------

HMI_Security Tasks

Yetki Özellikleri

[Kullanıcı Tanımla](#)

[Dinamik Özellik](#)

Connect

MasterPage : hml_Connect_Genel

[Connect Seçimi Yap](#)

Genel Özellikler

Login Zamanı (Sn)

[Diğer Özellikler](#)

HMI_Serial : Seri portun okuma ve yazmasını PLC adreslerine bağı olarak yapar ve Seri portdan alınan veriyi PLC e gönderir veya PLC den aldığı veriyi Seri Porta yazar

***Kontrol mutlaka MasterPage formunda olmalıdır.**

HMI_Serial Tasks
Tag List Özellikleri

SERİYE YAZ
Yazma_izin Aktif
Seçili Tag : None
Seriye Yaz Emri İçin Tag Seç
Seçili Tag : None
Seriye Yazıldı İçin Tag Seç
Seçili Tag : None
Yazılacak Veri İçin Tag Seç

SERİDEN OKU
Okuma_izin Aktif
Seçili Tag : None
Seriden Okundu İçin Tag Seç
Seçili Tag : None
Okunan Veri İçin Tag Seç

SERİ PORT VERİLERİ
PortName COM1
BaudRate 9600
Parity None
DataBits 8
StopBits One

Connect
None : None
Connect Seçimi Yap
Dock in Parent Container

Yazma İzin : Ancak bu özellik aktifse , Seriyeye yazar , Pasif ise , PLC den yaz emri gelsede yazmaz.

Seriye Yaz Emri : Seçilen Bool tipindeki tag adresi değeri , True olduğunda (Yükselen kenar) Yazılacak veri özelliğinde tanımlı tagın değeri seri porta gönderilir. Tekrar emir yollanması için tagın değeri False yapılması gerekmektedir. (yükselen kenar : PLC Terimidir , yani birkez çalışır ve tekrar çalışması için tag değerinin false olması beklenir)

Seriye yazıldı : Seri Port a yaz emri geldikten sonra PLC e geri bildirim yapılır. PLC tarafında bu adresin false yapılmasını unutmayınız , ideal i veri geldikten 2-3sn sonra false yapılmasıdır.

Yazılacak Veri : Seçilen String tipindeki tag adresi değeri , Seriyeye yaz emri geldikten sonra seri porta gönderilir. Yani seri porta gönderilecek veri bu özellik ile belirlenen tag adresinin değeridir.

Okuma İzin : Ancak bu özellik aktifse , seriden okur , Pasif ise , Seri portdan okuma işlemi yapmaz.

Seriden Okundu : Seri port dan bir veri gelirse (Barkod okuyucu vb.) Veriyi Okunan veri ile seçilen Tag adresine yazar daha sonra bu adresi true yapar. **Önemli uyarı : Bu adres true olduktan 1-2 sn sonra Okunan veri adresinden veriyi alınız. Önceki okunan veri ile bu verinin karışmaması için PLC de önlem almayı unutmayınız.**

Okunan Veri : Seri portdan okunan veri , burada seçtiğiniz String tipindeki tag adresine değer olarak gönderilir.

Connect Seçimi : HMI_Serial mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Seri Port Verileri : Seri Portdan bağlanacağınız cihazın Seri Port Bağlanma Verileri buraya yazılır.

Connect Seçimi : HMI_Serial mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

***İşlem Yapmadığınız Kanalı Pasif Yapınız , Okuma Yapmayacaksanız Okuma izin veya Yazma Yapmayacaksanız Yazma izin i Pasif Yapınız.**

DB11 SerialPort datablock unu tanımlıyoruz

DB11(SerialPort)			
	Name	Data type	Offset
1	Static		
2	Write_To_SerialPort	Bool	0.0
3	Written_To_SerialPort	Bool	0.1
4	Data_To_Write	String	2.0
5	Read_From_Serial	Bool	258.0
6	Data_Read	String	260.0

Bu DataBlock u HMI_Connect Tag Listesine Eklıyoruz

Tag Listesi

Ven Grubu

- Genel Marker
- Genel DB
- DB1(Alarms)
- DB6(Settings)
- DB7(Recipes)
- DB9(HMI_Communication)
- DB10(General)
- DB11(SerialPort)

Tag List

Tag ID	Açıklama	Tag Adresi
79	Write_To_SerialPort	DB11.DBX0.0@Bool
80	Written_To_SerialPort	DB11.DBX0.1@Bool
81	Data_To_Write	DB11.S2.255@String
82	Read_From_Serial	DB11.DBX258.0@Bool
83	Data_Read	DB11.S260.255@String

Cross Reference

Form Adı	Nesne Adı	Tag ID
----------	-----------	--------

HMI_Serial Tasks

Tag List Özellikleri

SERİYE YAZ

Yazma_izin

Seçili Tag : DB11.DBX0.0@Bool
[Seriye Yaz Emri İçin Tag Seç](#)

Seçili Tag : DB11.DBX0.1@Bool
[Seriye Yazıldı İçin Tag Seç](#)

Seçili Tag : DB11.S2.255@String
[Yazılacak Veri İçin Tag Seç](#)

SERİDEN OKU

Okuma_izin

Seçili Tag : DB11.DBX258.0@Bool
[Seriden Okundu İçin Tag Seç](#)

Seçili Tag : DB11.S260.255@String
[Okunan Veri İçin Tag Seç](#)

SERİ PORT VERİLERİ

PortName

BaudRate

Parity

DataBits

StopBits

Connect

MasterPage : hml_Connect_Genel
[Connect Seçimi Yap](#)
[Dock in Parent Container](#)

Yanda , Bu tanımladığımız Tag adresleri , ilgili özelliklere atanmıştır.

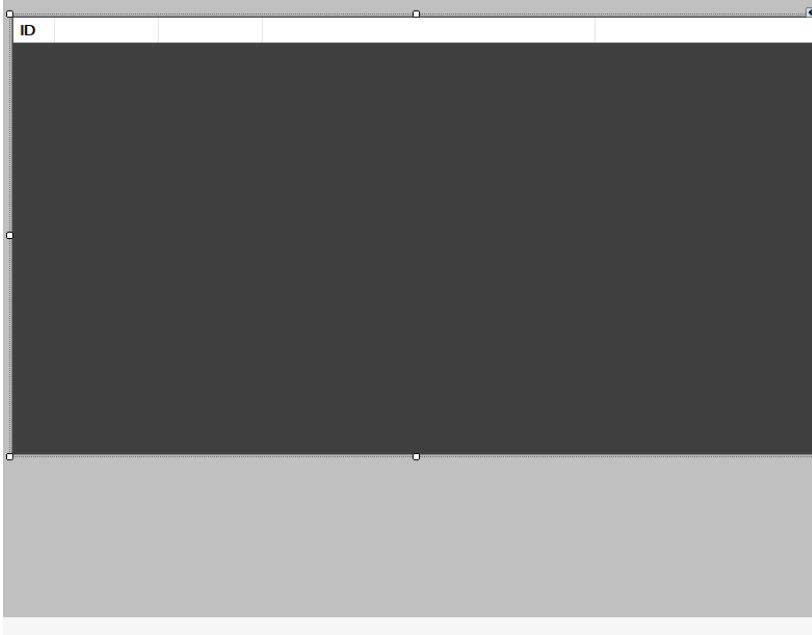
HMI_AlarmList : Alarmların Listelenmesi , projenize bir Alarm sistemi yüklersiniz.

Connect Seçimi : HMI_AlarmList mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Alarm Listesi : Alarm tanımlamaları buradan yapılmalıdır. Alarmlar için mutlaka Byte tipi değişken tanımlanmalıdır.

Diğer Özellikler : Sütunların Başlık isimlerini Dil desteği ile birlikte düzenleyebilir , görünürlüğü değiştirip , genişliğini ayarlayabilirsiniz. Ayrıca yazı hizalama font size gibi özellikler buradan değiştirilebilir. Size ve konum ile hassas Size ve konumlama yapabilirsiniz , Hassas Border girişi yapıp , Static bir şekilde Enable ve Visiblesini değiştirebilirsiniz.

Sütun Bilgileri : İlgili sütunları görünür yapıp genişliklerini değiştirebilirsiniz



HMI_AlarmList Tasks

Özellikler

Alarm Listesi

Dinamik Özellik

Connect

MasterPage : hml_Connect_Genel

[Connect Seçimi Yap](#)

Genel Özellikler

Back Color : 64; 64; 64

Fore Color : Gold

Border Color : Black

Border Size : 1

ID Sütunu

ID Genişlik : 40

Adres Genişlik : 100

Tarih Sütunu

Tarih Genişlik : 100

Saat Sütunu

Saat Genişlik : 100

Mesaj Sütunu

Mesaj Genişlik : 340

[Diğer Özellikler](#)

[Dock in Parent Container](#)

Diğer özellikler : Sütun Özellikleri ile Sütunların Başlık isimlerini Dil desteği ile birlikte düzenleyebilir , görünürlüğüne değiştirip , genişliğini ayarlayabilirsiniz. Bu bölüme mutlaka giriş yapınız.

Diğer Özellikler

HMI AlarmList

Sütun Özellikler | Text | Size | Konum | Genel

Adres Sütunu		Tarih Sütunu	
Genişlik : 100		Sütun Aktif : <input checked="" type="checkbox"/> Genişlik : 100	
Türkçe	English	Türkçe	English
Saad Sütunu		Alarm Mesajı Sütunu	
Sütun Aktif : <input type="checkbox"/> Genişlik : 100		Sütun Aktif : <input checked="" type="checkbox"/> Genişlik : 340	
Türkçe	English	Türkçe	English

Tamam
 İptal

Sistem Çalışmaya Hazır

Diğer Özellikler

HMI AlarmList

Sütun Özellikler | Text | Size | Konum | Genel

Adres Sütunu		Tarih Sütunu	
Genişlik : 100		Sütun Aktif : <input checked="" type="checkbox"/> Genişlik : 100	
Türkçe	English	Türkçe	English
Adres	Address	Tarih	Date
Saad Sütunu		Alarm Mesajı Sütunu	
Sütun Aktif : <input type="checkbox"/> Genişlik : 100		Sütun Aktif : <input checked="" type="checkbox"/> Genişlik : 340	
Türkçe	English	Türkçe	English
Saad	Time	Mesaj	Message

Tamam
 İptal

Sistem Çalışmaya Hazır

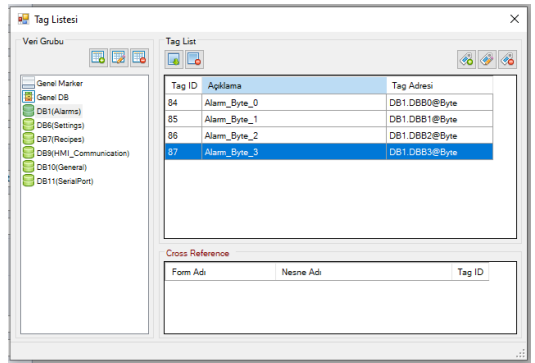
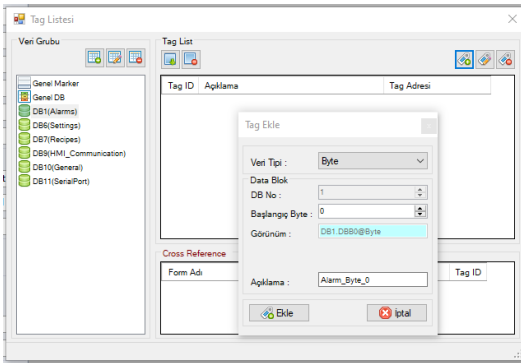
DB1(Alarms)			
	Name	Data type	Offset
1	Static		
2	Alarm_EmergencyStop	Bool	0.0
3	Alarm_AirOK	Bool	0.1
4	Alarm_Reserved_3	Bool	0.2
5	Alarm_Reserved_4	Bool	0.3
6	Alarm_Piston_1_Forward_Error	Bool	0.4
7	Alarm_Piston_1_Backward_Error	Bool	0.5
8	Alarm_Piston_1_Sensor_Error	Bool	0.6
9	Alarm_Piston_1_Permission_Error	Bool	0.7
10	Alarm_Piston_2_Forward_Error	Bool	1.0
11	Alarm_Piston_2_Backward_Error	Bool	1.1
12	Alarm_Piston_2_Sensor_Error	Bool	1.2
13	Alarm_Piston_2_Permission_Error	Bool	1.3
14	Alarm_Piston_3_Forward_Error	Bool	1.4
15	Alarm_Piston_3_Backward_Error	Bool	1.5
16	Alarm_Piston_3_Sensor_Error	Bool	1.6
17	Alarm_Piston_3_Permission_Error	Bool	1.7
18	Alarm_Piston_4_Forward_Error	Bool	2.0
19	Alarm_Piston_4_Backward_Error	Bool	2.1
20	Alarm_Piston_4_Sensor_Error	Bool	2.2
21	Alarm_Piston_4_Permission_Error	Bool	2.3
22	Alarm_Rezeve_21	Bool	2.4
23	Alarm_Rezeve_22	Bool	2.5
24	Alarm_Rezeve_23	Bool	2.6
25	Alarm_Rezeve_24	Bool	2.7
26	Alarm_Rezeve_25	Bool	3.0
27	Alarm_Rezeve_26	Bool	3.1
28	Alarm_Rezeve_27	Bool	3.2
29	Alarm_Rezeve_28	Bool	3.3
30	Alarm_Rezeve_29	Bool	3.4
31	Alarm_Rezeve_30	Bool	3.5
32	Alarm_Rezeve_31	Bool	3.6
33	Alarm_Rezeve_32	Bool	3.7

Bu alarm sistemini kullanabilmeniz için Alarmları yandaki gibi Ayrı bir datablock da Bool değişkenler olarak tanımlamanız gerekmektedir. Tanımladığınız bu alarmlar , true olursa görüntülenir , false olursa AlarmList den silinir.

- Ayrı bir Alarm DataBlock u tanımlayınız
- Bool değişkenler halinde bu alarmları listeleyiniz.
- Son alarmın adresinin (Offset) .7 olmasına dikkat ediniz
- Bunun için gerekirse rezerve tanımlayınız

HMI_Connect e Alarm taglarını ekleyelim , Bu ekleme diğer eklemelerden farklıdır. Eklemeyi Byte şeklinde yapacağız. Öncelikle DB1(Alarms) datablock unu Veri grubuna tanımlayalım , Daha sonra Manuel bir şekilde Byte olarak Tag eklemesini yapacağız . Aşağıdaki iki görseli dikkatlice inceleyiniz. 4 adet tag eklenmiştir. Bu bytelar ile DB1 deki tüm alarmlara ulaşabiliriz.

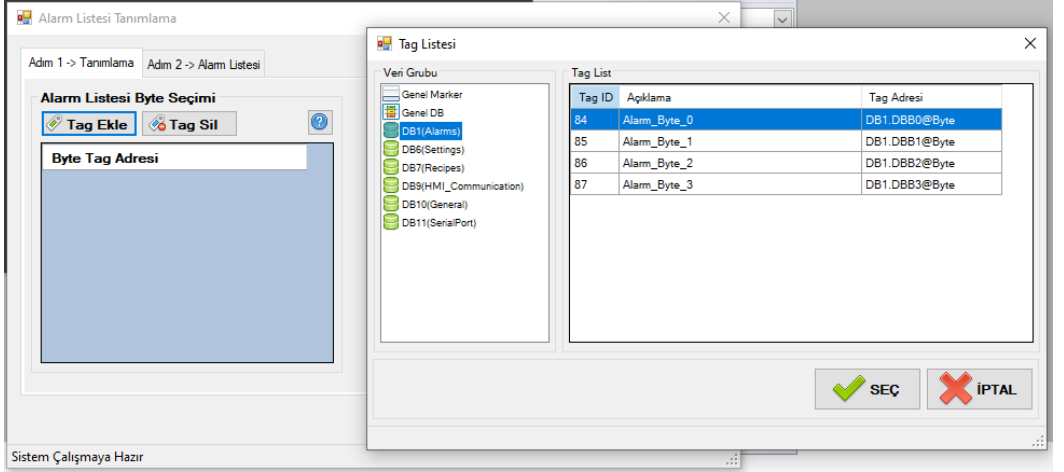
- [DB1.DBB0@Byte](#) ile 0.0 - 0.7 arasındaki alarmlara ulaşırız
- [DB1.DBB1@Byte](#) ile 1.0 - 1.7 arasındaki alarmlara ulaşırız
- [DB1.DBB2@Byte](#) ile 2.0 - 2.7 arasındaki alarmlara ulaşırız
- [DB1.DBB3@Byte](#) ile 3.0 - 3.7 arasındaki alarmlara ulaşırız



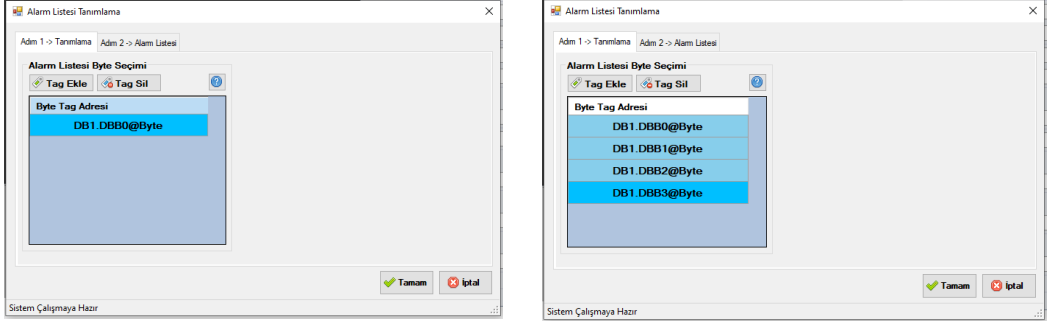
Alarm tanımlamalarını yapmak işi n , HMI_AlarmList in sağ üstündeki kulakçığa tıklayınız ve Alarm Listesi özelliğine tıklayınız.

Aşağıda Alar Listesi ekranını görüyorsunuz. Bu ekran iki sekmeden oluşmaktadır.

1. Alarmların bağlanacağı Byte adresleri TagList den seçeceğiniz bölüm. Ekranlarda Görüldüğü gibi 4 Adet adres aşağıdaki gibi Eklenmiştir.

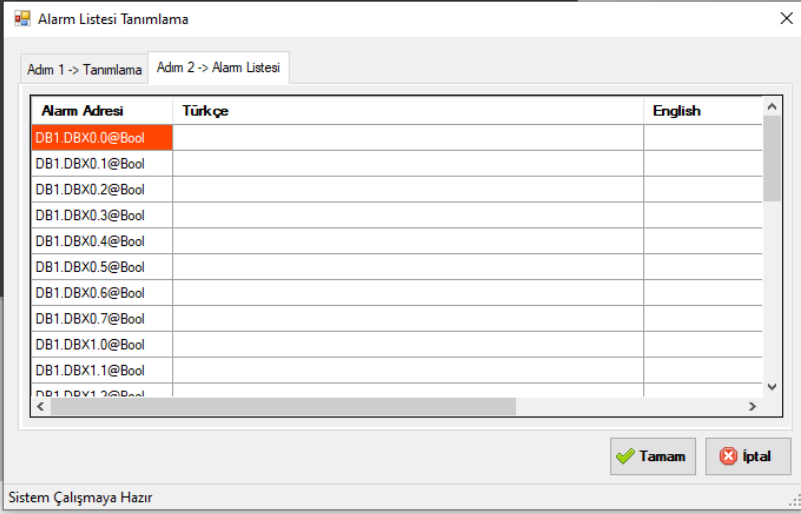


Aşağıdaki gibi tagları ekledikten sonra bir sonraki aşamaya , Alarm Listesini tanımlamaya geçebiliriz.



- Alarm listesi bölümünde ise , Adım 1 de tanımladığımız Byte alanlar , Bool alanlara otomatik dönüştürülerek listelenir. İsterseniz manuel Bu alanların karşılıklarını (Alarm metinleri) doldurabileceğiniz gibi , isterseniz kopyala-yapıştır ile de bu alanları doldurabilirsiniz.

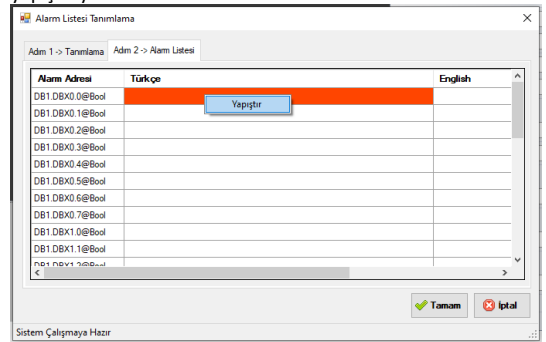
Biz Kopyala – yapıştır ile tüm Alarm metinlerini bu satırlara kolayca yapıştıracağız



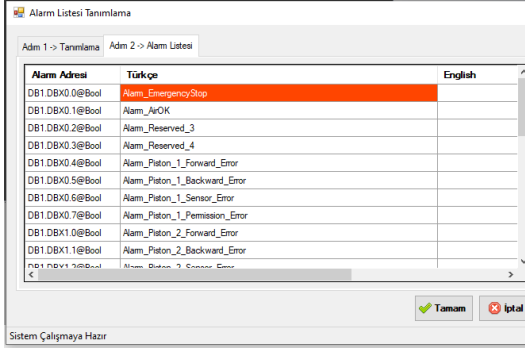
Öncelikle Alarm DataBlock undan tüm alarm metinlerini aşağıdaki gibi kopyalıyoruz.

Name	Data type	Offset	Start value
Static			
Alarm_EmergencyStop	Bool	1.6	false
Alarm_AirOK	Bool	1.7	false
Alarm_Reserved_3	Bool	2.0	false
Alarm_Reserved_4	Bool	2.1	false
Alarm_Piston_1_Forward_Error	Bool	2.2	false
Alarm_Piston_1_Backward_Error	Bool	2.3	false
Alarm_Piston_1_Sensor_Error	Bool	2.4	false
Alarm_Piston_1_Permission_Error	Bool	2.5	false
Alarm_Piston_2_Forward_Error	Bool	2.6	false
Alarm_Piston_2_Backward_Error	Bool	2.7	false
Alarm_Piston_2_Sensor_Error	Bool	3.0	false
Alarm_Piston_2_Permission_Error	Bool	3.1	false
Alarm_Piston_3_Forward_Error	Bool	3.2	false
Alarm_Piston_3_Backward_Error	Bool	3.3	false
Alarm_Piston_3_Sensor_Error	Bool	3.4	false
Alarm_Piston_3_Permission_Error	Bool	3.5	false
Alarm_Reserved_24	Bool	3.6	false
Alarm_Reserved_25	Bool	3.7	false
Alarm_Reserved_26	Bool	3.8	false
Alarm_Reserved_27	Bool	3.9	false
Alarm_Reserved_28	Bool	4.0	false
Alarm_Reserved_29	Bool	4.1	false
Alarm_Reserved_30	Bool	4.2	false
Alarm_Reserved_31	Bool	4.3	false
Alarm_Reserved_32	Bool	4.4	false

Alarm Listesinden ilk satırı seçiyoruz ve farenin sağ tuşuna tıklıyoruz. Sağ tıkladığımızda ekrana Yapıştır seçeneği gelir. Bu seçeneği tıklayıp kopyaladığımız Alarm Metinlerini yapıştırıyoruz.



Önce yapıştırma işlemini isteyip istemediğinize dair bir uyarı mesajı gelir ve siz Tamam dersanız , Alarm metinlerini aşağıdaki gibi kolayca yapıştırırsınız.



Alarm Listesi Tanımlama

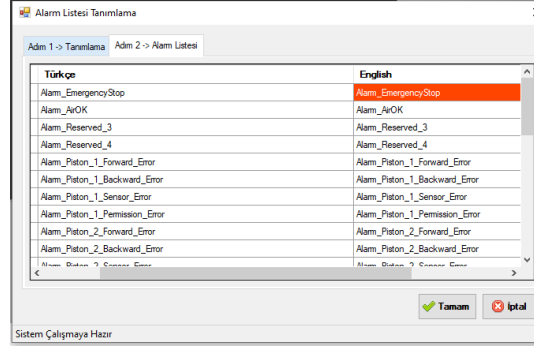
Adm 1 -> Tanımlama Adm 2 -> Alarm Listesi

Alarm Adresi	Türkçe	English
DB1.DBX0.0@Bool	Alarm_EmergencyStop	
DB1.DBX0.1@Bool	Alarm_AirOK	
DB1.DBX0.2@Bool	Alarm_Reserved_3	
DB1.DBX0.3@Bool	Alarm_Reserved_4	
DB1.DBX0.4@Bool	Alarm_Piston_1_Forward_Error	
DB1.DBX0.5@Bool	Alarm_Piston_1_Backward_Error	
DB1.DBX0.6@Bool	Alarm_Piston_1_Sensor_Error	
DB1.DBX0.7@Bool	Alarm_Piston_1_Permission_Error	
DB1.DBX1.0@Bool	Alarm_Piston_2_Forward_Error	
DB1.DBX1.1@Bool	Alarm_Piston_2_Backward_Error	
DB1.DBX1.2@Bool	Alarm_Piston_2_Sensor_Error	

Sistem Çalışmaya Hazır

Tamam İptal

Alarm metinlerini Türkçe bölümüne yapıştırdık , aynı şekilde tekrar kopyalayıp aşağıdaki gibi İngilizce bölümünde yapıştırabilirsiniz. Seçimini yaptığınız dilin Alarm metinleri , ekranda gösterilir.



Alarm Listesi Tanımlama

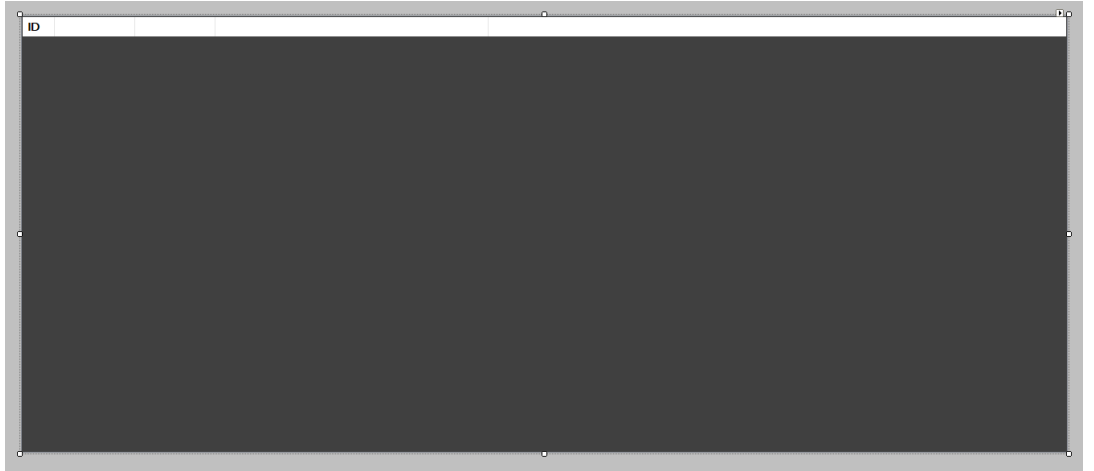
Adm 1 -> Tanımlama Adm 2 -> Alarm Listesi

Türkçe	English
Alarm_EmergencyStop	Alarm_EmergencyStop
Alarm_AirOK	Alarm_AirOK
Alarm_Reserved_3	Alarm_Reserved_3
Alarm_Reserved_4	Alarm_Reserved_4
Alarm_Piston_1_Forward_Error	Alarm_Piston_1_Forward_Error
Alarm_Piston_1_Backward_Error	Alarm_Piston_1_Backward_Error
Alarm_Piston_1_Sensor_Error	Alarm_Piston_1_Sensor_Error
Alarm_Piston_1_Permission_Error	Alarm_Piston_1_Permission_Error
Alarm_Piston_2_Forward_Error	Alarm_Piston_2_Forward_Error
Alarm_Piston_2_Backward_Error	Alarm_Piston_2_Backward_Error
Alarm_Piston_2_Sensor_Error	Alarm_Piston_2_Sensor_Error

Sistem Çalışmaya Hazır

Tamam İptal

Alarm listesini bu şekilde kolayca hazırlayabilirsiniz. Projenizi çalıştırdığınızda ilgili sütunların başlıklarında ekranda gösterilecektir.



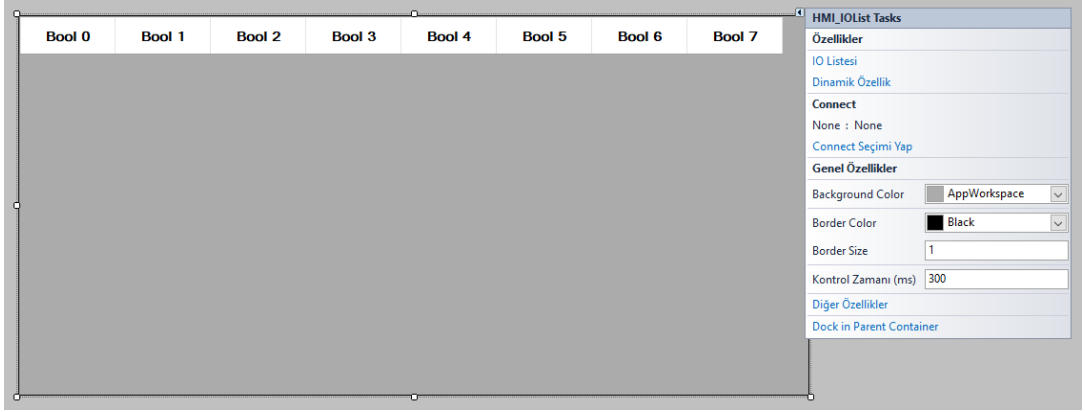
ID

HMI_IOList : Fiziksel IO ların durumlarının gösterilmesi , projenize rahatlıkla fiziksel giriş çıkışların durumunu gösteren bir kontrol yükleyebilirsiniz.

Connect Seçimi : HMI_IOList mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

IO Listesi : IO Listesi tanımlamaları bu özellik ile yapılmaktadır.

Kontrol Zamanı : IOListin güncelleme zamanıdır. Tabiki PLC iletişimine göre bu güncelleme değişiklik gösterebilir.

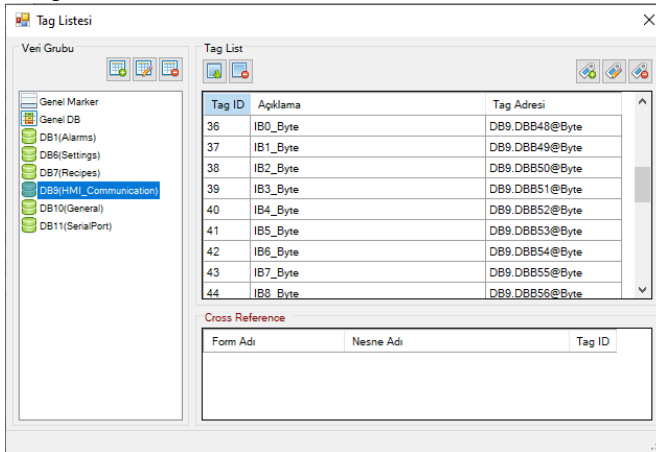


Öncelikle HMI_Connect nesnesi içerisinde Fiziksel IO ile bağlantılı Adresleri Tanımlıyoruz. Bu örnekte DB9 içerisinde adresleri ekledik , HMI_IOList component konusunun sonunda da Fiziksel adresin LAD de Bu adreslere nasıl atanacağını (Move) göstereceğiz.

Fiziksel IB0 -> [DB9.DBB48@Byte](#)

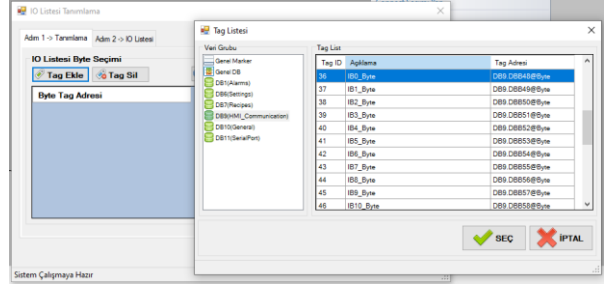
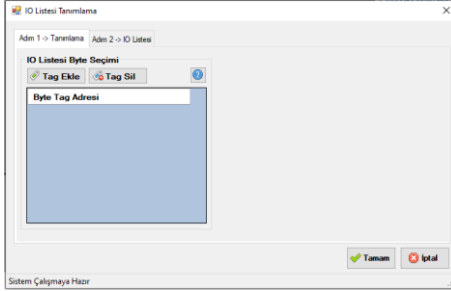
Fiziksel IB1 -> [DB9.DBB49@Byte](#)

Adresleri ile IO Gösterimi sağlanacaktır.

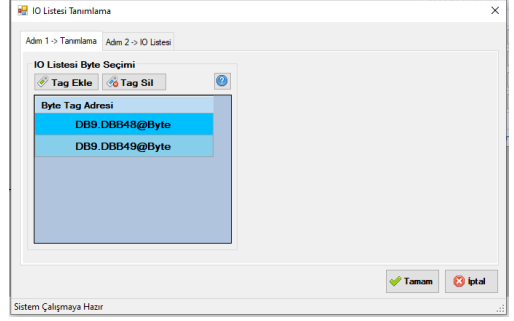
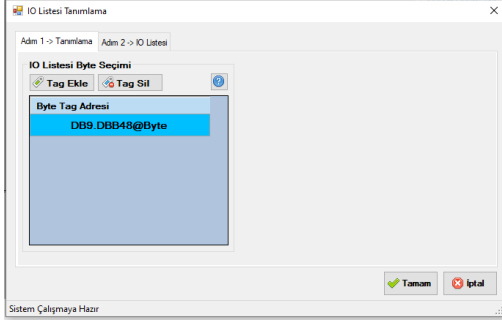


HMI_IOList kontrolünün sağ üstündeki kulakçığı tıklayınız ve IO Listesi özelliğine tıklayınız.

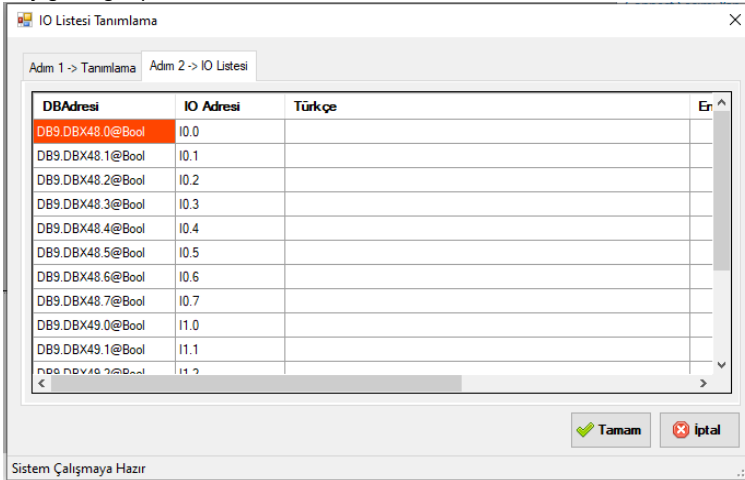
Buradan HMI_Connect de tanımlı [DB9.DBB48@Byte](#) ve [DB9.DBB49@Byte](#) adreslerini ekleyiniz.



Adım 1 de Tag Ekle ile iki Tag adresini ekliyoruz. Aşağıdaki görselleri dikkatlice inceleyiniz.



Adım 2 de ise otomatik olarak Bu tag adresleri Bool değerlere dönüştürülmüştür. Her bir Bool adresin karşılığındaki Fiziksel IO adresini aşağıdaki gibi yazınız.



Artık Fiziksel adreslerin karşılığı olan IO Listemiz hazır.

Bool 0	Bool 1	Bool 2	Bool 3	Bool 4	Bool 5	Bool 6	Bool 7
I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7
I1.0	I1.1	I1.2	I1.3	I1.4	I1.5	I1.6	I1.7

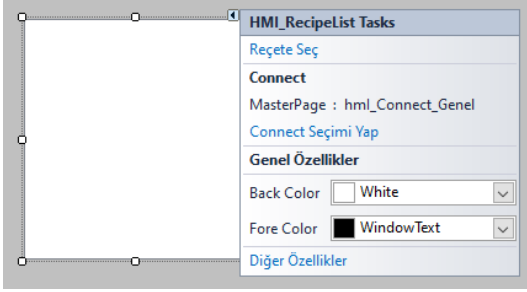
Aşağıdaki gibi HMI_IOList kontrolümüzün üstüne HMI_Label kontrolü nü ekleyerek görsel bir başlık oluşturabiliriz.

INPUTS							
Bool 0	Bool 1	Bool 2	Bool 3	Bool 4	Bool 5	Bool 6	Bool 7
I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7
I1.0	I1.1	I1.2	I1.3	I1.4	I1.5	I1.6	I1.7

Aşağıda ise LAD ile fiziksel IO nun değerinin [DB9.DBB48@Byte](#) ve [DB9.DBB49@Byte](#) adreslerine atanması gösterilmiştir. Bu atamadan sonra Fiziksel IO durumları HMI_IOList kontrolünde gösterilmektedir. **Aktif olan Giriş IOListimizde yeşil olarak görüntülenecektir.**



HMI_RecipeList : Bir reçete sistemi oluşturmanızı sağlar. Reçete satırlarının gösterilip , her satıra tıklandığında O reçete ile ilgili bilgilere erişmenizi sağlar , HMI_TextBox , HMI_NumericBox , HMI_TextList_SelectBox ve HMI_Button kontrolleri ile birlikte kullanılır.

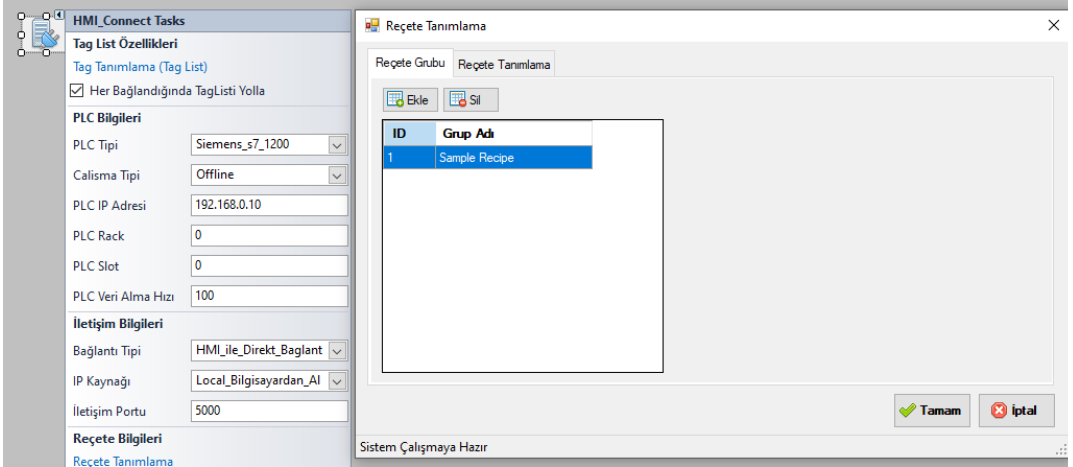


Connect Seçimi : HMI_RecipeList mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Reçete Seç : HMI_Connect nesnesi içerisinde tanımlı Reçete Gruplarından biri Seçilir.

***Bu kontrolün kullanılması için mutlaka HMI_Connect içerisinde reçete tanımlaması yapılmalıdır. , HMI_Connect konusundaki Reçete bölümünü lütfen tekrar okuyunuz.**

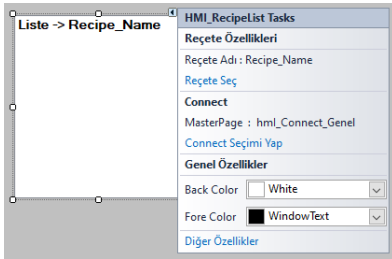
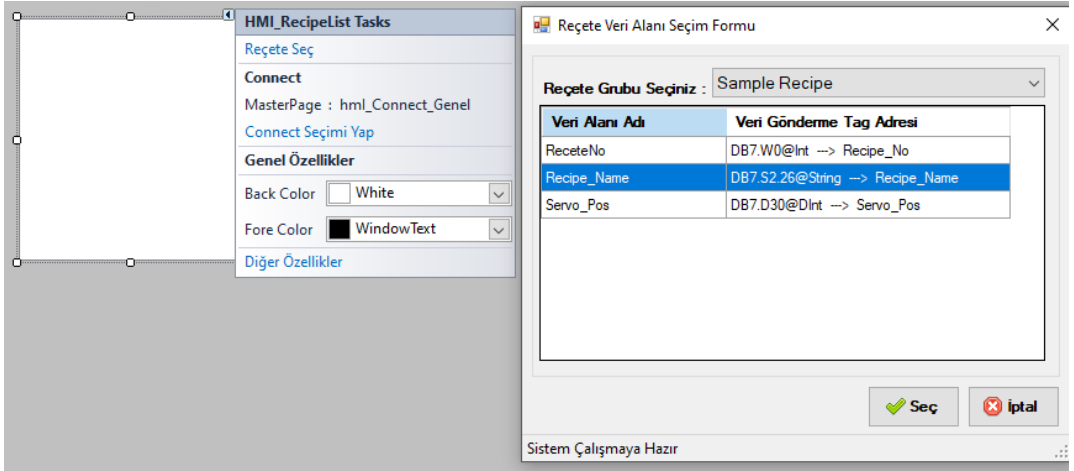
HMI_Connect kontrolü içerisinde Sample Recipe tanımlanmıştır ve bu reçete grubuna ait 3 alan tanımlanıp bu alanlar ilgili Tag adreslerine bağlanmıştır.



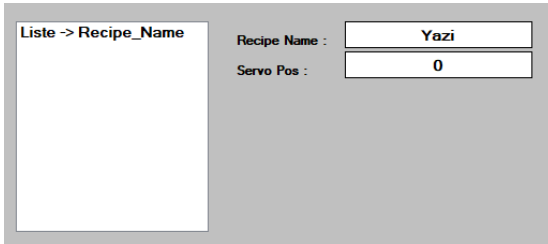
HMI_Connect kontrolü içerisinde Sample Recipe tanımlanır ve bu reçete grubunun da 3 adet Alanı vardır. Bu alanlarda PLC deki 3 Tag adresine bağlanmıştır.

Örneğin 10 adet reçete tanımlandı ve biri geçerli reçete yapılmak istendiğinde , bu 10 adet reçeteden seçili reçetenin verileri , bu tag adreslerine yolların.

Recipe Name Seçilir ve Seç Butonuna Tıklanır. HMI_RecipeList Kontrolündeki Listelenen reçetelerin Hangi Alan ile çağrılacağını , O satırlarda hangi alanın görüntüleneceğini belirtmemiz gerekir. Yani satırlarda hangi alan listelenecek , bu örnekte , **Recipe Name seçilmiştir.**



Seçim yapıldıktan sonra kontrol üzerinde Recipe_Name yazmaktadır. Yani kontrol üzerinde reçetelerin Recipe_Name leri listelenmektedir.



Bu reçete grubunun 3 adet alanı vardır.

- ReceteNo (Özel Alan, otomatik tanımlanır)
- Recipe_Name
- Servo_Pos

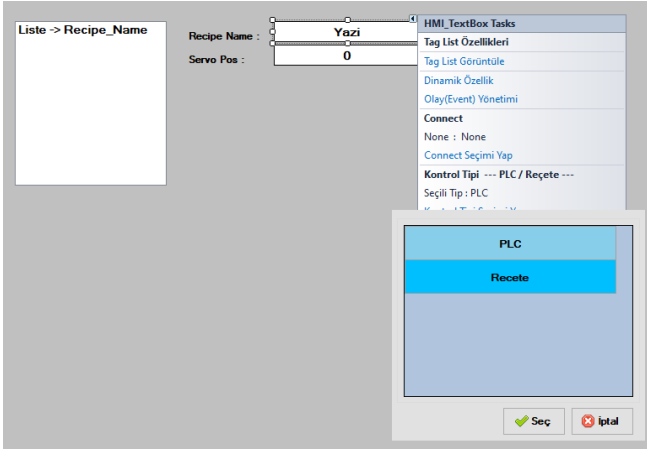
Her bir reçete eklendiğinde , eklenen reçetenin veri alanlarını temsil eder.

HMI_RecipeList de listelenen her bir reçete satırına tıkladığımızda , o reçeteye ait diğer veri alanlarının görüntülenip değiştirilebilmesi için yandaki gibi , o veri alanlarına ait iki kontrol ekliyoruz.

Bu örnekte :

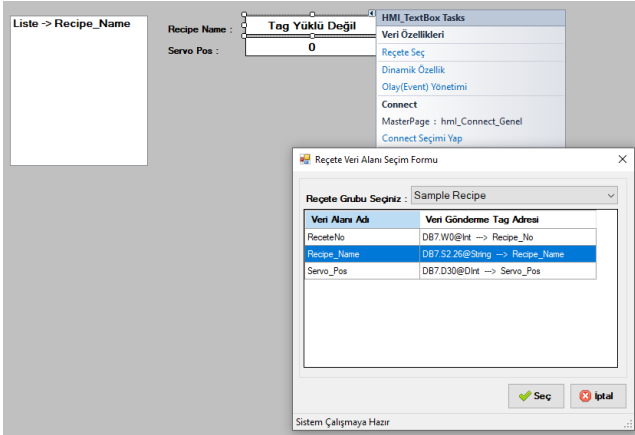
- Recipe_Name için HMI_TextBox
- Servo_Pos için HMI_NumericBox

Eklenmiştir.



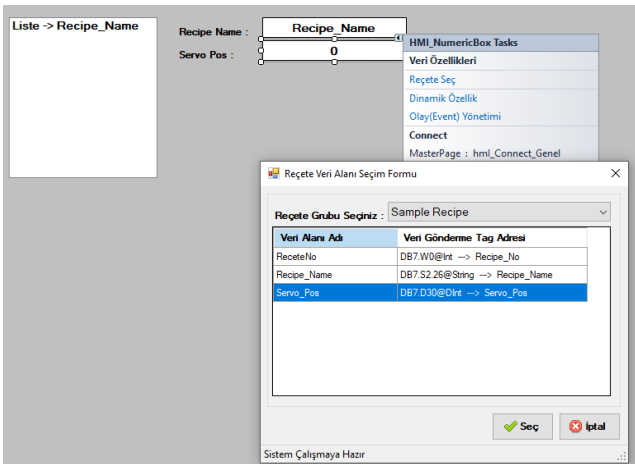
Heri iki kontrol içinde **Kontrol Tipi** Recete olarak değiştirilmelidir.

Connect Seçimi Yapmayı unutmayınız.



Kontrolün Reçete Seç Özelliğine tıklayınız , gelen ekrandan , Reçete Grubunu seçiniz ve bu kontrolün bağlanacağı Veri Alanını Seçiniz.

Bu örnekte : HMI_TextBox Recipe_Name ile bağlanmıştır.



Aynı işlemi Servo_Pos içinde yapıyoruz.

Kontrol Tipini Recete olarak değiştirmeyi unutmayınız.

Connect Seçimi Yapmayı unutmayınız

Liste -> Recipe_Name

Recipe Name :

Servo Pos :

New Recipe

Delete Recipe

Save Recipe

Make Current Recipe

Yandaki gibi HMI_RecipeList in altına 4 adet HMI_Button nesnesi ekleyiniz.

- Yeni Reçete
- Reçete Sil
- Reçete Kaydet
- Geçerli Reçete Yap

Her bir button için Connect tanımlaması yapınız.

Evet Olay Yönetiminden İlgili Reçete Komutunu seçip gerekli düzenlemeyi yapacağız.

HMI_TextBox ve HMI_NumericBox un daha estetik gözükmesi için yandaki gibi GroupBox içerisine alabilirsiniz

Yeni Reçete Ekleme İçin “New Recipe” HMI_Buttonunun Event Olay Yönetim Özelliğine Giriniz (Buttonu seçiniz , sağ üstteki kulakçığı tıklayıp , Event Olay Yönetim Özelliğine Giriniz)

Daha sonra aşağıdaki gibi Komut Listesinden Recete_Yeni Komutunu Seçiniz ve Ekleyiniz
Bu Komut İle Parametre olarak seçilen reçete grubuna yeni reçete eklenir.

Olay (Event) Yonetimi

Olay Listesi

- OnClick
- OnTouchPress
- OnTouchDown
- OnTouchUp

Tanımlı Komut Listesi

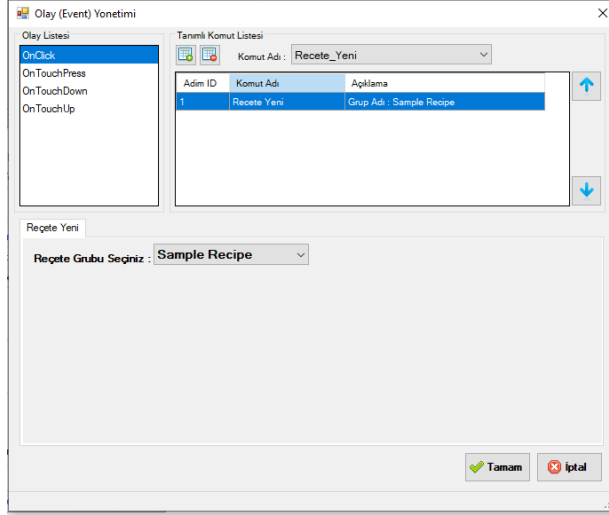
Komut Adı : None

Adım ID	Komut Adı
	None
	Set_Bit
	Reset_Bit
	Set_Reset_Bit
	Set_Bit_When_Touch_Press
	Toggle_Bit
	Dil_Degistir
	Login
	LogOut
	Veri_Gonder
	Form_Ac
	Bekleme
	Mesaj_Goruntule
	Makro_Calistir
	if_Kosul
	Adima_Git
	Adim_Sonu
	Recete_Kaydet
	Recete_Yeni
	Recete_Sil
	Recete_Gonder
	SQL_Sorgu_Yolla

None

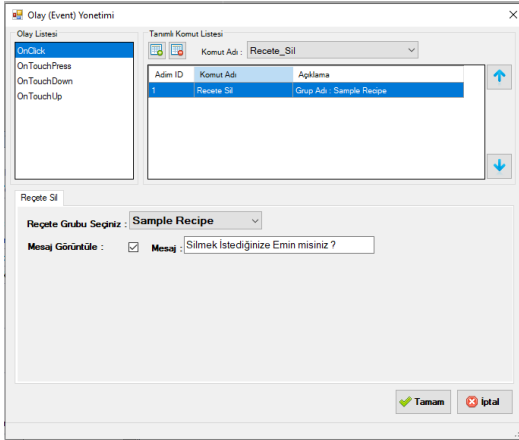
Tamam İptal

Reçete Grubunda seçili Connect kontrolünde tanımlanmış reçeteler listelenir , ilgili reçete grubunu seçiniz

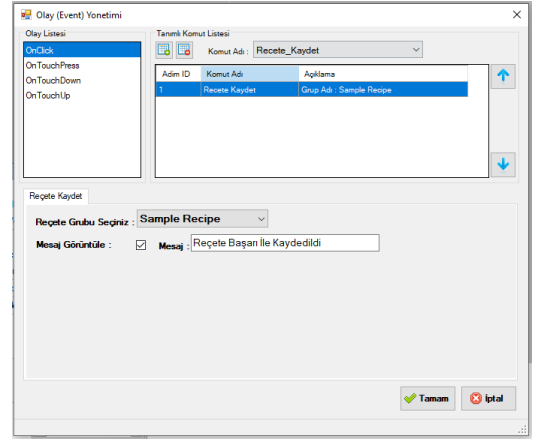


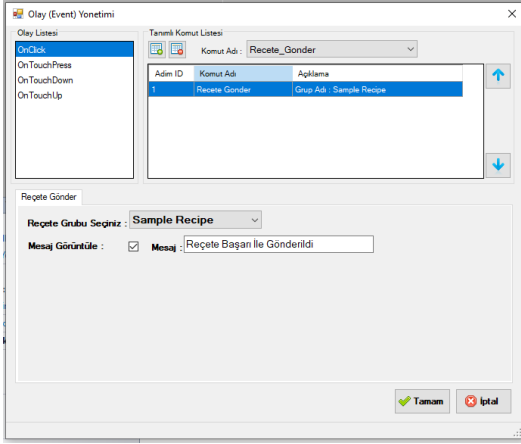
Diğer Butonlar içinde buttonun görevine göre ilgili komut seçilir.

Reçete Sil Buttonu “Delete Recipe”



Reçete Kaydet Buttonu “Save Recipe”





Geçerli Reçete Yap "Make Current Recipe" için ise Recete_Gonder komutu eklenir.

Bu komut ile , buttona tıklandığında HMI_RecipeList deki seçili reçetenin verileri , o reçete grubunun , veri alanlarının bağlandığı tag adreslerine gönderilir.

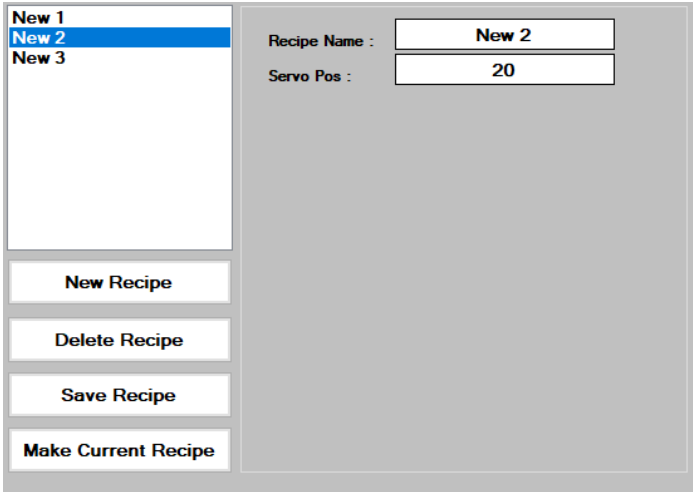
Örneğin 10 adet reçetemiz var ve HMI_RecipeList de Reçete 4 seçili , Bu reçete de Servo_Pos değeri ise 25

Bu buttona tıklandığında aşağıdaki alanlara karşılığındaki veriler gönderilir.

Servo_Pos [DB7.D30@Dint](#) : 25
Recipe_Name [DB7.S2.26@String](#) : Reçete 4

Verileri gönderilir.

Projemizi çalıştırıp aşağıdaki gibi 3 adet reçete eklemesi yaptık. Aşağıda **seçili reçete** New 2 dir. "Make Current Recipe" HMI_Buttununa tıklanırsa Bu satırın verileri , yani "New 2" ve 20 değeri , o reçete grubunun , veri alanlarının bağlandığı tag adreslerine gönderilir.

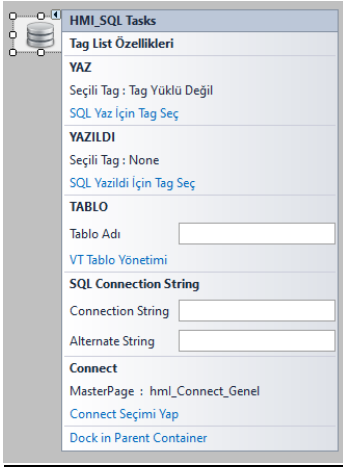


HMI_SQL : Geniş bir SQL Erişim Sisteminin parçasıdır , Olay Komut ları ile geniş bir kullanım alanı sunar. Tek başına kullanılacağı gibi , HMI_Button , HMI_Triger , DataGridView ile birlikte kullanılabilir.

Tek başına kullanıldığında ; PLC den gelen Yaz komutu ile , tanımlı tabloya (Tablo Adı) Satır ekler. Tablo Sütunları ise **VT Tablo Yönetimi** Özelliğinde Tanımlanır. SQL Parametreleri (Mssql server adresi , hangi Database) Connection String ile tanımlanır. Tablo DataBase içerisinde yoksa bile o tabloyu oluşturur , tablo varsa tabloya , PLC den aldığı verileri satır olarak ekler.

***MSSQL içerisinde mutlaka DataBase oluşturulmalıdır. Tablo oluşturmanıza gerek yok , sistem tabloyu otomatik oluşturur**

Mutlaka MasterPage Formuna Eklenmelidir. Diğer Formlarda Kullanılmaz. MasterPage e eklendikten sonra , Diğer formlardan da erişilebilir.



Connect Seçimi : HMI_SQL mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Yaz : PLC den gelen veri yaz komutunun tag adresi buradan seçilir.

Yazıldı : PLC e yazıldı geri bildirimini yapılır.

Tablo Adı : Arada boşluk bırakmadan tablo adı yazınız. Eğer tablo DataBase içerisinde yoksa otomatik oluşturur.

VT Tablo Yönetimi : Tablo sütunları ile Bu Sütunların PLC de bağlandığı Tag adresleri burada tanımlanır. Bu tanımlamaları yaptıktan sonra yeni sütun tanımlarsanız , o sütun yeni sütun olarak tabloya eklenir.

Connection String : Database bağlantı parametreleri ile DataBase adı burada belirlenir.

Alternate String : İlk Connection String ile bağlantı gerçekleşmez ise bu parametre ile bağlantı sağlanır. Projenin yazıldığı bilgisayar ile çalıştığı bilgisayar farklı ise kullanışlı bir özelliktir. İlkinde çalıştığı bilgisayarın ikincisinde ise yazıldığı bilgisayarın MSSQL Connection Stringi yazılır.

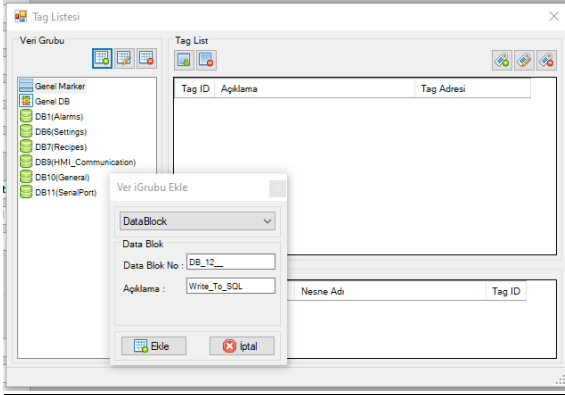
Connection String e kısaca değinilecektir.

DB12(Write_To_SQL)			
	Name	Data type	Offset
1	Static		
2	Write	Bool	0.0
3	Written	Bool	0.1
4	Record_Date	DTL	2.0
5	Flow_Value	Real	14.0
6	Pressure_Value	Real	18.0
7	Operator_ID	DInt	22.0

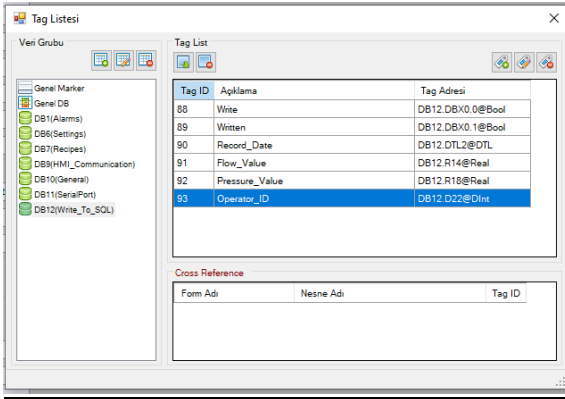
Öncelikle yandaki gibi PLC içerisinde DataBlock tanımlayınız. **DataBlock un Optimized Block Access özelliğini kaldırmayı unutmayınız.**

Bu Örneğimizde DB12(Write_To_SQL) Data Blocku tanımlanmıştır. Write ve Written dışındakiler , Rapor da kayıt altına alınacak Adreslerdir.

Bu adreslerin değerleri , her Write komutu geldiğinde ilgili tabloya satır olarak eklenecektir.



MasterPage üzerindeki HMI_Conect nesnesine DB12(Write_To_SQL) Data Blocku nu Eklıyoruz

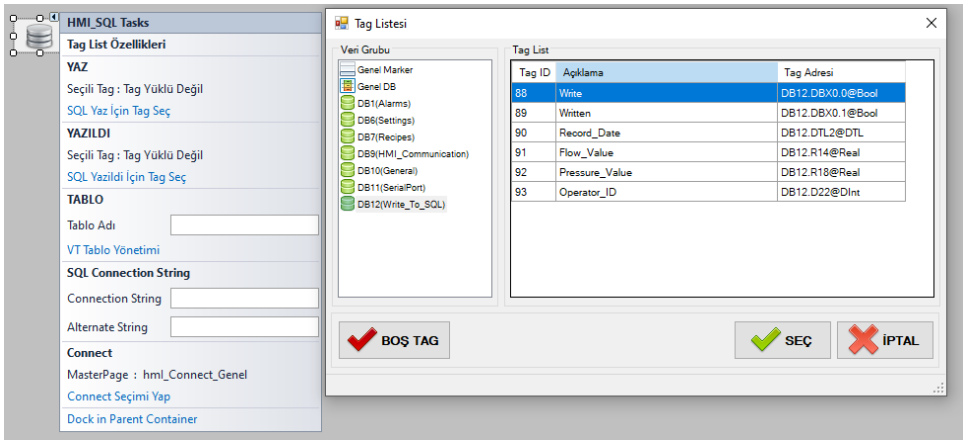


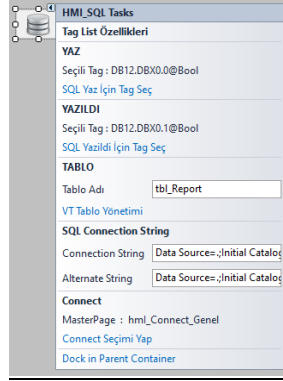
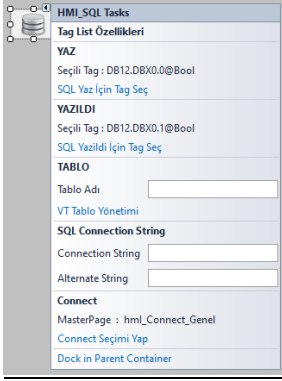
DB12(Write_To_SQL) Data Blockunun Taglarını TagList e ekliyoruz.

İsterseniz tek tek , isterseniz HMI_Connect konusunda anlattığımız gibi topluca ekleyebilirsiniz.

Topluca ekledikten sonra mutlaka kontrol ediniz.

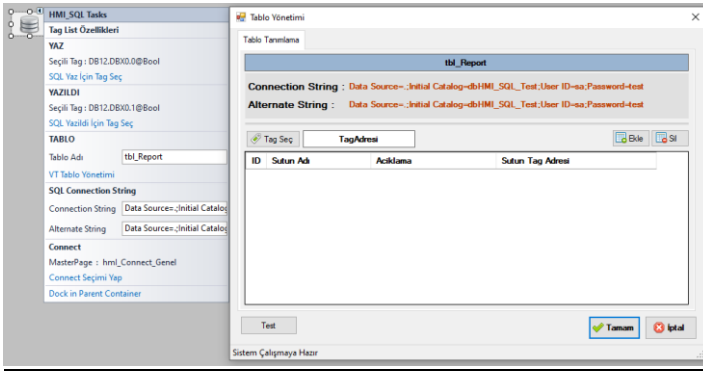
HMI_SQL in komut adreslerini giriyoruz. Yaz komutu için seçtiğimiz tag adresinin değeri true olduğunda (yüksele kenarında) VT Tablo Yönetiminde tanımladığımız adreslerin değerleri , ilgili tabloya (Tablo Adı ile tanımladığımız) satır olarak eklenecektir.





Yaz ve Yazıldı için Tag adresi seçimi yapılmıştır.

Tablo adı ve Connection String parametreleri girilmiştir.



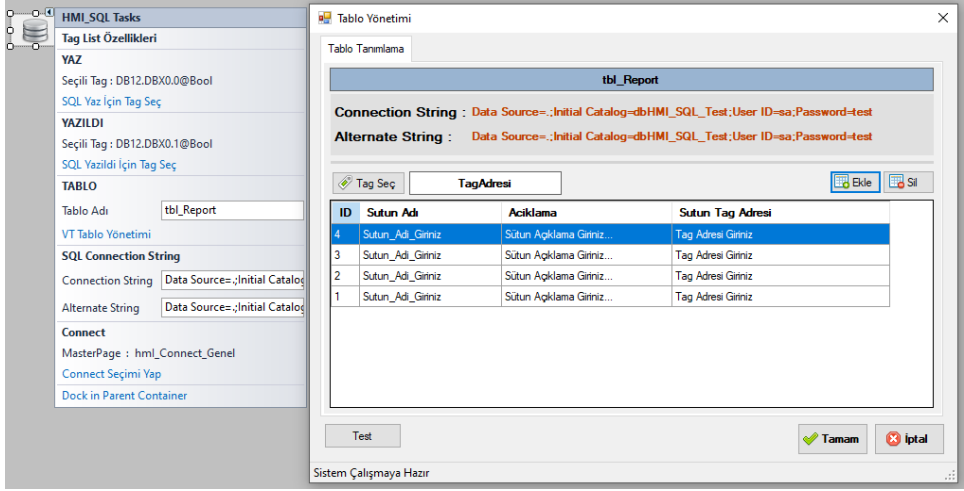
VT Tablo Yönetimi Özeliği İle

Tablonun Sütunları tanımlanır ,

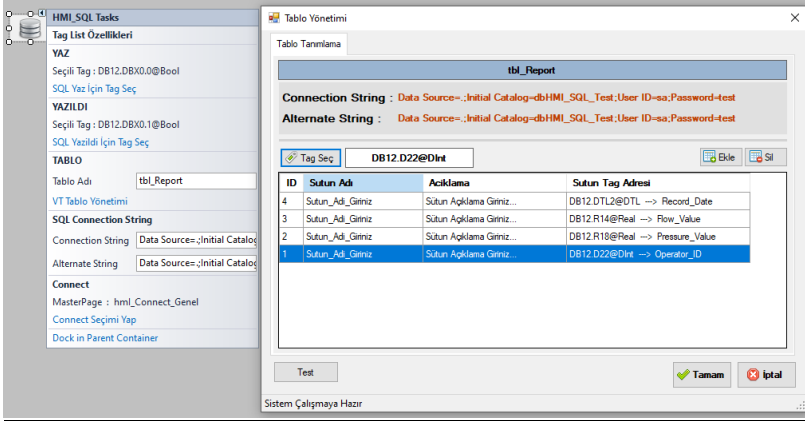
Bu Sütunların değerini alacağı Tag Adresleri Bağlanır

Eğer bu tablonun değerleri bir DataGridView kontrolünde gösterilecekse , Sütun Başlıklarında gösterilmesi için Açıklama Yazılır yani açıklama da yazdığımız DataGridView Kontrolündeki Sütun Başlığıdır.

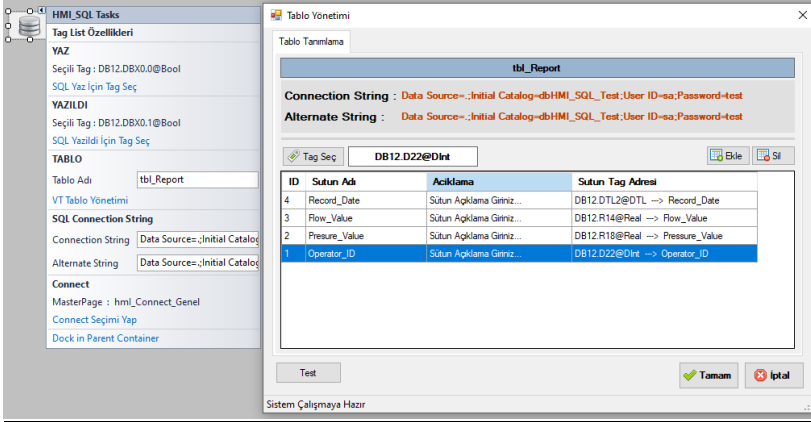
Aşağıda 4 adet sütun tanımlanmıştır.



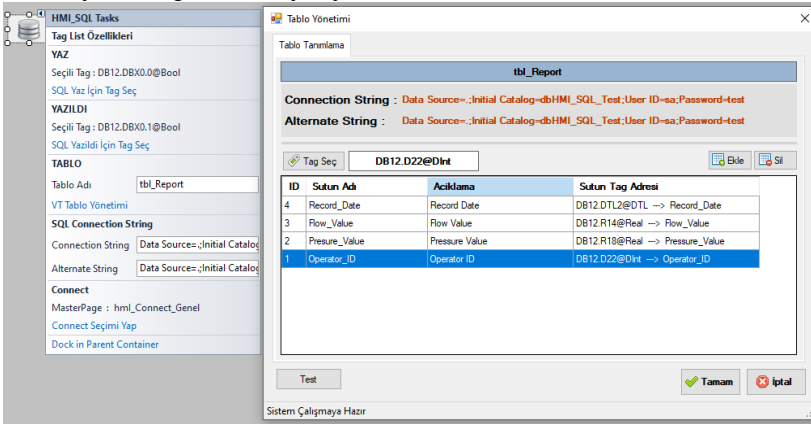
İlk Olarak bu sütunlar için tag adresi ataması yaptık , Sütun Adlarını bu tag adreslerinin adlarına göre yapacağız.



Daha Sonra aşağıdaki gibi sütun adlarını giriyoruz , metin girişi için ilgili hücreye çift tıklayınız.



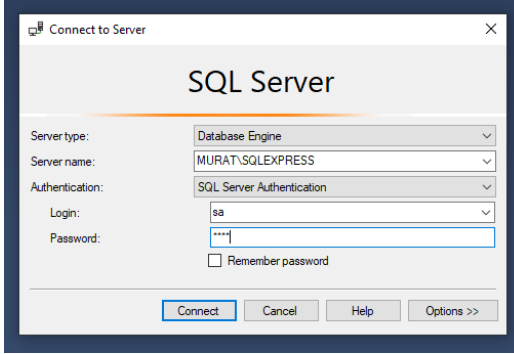
Aşağıda ise açıklama girişi yapılmaktadır. Eğer bu tablonun değerleri bir DataGridView kontrolünde gösterilecekse , Sütun Başlıklarında gösterilmesi için Açıklama Yazılır.



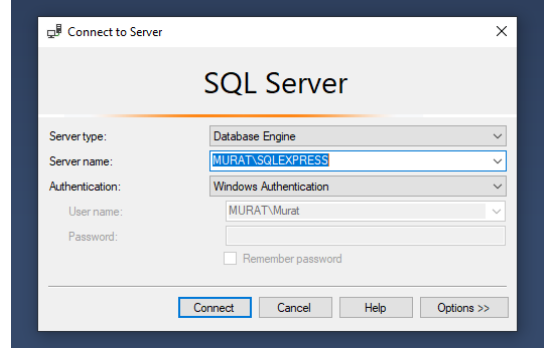
Connection String : Database bağlantı parametreleri ile DataBase adı burada belirlenir. Kullanıcı adı ve şifre ile veri tabanına giriş sağlayabileceğiniz gibi , Windows kullanıcısı olarak da bağlanabilirsiniz.

Data Source veya Server Name i öğrenmek için Managment Studio yu açınız oradan Server Name yi öğrenebilirsiniz.

Parametreleri aşağıdaki gibi test edebilirsiniz
sa kullanıcısi ile giriş



Windows kullanıcısı ile kullanıcı adı ve şifresi olmadan giriş



Örnek 1:

Data Source=.;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source=.; Programın çalıştığı Bilgisayarda yüklü MSSQL e bağlanır.
Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır
User ID=sa; kullanıcı adı : sa
Password=test şifre ise test dir

Örnek 2 :

Data Source= MURAT\SQLEXPRESS;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source= MURAT\SQLEXPRESS; MURAT Bilgisayarındaki Adı SQLEXPRESS olan MSSQL e bağlanır
Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır
User ID=sa; kullanıcı adı : sa
Password=test şifre ise test dir

Örnek 3 :

Data Source= 192.168.0.5\SQLEXPRESS;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source= 192.168.0.5\SQLEXPRESS; 192.168.0.5 Bilgisayarındaki Adı SQLEXPRESS olan MSSQL e bağlanır , IP ile bağlanmada gerekli port ayarlarında yapılmalıdır. Daha fazla bilgi için MSSQL e Uzak Bağlantı İle Bağlanmayı araştırınız.

Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır
User ID=sa; kullanıcı adı : sa
Password=test şifre ise test dir

Örnek 4:

Server=localhost;Database=db_HMI_SQL;Integrated Security=True;

Server=localhost;Programın çalıştığı Bilgisayarda yüklü MSSQL e bağlanır.

Database=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır

Integrated Security=True; Windows Kullanıcısı ile MSSQL e giriş yapılıyor , Bu girişte kullanıcı adı ve şifreye gerek duyulmaz.

Bilgisayara Admin yetkisine sahip kullanıcı ile giriş yapılmalıdır.

Aşağıda Connection String değiştirilmiştir ve Windows Kullanıcısı ile MSSQL e giriş yapılacak şekilde ayarlanmıştır.

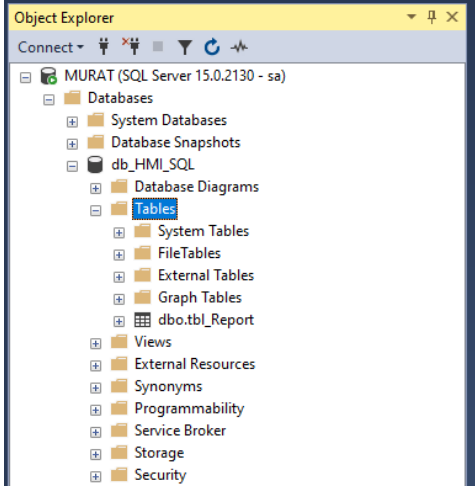
The screenshot shows two windows. The left window is titled 'HMI SQL Tasks' and displays the configuration for the 'tbl_Report' table. It includes sections for 'Tag List Özellikleri', 'YAZ', 'YAZILDI', 'TABLO', and 'SQL Connection String'. The 'SQL Connection String' is set to 'Server=localhost;Database=db_HMI_SQL;Integrated Security=True;'. The 'Connect' button is visible. The right window is titled 'Tablo Yönetimi' and shows the table definition for 'tbl_Report'. It includes a table with columns: ID, Sütun Adı, Açıklama, and Sütun Tag Adresi. The table has 4 rows: 1. Operator_ID, 2. Pressure_Value, 3. Flow_Value, 4. Record_Date. The 'Connection String' and 'Alternate String' are both set to 'Server=localhost;Database=db_HMI_SQL;Integrated Security=True;'. The 'Tag Seç' dropdown is set to 'DB12.DTL2@DTL'. The 'Test' button is visible at the bottom.

The screenshot shows the 'Object Explorer' window in SQL Server Enterprise Manager. The 'Connect' bar shows 'MURAT (SQL Server 15.0.2130 - sa)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', and 'db_HMI_SQL'. The 'db_HMI_SQL' database is selected, and the 'Tables' folder is expanded, showing the 'tbl_Report' table.

Yandaki gibi , mutlak MSSQL içerisinde db_HMI_SQL adında bir DataBase tanımlayınız.

Programı çalıştırdığınızda , parametreler doğru girilmişse tablo ve sütunları , Tag Adreslerinin tiplerine göre tanımlanır.

Programı çalıştırdığınızda tablonun otomatik olarak oluşturulduğunu göreceksiniz.



Tablo sütunları da Tag Adreslerinin tipine göre otomatik olarak oluşturulur.

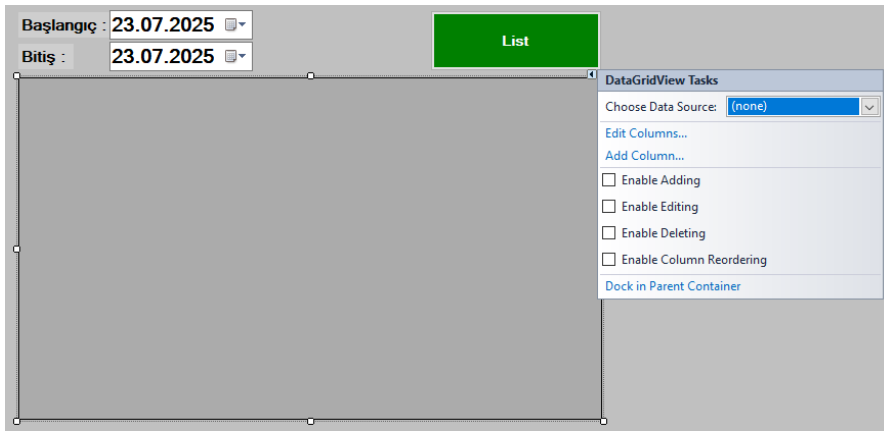
Column Name	Data Type	Allow Nulls
ID	bigint	<input type="checkbox"/>
Operator_ID	bigint	<input type="checkbox"/>
Pressure_Value	float	<input type="checkbox"/>
Flow_Value	float	<input type="checkbox"/>
Record_Date	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

Event Olay Yönetimi İle SQL İşlemleri : HMI_Button ve HMI_Trigger daki Event Olay Yönetimi ile , kod gerektirmeden SQL sorgu yollayabilir , DataGridView de görüntülenmesini sağlayabilirsiniz. NikCom_HMI bunun için size geniş bir kütüphane sunmaktadır.

Bu işlemler için MasterPage da tanımlanmış bir HMI_SQL kontrolüne ihtiyacınız vardır. Bu nesne üzerinden Veri Tabanına bağlanılacaktır.

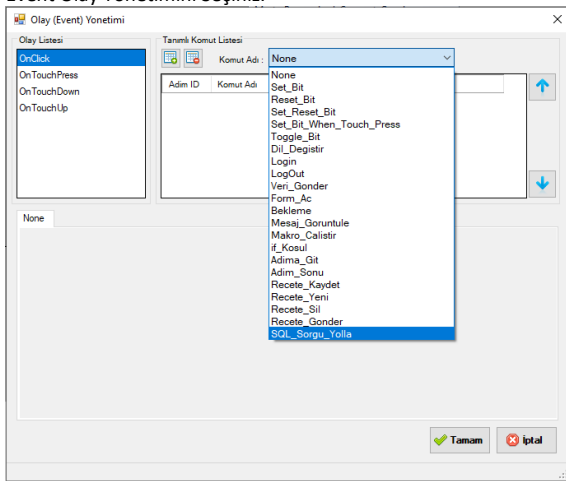
Aşağıdaki gibi Form üzerine kontrolleri yerleştiriniz.

- 2 Adet HMI_Label
- 2 Adet DateTimePicker , dateTimePicker_Baslangic ve dateTimePicker_Bitis
- 1 Adet HMI_Button , Text Düzenle Özelliğine "List" yazınız.
- 1 Adet DataGridView , ekranda görüldüğü gibi Enable Adding , Enable Editing , Enable Deleting özelliklerindeki seçimi kaldırınız



HMI_Buttonu seçiniz ve üst sağdaki kulakçığı tıklayınız , açılan pencereden Event Olay Yönetimini Seçiniz.

Gelen ekrandan (yandaki ekran) komut listesinden SQL_Sorgu_Yolla komutunu çalıştırınız.



Aşağıdaki Ekranı açıklayalım.

Sorgu Tipi ile hangi sorguyu yollayacağınızı belirleyebilirsiniz.

- Select Sorgusu
- Ekleme Sorgusu
- Silme Sorgusu
- Genel Sorgu : Bu sorgu tipi ile kendi kompleks sorgularınızı rahatlıkla Mssql e yollayabilirsiniz.

DataGrid Seç : Sorgunuz eğer bir Select sorgusu ise bu sorgu sonucunun görüntüleneceği bir DataGridView kontrolü mutlaka olmalıdır.

SQL Seçimi : Bu komut mutlaka bir HMI_SQL kontrolüne ihtiyaç duyar. Bağlantı parametrelerini bu kontrolden alır , Genel Sorgu dışındaki sorgularda işlem yapacağı tablo ve sütun bilgilerini bu kontrolden alır.

Genel Sorgu HMI_SQL Kontrolünün bağlantı parametrelerini kullanır ama HMI_SQL in tablosuna erişmek zorunda değildir.

Editör Ekranı : Kullandığınız sorgunu ilgili editörleri bu ekranda görüntülenir.

Parametre Bilgileri : Sorgularda kullanmak için 8 adet parametre kullanabilirsini , bu parametreler 0 ile 7 arasında numarandırılır. Parametre veri kaynağı Tag adresi olabileceği gibi Form üzerindeki tanımlı bir kontrolde olabilir (DateTimePicker , TextBox gibi)

Olay (Event) Yonetimi

Olay Listesi

- OnClick
- OnTouchPress
- OnTouchDown
- OnTouchUp

Tanımlı Komut Listesi

Komut Adı : SQL_Sorgu_Yolla

Adım ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Select_Sorgusu

SQL Sorgu Yolla

Sorgu Tipi Seçiniz: Select_Sorgusu

DataGrid Seç: DataGrid Seçiniz...

SQL Seçiniz: **Seç** **SQL Seçiniz**

Select Editör

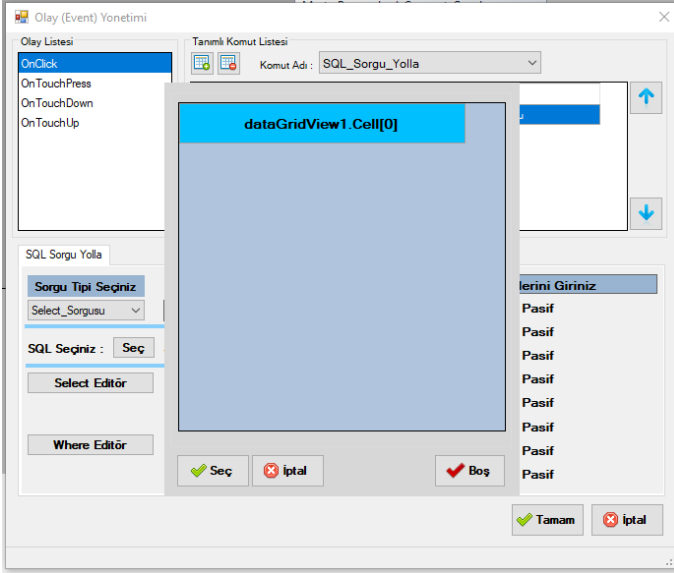
Where Editör

Parametre Bilgilerini Giriniz

Parametre	Statü	Seçim	Statü
{0}	Pasif	Seç	Pasif
{1}	Pasif	Seç	Pasif
{2}	Pasif	Seç	Pasif
{3}	Pasif	Seç	Pasif
{4}	Pasif	Seç	Pasif
{5}	Pasif	Seç	Pasif
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

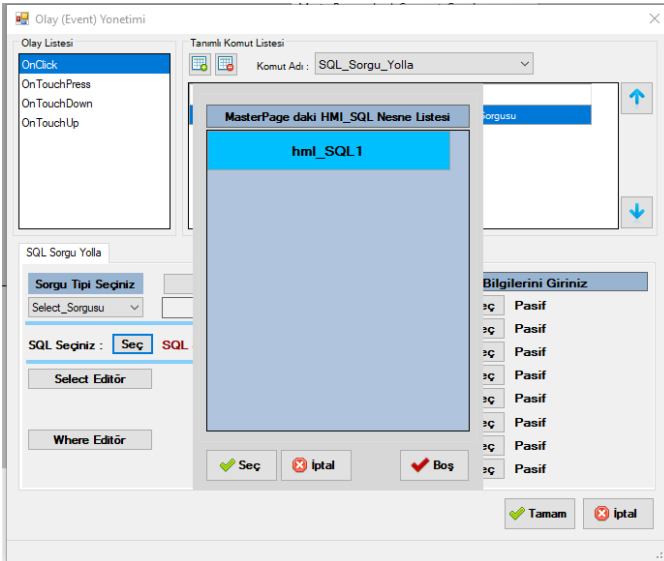
Tamam İptal

Select Sorgusu ile Başlayalım :



Öncelikle Select sorgumuz için bir DataGridView kontrolü tanımlayalım.

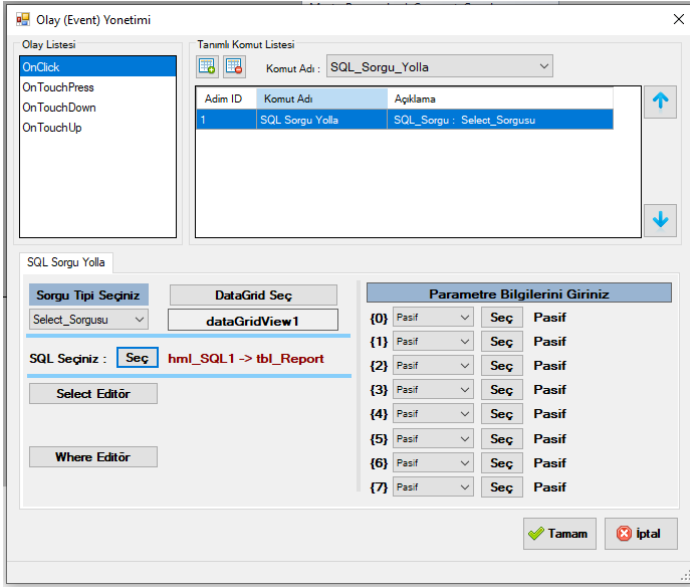
Form üzerine yerleştirdiğimiz DataGridView kontrolünü Seçiyoruz.



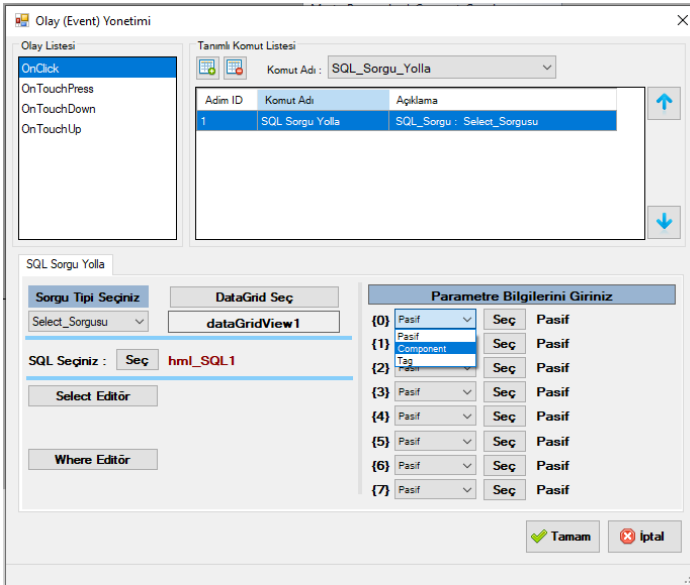
Daha sonra ise MSSQL ile bağlantımızı sağlayacak HMI_SQL nesnesini seçiyoruz.

Select sorgusunda sorgulanacak tablo , HMI_SQL nesnesinde tanımladığımız tablodur.

Başka tablonun sorgulanmasını istiyorsanız Genel_Sorgu yu seçiniz.



Yandaki ekranda gördüğünüz gibi tbl_Report tablosunda Select sorgusu yapacağız.



Artık sorgumuzda kullanacağımız parametreleri seçebiliriz.

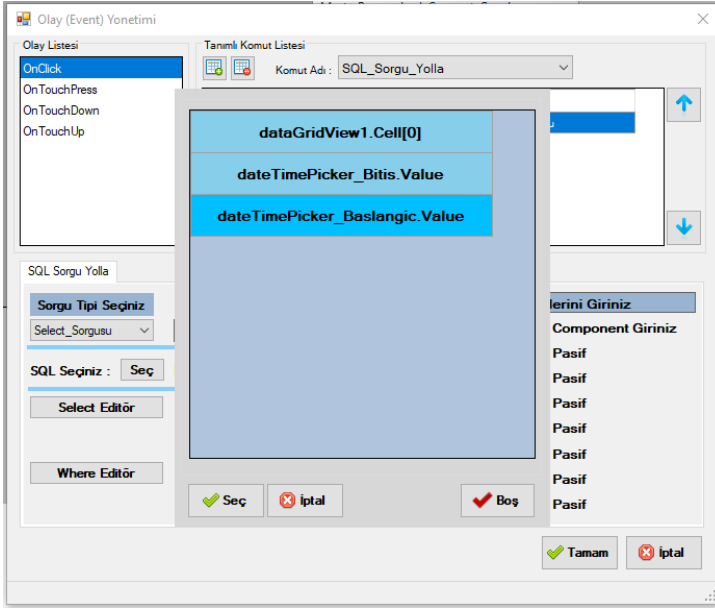
Component veya Tag olarak parametre tanımlaması yapabiliriz.

Sorgumuzu

- dateTimePicker_Baslangic
- dateTimePicker_Bitis

Kontrolerine göre yapacağız.

Bu kontrollerden tarih seçeceğiz , iki tarih aralığındaki satırları getireceğiz

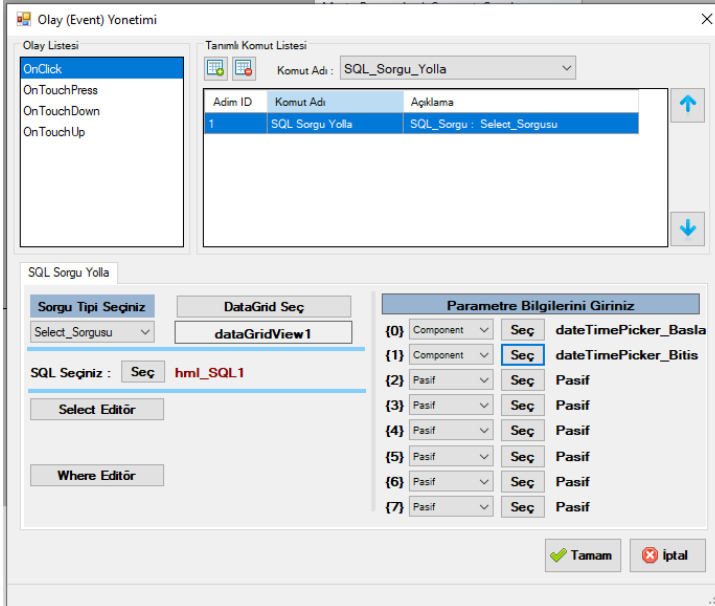


Component seçeneğini seçiniz ve Seç butonuna tıklayınız.

Ekrandaki gibi seçebileceğiniz kontroller ekrana gelir.

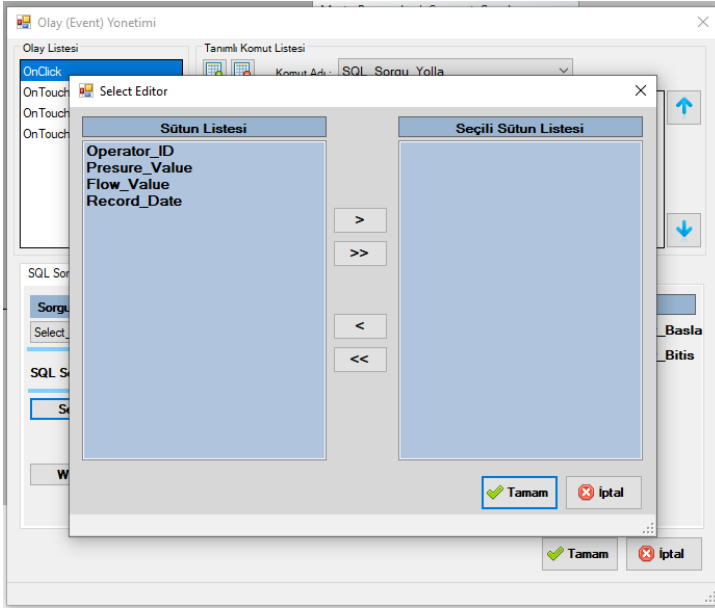
Aşağıdaki gibi 2 adet parametre tanımlayınız.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis

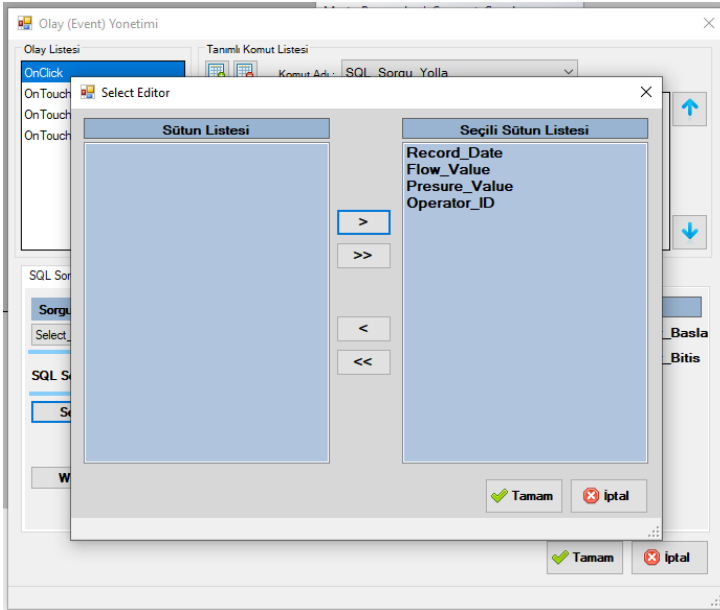


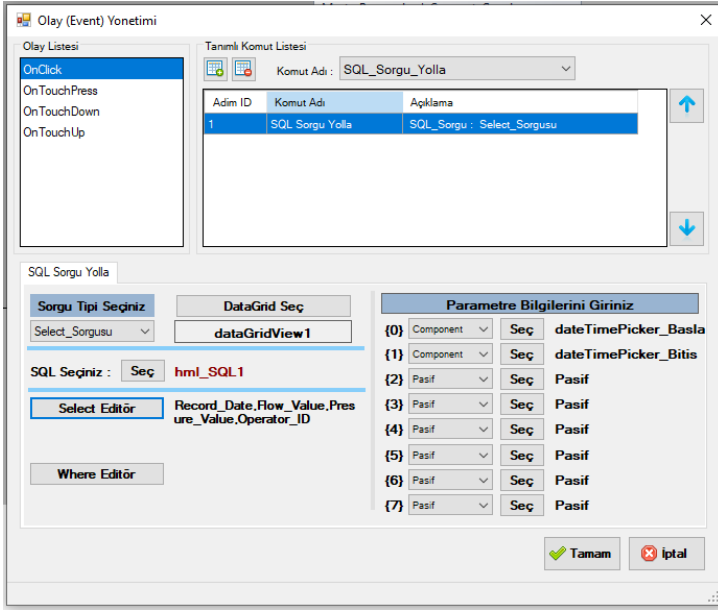
Yanda 2 adet parametre tanımlanmıştır.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis



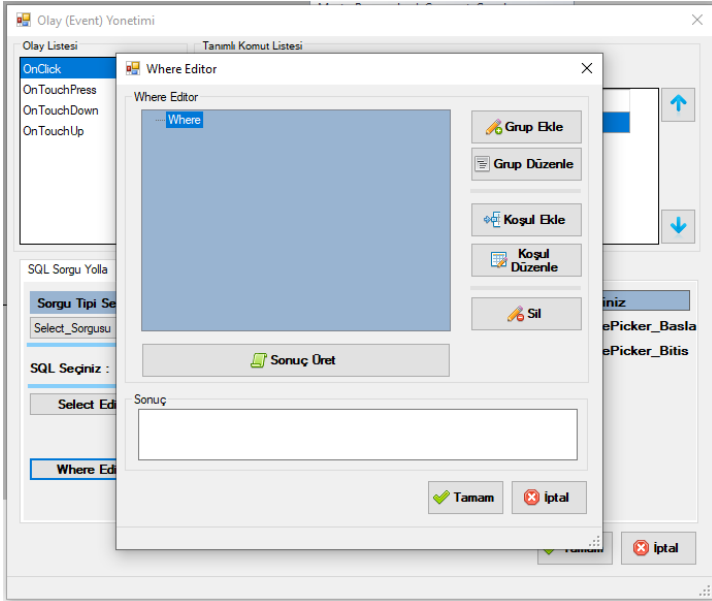
Select Editör Buttonuna tıklayınız , DataGridView de görüntülenmesini istediğiniz sütunları sağ tarafa taşıyınız.





Seçilen 4 sütünü dataGridView1 kontrolünde görüntüleyeceğiz

Bu şekilde bırakırsak tüm satırlar listelenir ama biz Where ile tarih aralığı belirlemek istiyoruz.



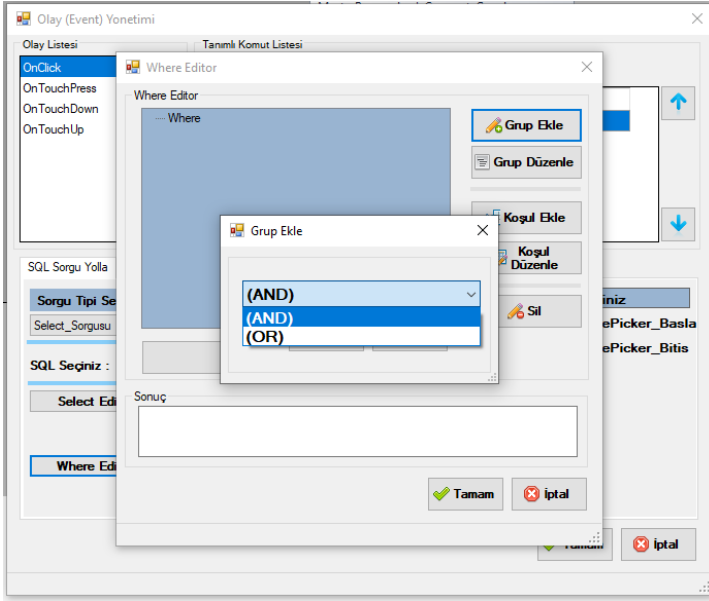
Where Editöre tıklayınız.

Bu editör kompleks Where sorgularını oluşturmamızı sağlar.

Grup Ekle ile AND , OR işlemleri Yapabiliriz

Koşul Ekle ile karşılaştırma komutlarını yazabiliriz.

Sonuç Üret e tıkladığımızda ise sorgumuzun son hali aşağıya yazılır.

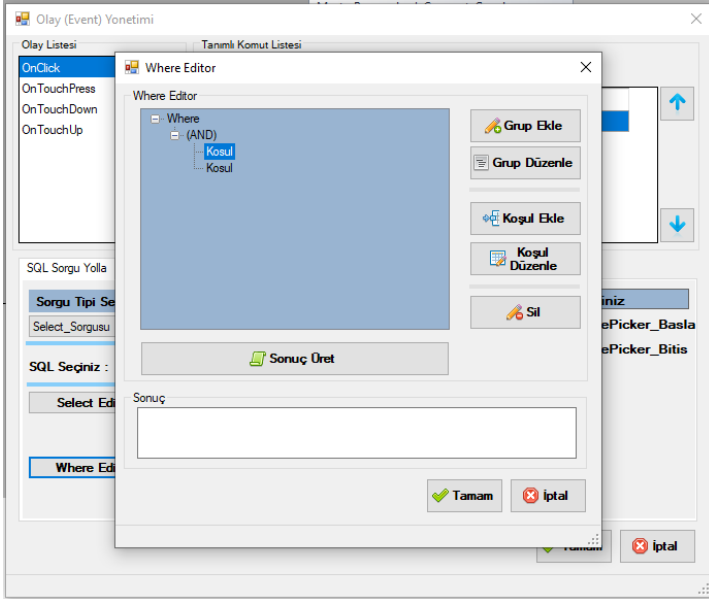


Sorgumuz bir aralık sorgusudur. Başlangıçtan büyük ve Bitişten küçük satırlar listelenecektir.

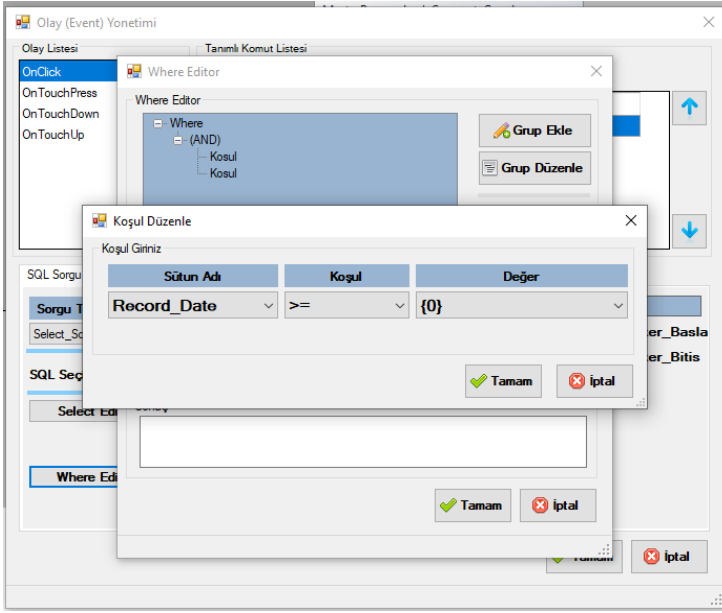
Yani iki koşul ve bu iki koşul da AND ile bağlanacaktır.

Başlangıçtan büyük AND Bitişten küçük

Grup Ekle ile AND operatörünü ekliyoruz ve bu operatöre 2 adet koşul tanımlıyoruz



AND Operatörünü eklediğimizde 2 adet koşul Default olarak gelir. İsterseniz daha fazla koşulu bu operatöre bağlayabilirsiniz.



Koşul seçeneğine **çift tıkladığımızda** , yada Koşul seçeneğini seçip **Koşul Düzenle** butonuna tıkladığımızda ekrana koşul tanımlaması yapacağımız Form gelir.

Yandaki gibi Koşulumuzu yazabiliriz , burada

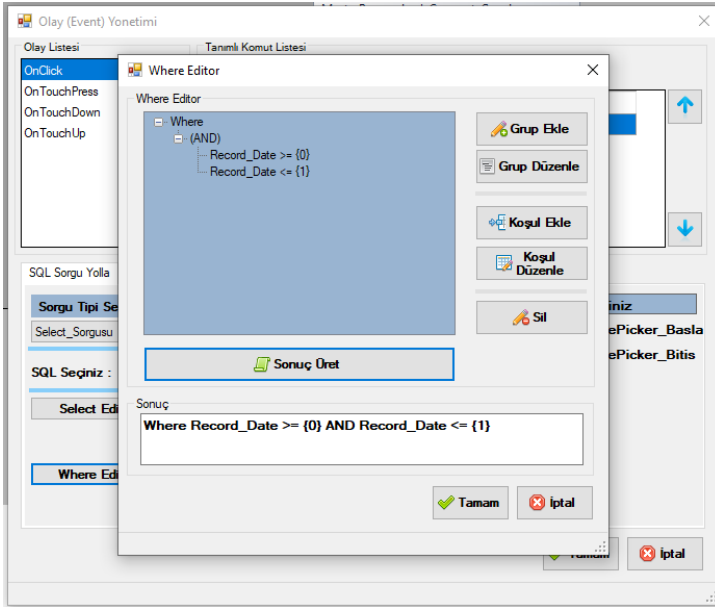
{0} : dateTimePicker_Baslangic

Kontrolüdür.

Record_Date büyük eşit dateTimePicker_Baslangic daki seçili tarihten

İsterseniz Değer bölümünden **Dinamik** Seçimi yapıp kendiniz de yazı ile parametre girişi yapabilirsiniz.

Dinamik girişi kompleks bir yapıdır ve daha sonra anlatılacaktır.



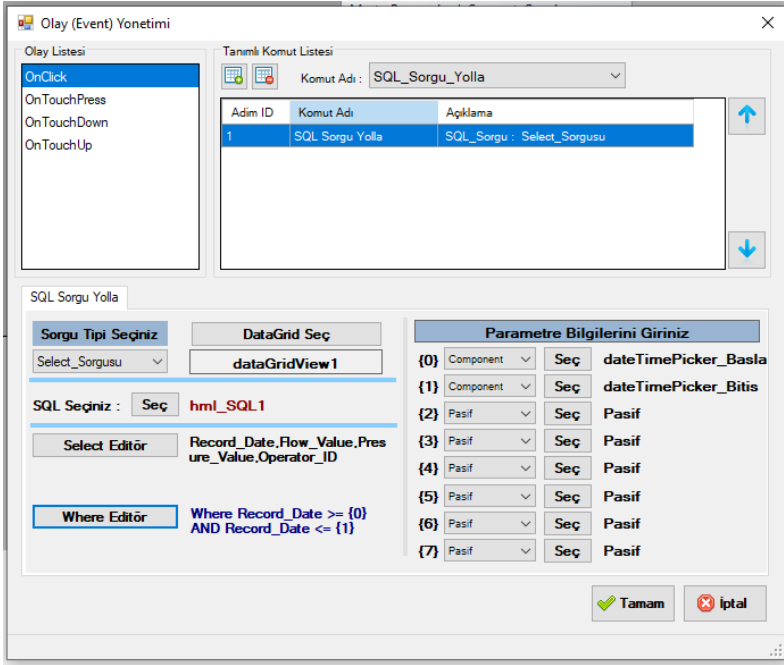
Yandaki ekranda İki Koşul tanımlanmıştır.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis

Sonuç Üret Butonuna tıklayınız ve Where sorgusunu kontrol ediniz.

Where Editör ile çok kompleks sorgular yazabilirsiniz.

Tamam Butonuna tıklayınız ve Event Olay Yönetimi Ekranına geçiniz.



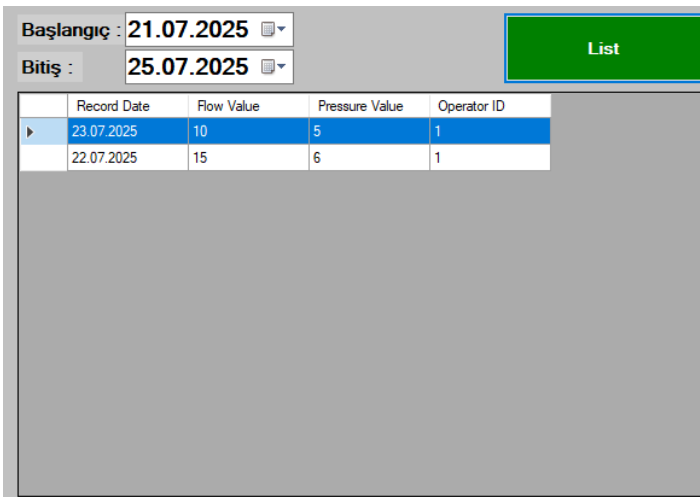
Sorgumuzun tüm girdileri yandaki gibidir.

Tamam Buttonuna tıklatınız.

Artık sorgumuzu kontrol edebiliriz.

ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date
1	1	5	10	2025-07-23 00:00:00
2	1	6	15	2025-07-22 00:00:00

Tablomuz Management Studio u kullanarak yandaki gibi 2 satır ekleyiniz. Tarihler dikat ediniz.



Projemizi çalıştırılm ve yandaki gibi Başlangıç ve Bitiş tarih aralığı verip List HMI_Buttonuna tıklayınız

İlgili Sonuçlar dataGridView1 kontrolünde listelenmiştir.

```
2 references
public partial class Rapor : Form
{
    0 references
    public Rapor()
    {
        InitializeComponent();
    }

    1 reference
    private void Rapor_Load(object sender, EventArgs e)
    {
        dateTimePicker_Baslangic.Value = DateTime.Now;
        dateTimePicker_Bitis.Value = DateTime.Now;
    }
}
```

Eğer DateTimePicker kontrollerinin , Form her yüklendiğinde O günün tarihini almasını istiyorsanız

- Formda boş bir yeri çift tıklayınız
- Gelen Formun Load olayı , yani Rapor_Load olayına yandaki kodu yazınız.
- Büyük küçük harf ayırımına dikkat ediniz.

Ekleme Sorgusu :

Adım ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Ekleme_Sorgusu

Adım ID	Durum	Seçim	Değer
{0}	Tag	Seç	DB12.DTL2@DTL
{1}	Tag	Seç	DB12.R14@Real
{2}	Tag	Seç	DB12.R18@Real
{3}	Tag	Seç	DB12.D22@DInt
{4}	Pasif	Pasif	Pasif
{5}	Pasif	Pasif	Pasif
{6}	Pasif	Pasif	Pasif
{7}	Pasif	Pasif	Pasif

Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

SQL_Sorgu_Yola Komutunu ekleyiniz

Sorgu Tipi : Ekleme_Sorgusu yapınız.

SQL Seçiniz.

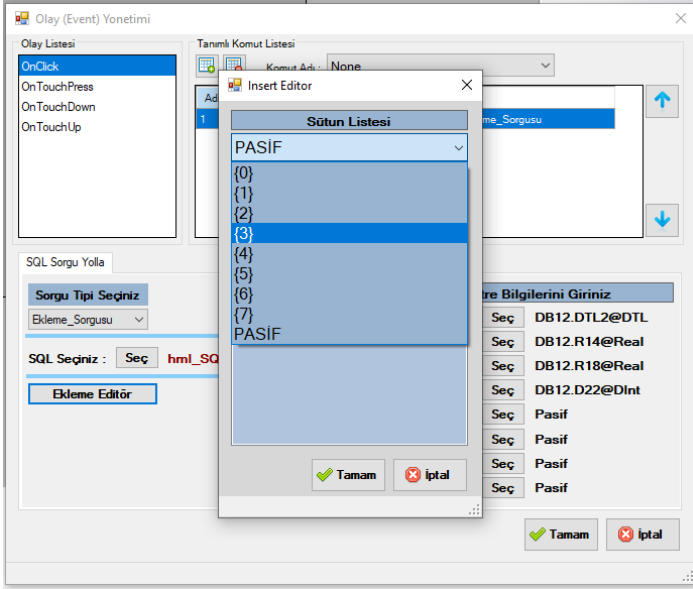
DB12(Write_To_SQL) datablock unda tanımladığımız 4 adet Tag adresini parametre olarak tanımlayınız.

Eklenecak satır , değerlerini bu Tag Adreslerinden alacaktır.

Yani HMI_Buttona tıkladığında , eklenen satırın , alan değerleri , Bu tag adreslerinin değeridir.

Ekleme işleminde hangi sütunun hangi parametre ile eşleşeceğini buradan belirliyoruz.

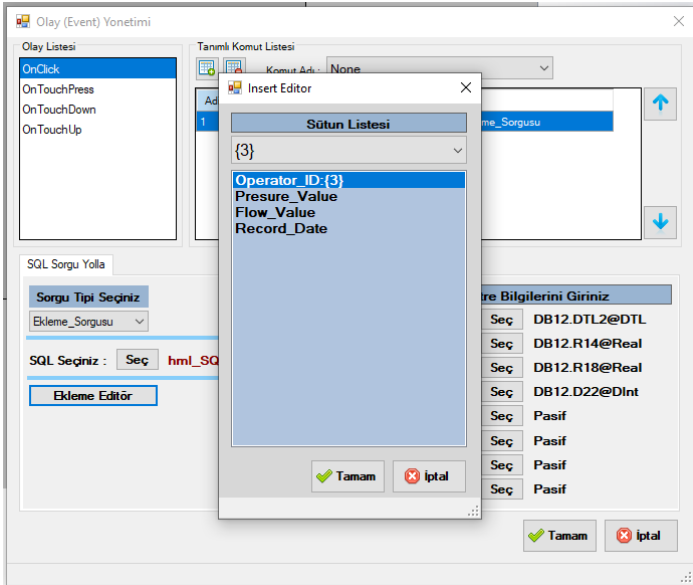
Adım ID	Durum	Seçim	Değer
Seç	DB12.DTL2@DTL		
Seç	DB12.R14@Real		
Seç	DB12.R18@Real		
Seç	DB12.D22@DInt		
Seç	Pasif		
Seç	Pasif		
Seç	Pasif		
Seç	Pasif		



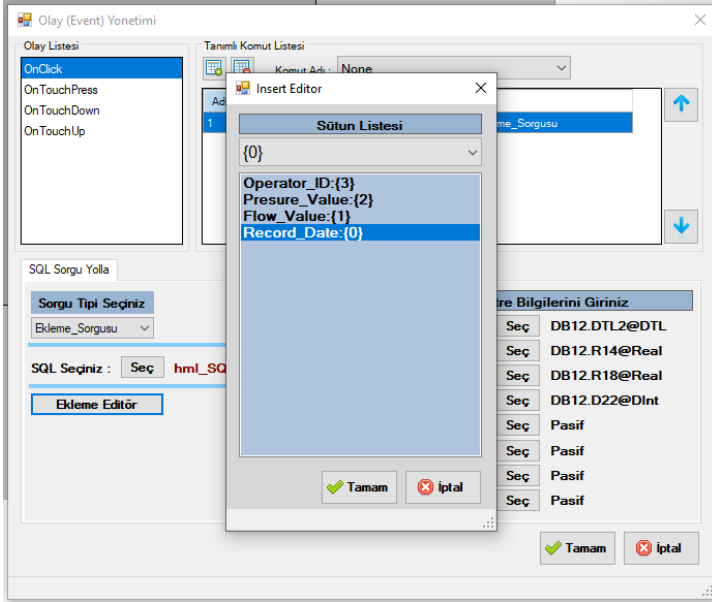
Yandaki ekranda Operator_ID sütunu ile {3} parametresini eşleştirdik.

Yani Her satır eklendiğinde , satırın Operator_ID değeri , {3} nolu parametrenin değeridir.

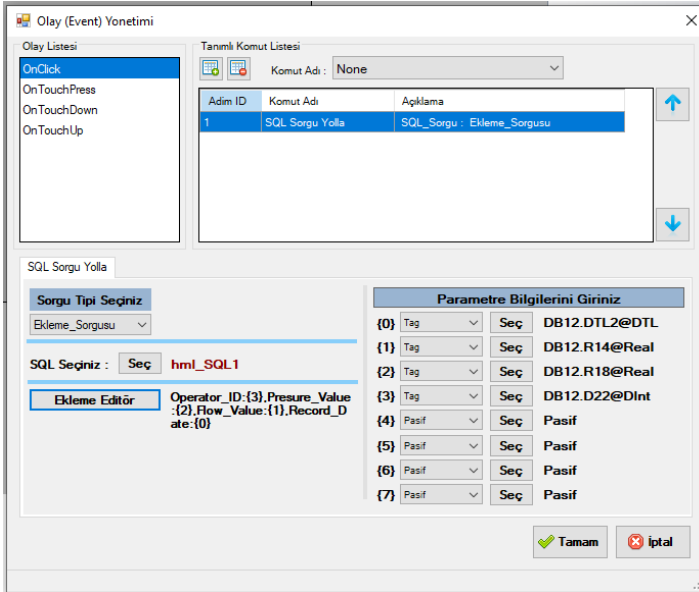
Operator_ID değerini {3} nolu parametreden alır.



Operator_ID değerini {3} nolu parametreden alır.



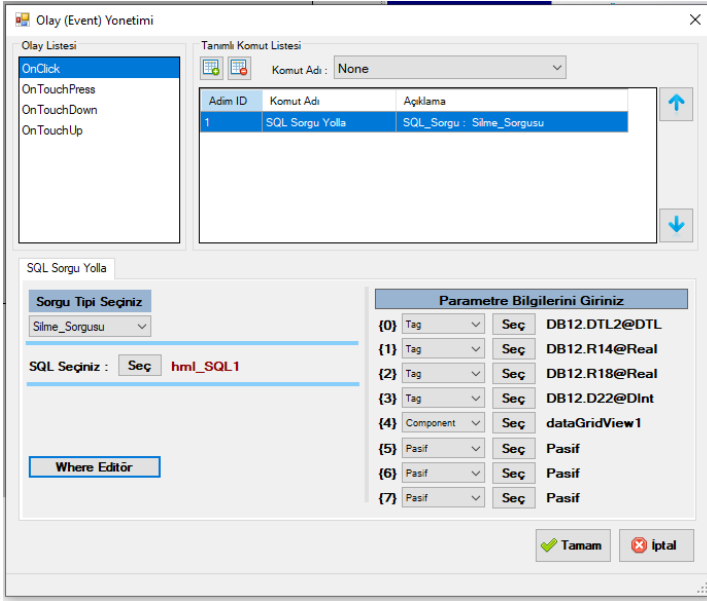
Diğer sütunlar içinde , yandaki gibi parametre seçimi yapılmıştır.



Ekleme Sorgumuzun son hali yandaki gibidir.

Projemizi çalıştırıp HMI_Button a tıkladığımızda; Sorgumuz çalışır , Parametrede belirlediğimiz tag adreslerinin değerlerini alır ve ekleme işlemini yapar.

Silme Sorgusu :



Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

SQL_Sorgu_Yola Komutunu ekleyiniz

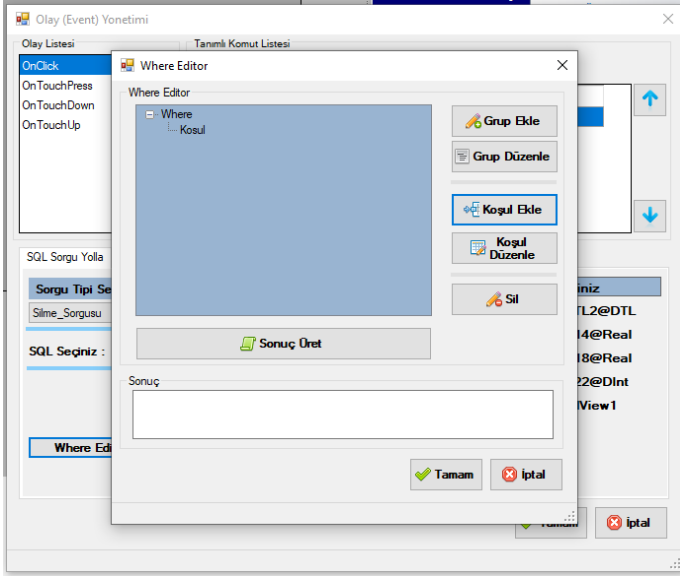
Sorgu Tipi : Silme_Sorgusu yapınız.

{4} nolu parametreye dataGridView1 kontrolü eklenmiştir. dataGridView1 kontrolünün parametre için alınan hücresi Cell[0] dir , bu hücrenin visiblesi false de olabilir , yani visible false olsa bile Cell[0] parametre değeri olarak alınır.

Önemli Uyarı : HMI SQL sisteminde otomatik tanımlanan vede visible false olan ID sütunu vardır. Bu sütun özel tanımlı bir sütundur ve her satır için değeri farklıdır. HMI SQL kontrolünde , otomatik olarak tanımlanır. HMI SQL Select sorgularında bu sütun alınır DataGridView kontrolüne eklenir ama gösterilmez. DataGridView in Cell[0] ı HMI SQL Select sorgusunda işte bu ID sütunudur

Silme Sorgusunda Editör olarak sadece Where editörü kullanılır.

HMI_SQL de tanımlı tabloya Where komutu ile sorgu yollanır. Eğer Where komutunu kullanmazsanız , HMI_Button nesnesine tıkladığınızda , tüm satırlar silinir. Where komutu ile ; silinecek satır için koşul oluşturuyoruz.

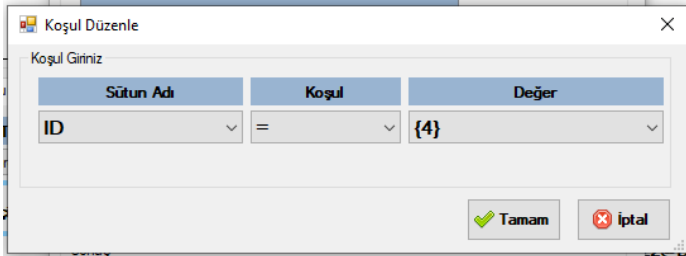


Where Editor Butonuna tıklayınız.

Yandaki gibi açılan ekranda sadece 1 adet koşul ekleyiniz.

Bu koşul içerisine ID sütunu eşit ise (otomatik tanımlanan) dataGridView1 in seçili satırının Cell[0] hücreesine koşulunu oluşturacağız.

Yani dataGridView1 de seçili satırı sileceğiz.

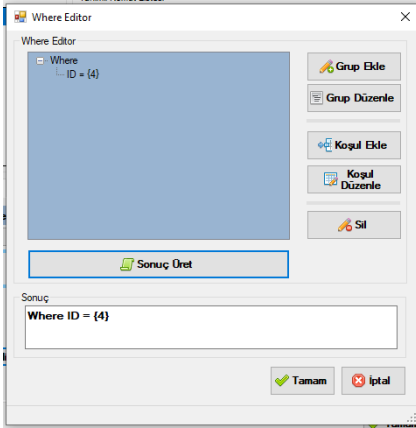


Özel tanımlı ID sütunu Koşul Düzenle ekranında mutlaka bulunur. Bu sütun HMI_SQL kontrolünde kullanıcıdan bağımsız olarak , otomatik tanımlanan sütundur ve değeri her satır için farklıdır.

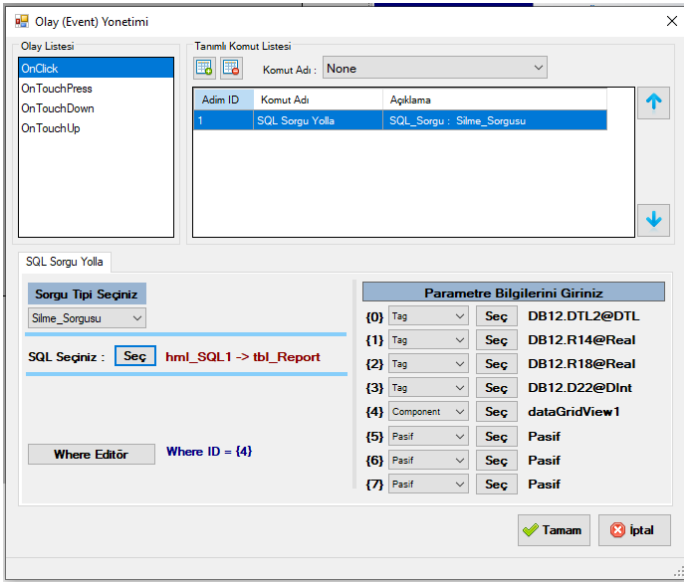
ID = {4}

ID sütunu eşit ise (otomatik tanımlanan) dataGridView1 in seçili satırının Cell[0] hücreesine

{4} : dataGrigView1 SelectedCell[0]



Where Sorgumuzun son hali yandaki gibidir.



Silme_Sorgu muzun son hali yandaki gibidir.

HMI_Button a tıklandığında silme sorgusu çalışır.

Yani dataGridView1 de seçili satırı silinir.

Genel Sorgu : Bu sorgu tipi ile kendi kompleks sorgularınızı rahatlıkla Mssql e yollayabilirsiniz.

Genel_Sorgu HMI_SQL Kontrolünün bağlantı parametrelerini kullanır ama HMI_SQL in tablosuna erişmek zorunda değildir. İçerisinde Kompleks Join işlemleri gibi sorgularıda yapabilirsiniz.

Adım ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Genel_Sorgu

Index	Tag	Seçim	Değer
{0}	Tag	Seç	DB12.DTL2@DTL
{1}	Tag	Seç	DB12.R14@Real
{2}	Tag	Seç	DB12.R18@Real
{3}	Tag	Seç	DB12.D22@DInt
{4}	Component	Seç	dateTimePicker_Basla
{5}	Component	Seç	dateTimePicker_Bitis
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

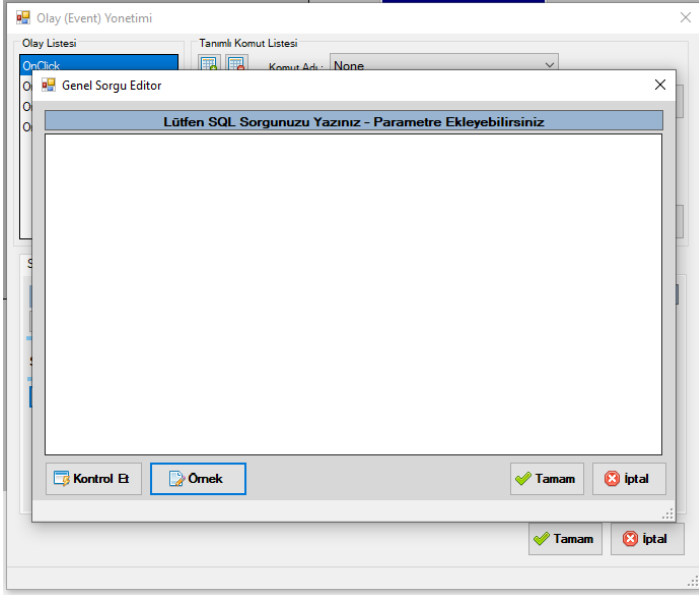
SQL_Sorgu_Yola Komutunu ekleyiniz

Sorgu Tipi : Genel_Sorgu yapınız.

Bu sorguda , Genel Sorguyu Select komutu ile kullanacağız.

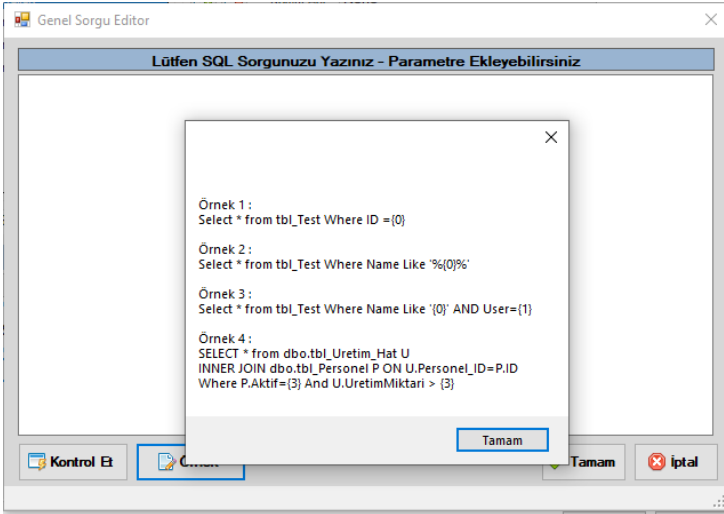
{4} : dateTimePicker_Baslangic
{5} : dateTimePicker_Bitis

Parametrelerini tanımladık ve bu kontrollerden tarih seçeceğiz , iki tarih aralığındaki satırları getireceğiz.

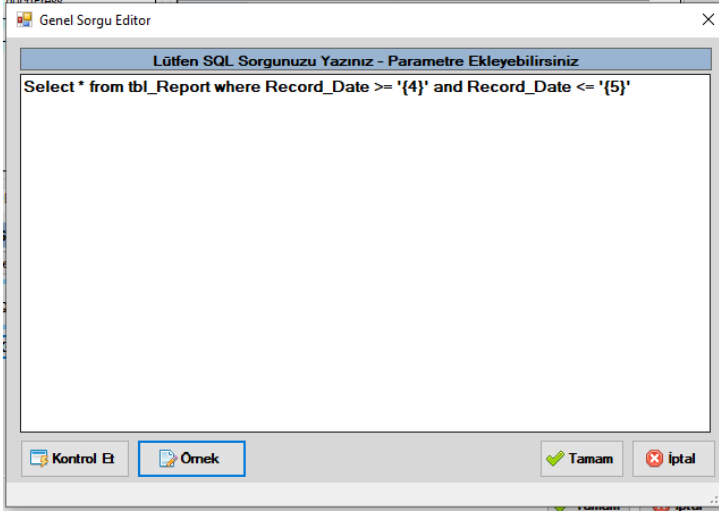


Genel Sorgu Editorüne tıkladığımızda yandaki sorgu ekranı çıkar.

Burada MSSQL de New Query de olduğu gibi direkt sorgunuzu yazabilirsiniz, parametre işaretini kullanarak {}, parametrelerinizi sorgu içerisine yerleştirebilirsiniz.



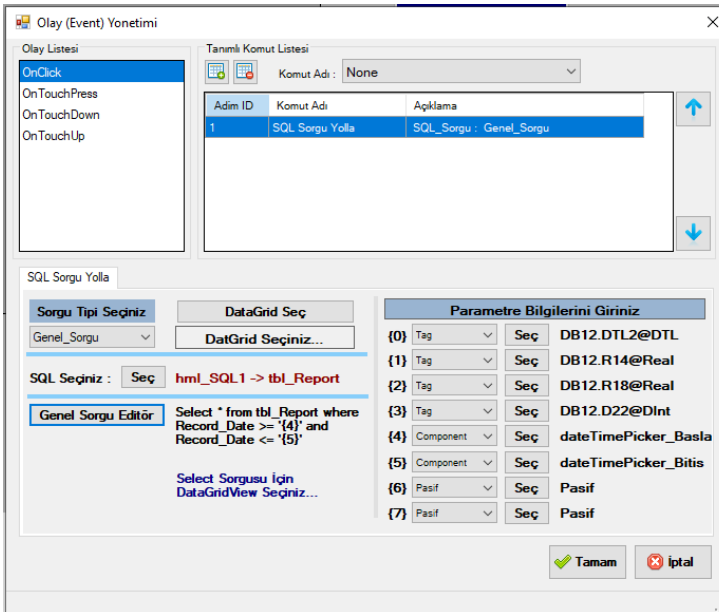
Örnek Buttonuna tıkladığınızda bazı sorgu örneklerini görebilirsiniz.



Yanda gördüğünüz gibi sorgunuzu rahatlıkla yazabilirsiniz.

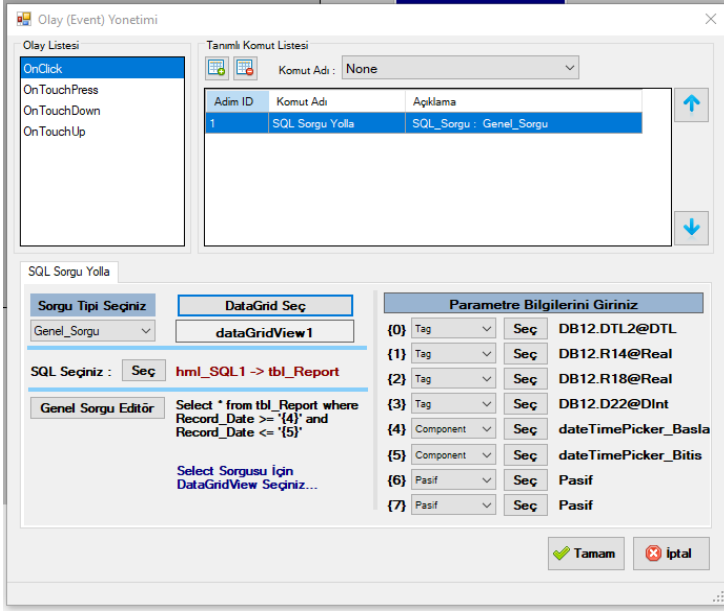
Genel_Sorgu da SQL Sorgu kuralları geçerlidir , bundan dolayı String , Char ve Date tipleri için tırnak(") kullanmayı unutmayınız.

Burada tbl_Report tablosunda sorgu işlemi yapılmıştır. **Genel_Sorgu için bu zorunlu değildir.** HMI_SQL kontrolünün kullandığı DataBase in diğer tabloları içinde sorgu yazabilirsiniz.



Sorgumuz bir Select sorgusu olduğu için , sonucun yazılacağı bir DataGridView e de ihtiyacımız var.

DataGrid Seç buttonuna tıklayıp , DataGridView Seçiniz.



Sorgumuzun son hali yandaki gibidir.

Genel_Sorgu ile

HMI_SQL kontrolünün kullandığı DataBase in tablolarında ,

Insert , Update , Delete , Select sorgularını parametre kullanarak rahatlıkla yazabilirsiniz.

Öncelikle Tarih Aralığı Seçiniz ve içerisine Genel_Sorgu ile Select Sorgusu yazılmış HMI_Button kontrolüne tıklayınız.

Başlangıç : 20.07.2025

Bitiş : 25.07.2025

Send Query

ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date
1	1	5	10	23.07.2025
2	1	6	15	22.07.2025

Where Editör Koşul İçerisinde {Dinamik} Kullanımı : Koşul değişkeninin daha esnek bir şekilde yazılmasını sağlar.

Örneğin ; Operatör adı ile ilgili bir sorgu yapacağımızı varsayalım. Br HMI_TextBox tanımlayalım , HMI_TextBox a girdiğimiz veriyi Operatör Adlarında aratalım. Bunu yaparkende **içerisinde geçen** sorgusunu yapalım. Yani HMI_TextBox a girdiğimiz metni , Operatörün adının içinde arasın , Operatörün Adı Murat ise girdiğimiz metin “mu” ise eşleşme sağlansın (Murat in içerisinde “mu” geçtiği için) . Bunu yapmak için MSSQL deki % operatörünü kullanacağız.

Aşağıdaki SQL Sorguları bu işlemi yapar.

```
Select * from tbl_Report Where Operator_Name Like '%mu%'
```

yada

```
Select Record_Date, Flow_Value, Pressure_Value, Operator_ID, Operator_Name  
from tbl_Report Where Operator_Name Like '%mu%'
```

İşte biz bu sorguyu sağlayacak formu tasarlayıp , dinamik parametreyi kulanarak , parametremizi '{2}' şeklinde yazacağız , {2} verisini HMI_TextBox dan alacak.

Başlayalım :

DB12(Write_To_SQL)			
	Name	Data type	Offset
1	Static		
2	Write	Bool	0.0
3	Written	Bool	0.1
4	Record_Date	DTL	2.0
5	Flow_Value	Real	14.0
6	Pressure_Value	Real	18.0
7	Operator_ID	DInt	22.0
8	Operator_Name	String[50]	26.0

Öncelikle DB12(Write_To_SQL) DataBlock una Operator_Name isminde String[50] tipinde bir değişken tanımlayalım.

Tag ID	Açıklama	Tag Adresi
88	Write	DB12.DBX0.0@Bool
89	Written	DB12.DBX0.1@Bool
90	Record_Date	DB12.DTL2@DTL
91	Flow_Value	DB12.R14@Real
92	Pressure_Value	DB12.R18@Real
93	Operator_ID	DB12.D22@DInt
94	Operator_Name	DB12.S26.50@String

Cross Reference

Form Adı	Nesne Adı	Tag ID
----------	-----------	--------

MasterPage üzerindeki HMI_Connect kontrolüne bu tagı ekleyelim.

Yanda DB12(Write_To_SQL) DataBlock una Operator_Name isminde String[50] tipinde bir değişkeni yazdık.

Bunu ister manuel ekleyebilirsiniz , isterseniz kopyalayıp yapıştırabilirsiniz.

Kopya-Yapıştır işlemi için HMI_Connect kontrolünün kullanımına bakabilirsiniz.

Daha sonra ise mutlaka , MasterPage de tanımladığımız **HMI_SQL** nesnesine bu yeni sütunu ekliyoruz. Aşağıda Operator_Name nin eklenmiş hali gösterilmiştir.

HMI_SQL Tasks

Tag List Özellikleri

YAZ
Seçili Tag : DB12.DBX0.0@Bool
SQL Yaz İçin Tag Seç

YAZILDI
Seçili Tag : DB12.DBX0.1@Bool
SQL Yazıldı İçin Tag Seç

TABLO
Tablo Adı : tbi_Report
VT Tablo Yönetimi
SQL Connection String
Connection String : Server=localhost;Database=
Alternate String : Server=localhost;Database=
Connect
MasterPage : hmi_Connect_Genel
Connect Seçimi Yap
Dock in Parent Container

Tablo Yönetimi

Tablo Tanımlama

tbi_Report

Connection String : Server=localhost;Database=db_HMI_SQL;Integrated Security=True;
Alternate String : Server=localhost;Database=db_HMI_SQL;Integrated Security=True;

Tag Seç : DB12.S26.50@String

ID	Sütun Adı	Açıklama	Sütun Tag Adresi
5	Operator_Name	Operator Name	DB12.S26.50@String -> Operator_Name
4	Record_Date	Record Date	DB12.DTL2@DTL -> Record_Date
3	Flow_Value	Flow Value	DB12.R14@Real -> Flow_Value
2	Pressure_Value	Pressure Value	DB12.R18@Real -> Pressure_Value
1	Operator_ID	Operator ID	DB12.D22@DInt -> Operator_ID

Test

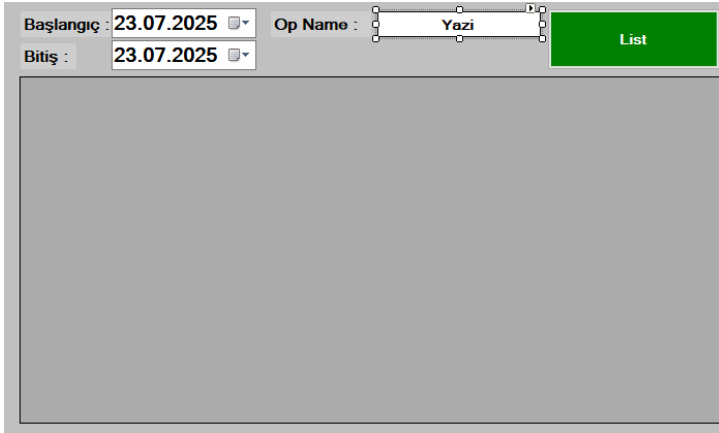
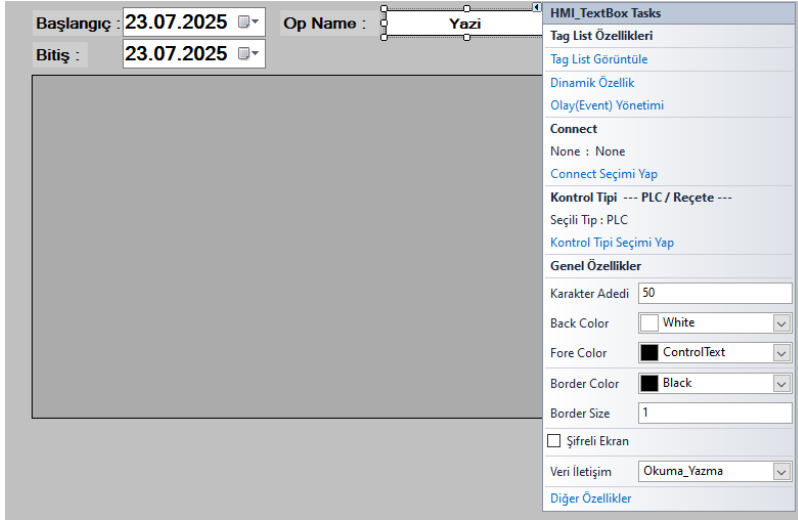
Tamam İptal

Sistem Çalışmaya Hazır

Artık yeni formumuzu tasarlayabiliriz. Aşağıdaki gibi OP Name için bir HMI_TextBox kontrolünü formumuza ekleyelim.

HMI_TextBox a Connect eklemesi yapmayınız. Eğer Connect ataması yapmazsanız , HMI_TextBox ve NumericBox u gene kullanabilirsiniz. Dokunmatik klavye ile veri girişi yapabilir ve ayrıca kod tarafında kontrolün Text özelliği ile veriyi alabilirsiniz.

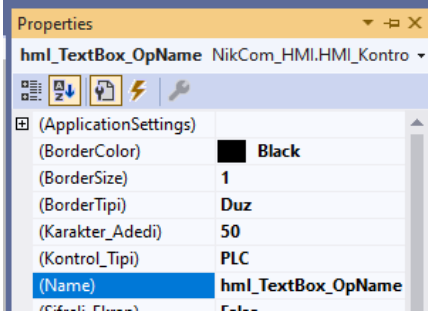
Yani HMI_TextBox ve HMI_NumericBox a Connect ataması yapmazsanız , size dokunmatik ekran klavyesini kullandırtan bir kontrol gibi davranır.



Yandaki ekranda Operator_Name için HMI_TextBox kontrolü eklenmiştir.

Bu kontrolün **Name** özelliği Properties ekranından

hmi_TextBox_OpName yapılmıştır.

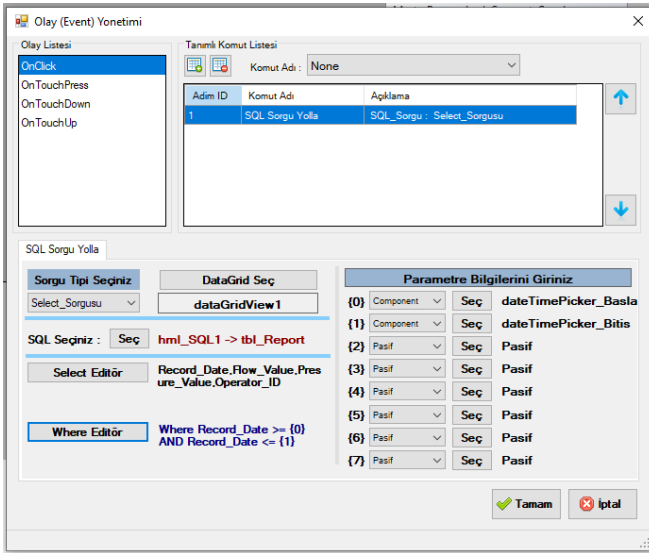


Yanda kontrolün Properties ekranı gösterilmiştir.

Properties ekranına Visual studio da

ile ulaşabilirsiniz.

Kontrolü tıkladığınızda , kontrolün özellikler Properties ekranında gösterilir.



Yanda daha önceden parametrelerini girdiğimiz "List" HMI_Button kontrolünün Event Olay Yönetim Özelliği görüntülenmiştir.

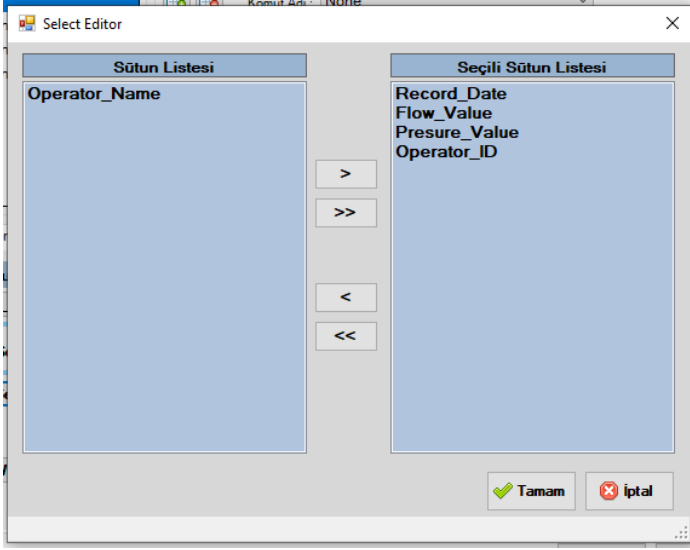
Bu Özelliğe

Select Editor den Operator_Name sütununu Sorgumuza ekleyeceğiz

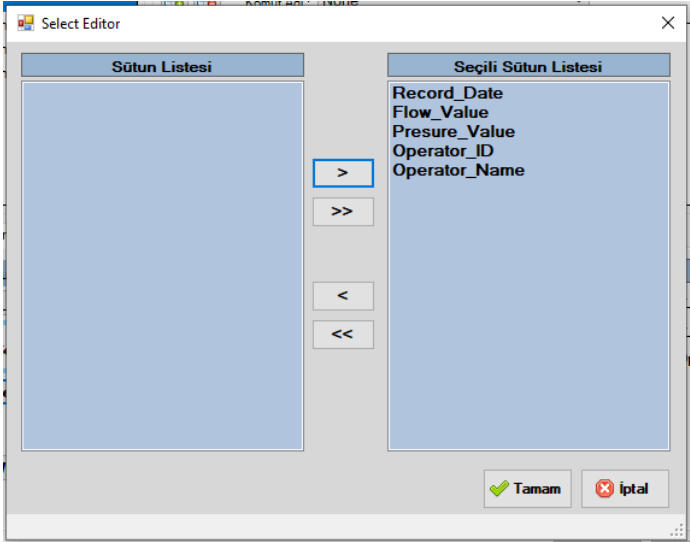
{2}: hml_TextBox_OpName

Parametresini ekleyeceğiz.

Where Editör ile de yeni koşul tanımlayıp , '%{2}%' için Dinamik değer girişi yapacağız.



Öncelikle Select Editör den Operator_Name Sütununu seçelim ve Sağa taşıyalım



Operator_Name Sütununu Sağa taşıdık.

Parametre Bilgilerini Giriniz		
{0}	Component	Seç dateTimePicker_Basla
{1}	Component	Seç dateTimePicker_Bitis
{2}	Pasif	Seç Pasif
{3}	Pasif	Seç Pasif
{4}	Pasif	Seç Pasif
{5}	Pasif	Seç Pasif
{6}	Pasif	Seç Pasif
{7}	Pasif	Seç Pasif

HMI_TextBox u Parametre bölümünde {2} nolu parametre olarak ekliyoruz.

Parametre Bilgilerini Giriniz		
{0}	Component	Seç dateTimePicker_Basla
{1}	Component	Seç dateTimePicker_Bitis
{2}	Component	Seç Component Giriniz
{3}	Pasif	Seç Pasif
{4}	Pasif	Seç Pasif
{5}	Pasif	Seç Pasif
{6}	Pasif	Seç Pasif
{7}	Pasif	Seç Pasif

{2} için Component i seçip , Seç buttonuna tıklayınız.

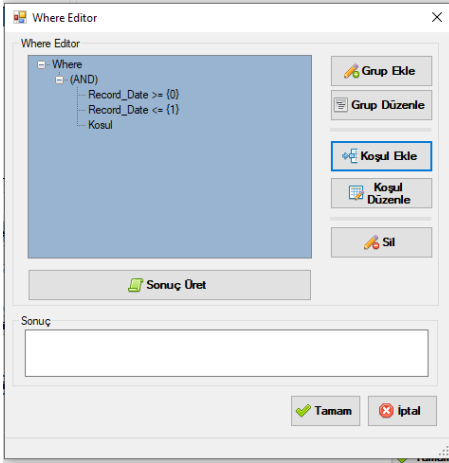
hmi_TextBox_OpName.Text
dataGridView1.Cell[0]
dateTimePicker_Bitis.Value
dateTimePicker_Baslangic.Value

Seç İptal Boş

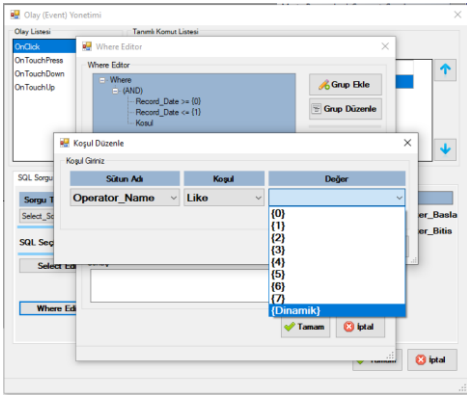
Gelen ekrandan , yeni eklediğimiz HMI_TextBox u seçip parametre olarak yüklüyoruz.

Parametre Bilgilerini Giriniz		
{0}	Component	Seç dateTimePicker_Basla
{1}	Component	Seç dateTimePicker_Bitis
{2}	Component	Seç hmi_TextBox_OpName
{3}	Pasif	Seç Pasif
{4}	Pasif	Seç Pasif
{5}	Pasif	Seç Pasif
{6}	Pasif	Seç Pasif
{7}	Pasif	Seç Pasif

{2} parametresine hmi_TextBox_OpName i yükledik.



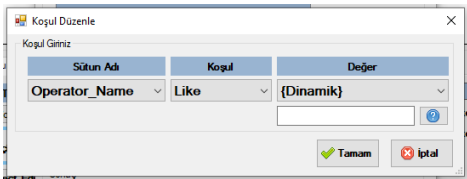
Artık Where editör e Koşul eklemesi yapabiliriz. AND altına yeni koşulumuzu ekleyelim.



Yeni eklediğimiz koşula çift tıklayalım ve açılan Koşul Düzenle ekranından

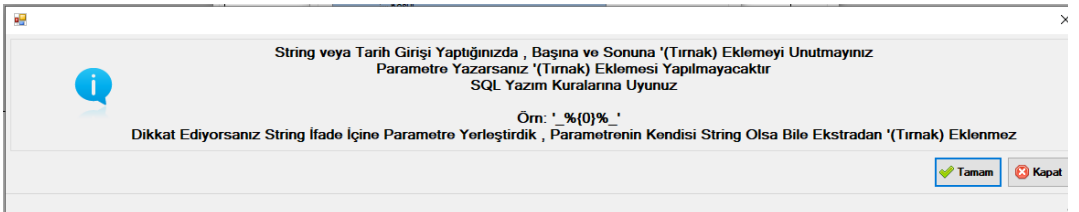
Operator_Name Like {Dinamik}

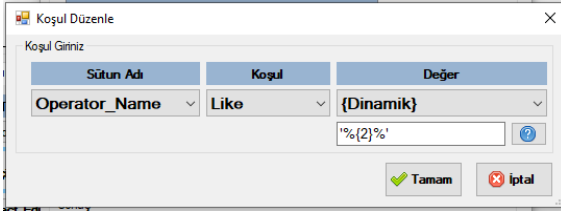
Seçimi Yapalım



Dinamik seçimi yaptığımızda aşağıda parametre girişi için bir TextBox görüntülenir.

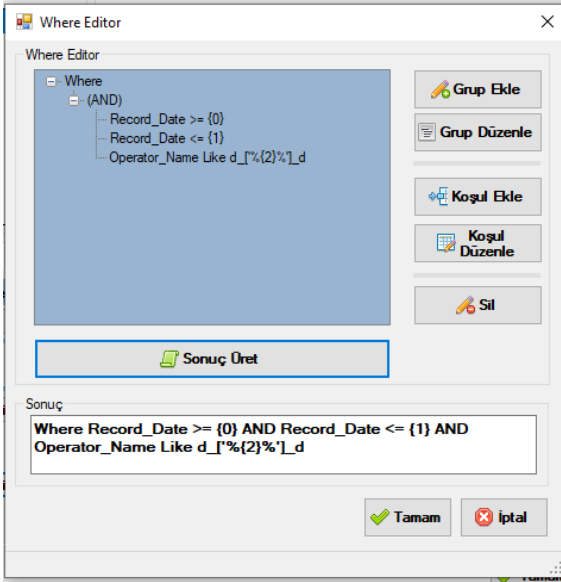
Yanındaki (?) Buttonuna mutlaka tıklayınız. Butona tıkladığınızda çıkan uyarı mesajını dikkatlice okuyunuz.





Parametre girişini yandaki gibi yapalım , burada " (Tırnak) işaretlerine dikkat ediniz.

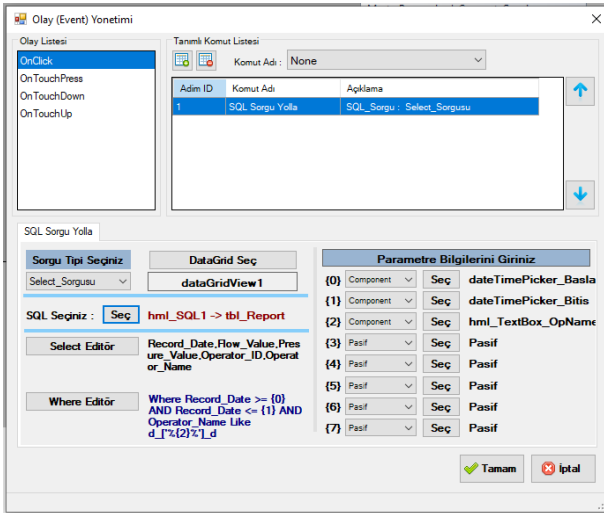
SQL Komutlarında String ve Date tipleri için tırnak kullanılmalıdır.



Dinamik parametre eklendiğinde sorgu

d_[']_d içerisine alınır.

Ekrandaki yazıma dikkat ediniz , bu Where Editörüne özgü bir yazım şeklidir.



Sorgumuzun son hali yandaki gibidir.

Column Name	Data Type	Allow Nulls
ID	bigint	<input type="checkbox"/>
Operator_ID	bigint	<input type="checkbox"/>
Pressure_Value	float	<input type="checkbox"/>
Flow_Value	float	<input type="checkbox"/>
Record_Date	datetime	<input type="checkbox"/>
Operator_Name	nvarchar(50)	<input type="checkbox"/>

ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date	Operator_Name
1	1	5	10	2025-07-23 00:0...	Murat
2	1	6	15	2025-07-22 00:0...	Murat
3	2	8	20	2025-07-22 00:0...	Remzi

Başlangıç : 21.07.2025 Op Name :

Bitiş : 25.07.2025

Record Date	Flow Value	Pressure Value	Operator ID	Operator Name
23.07.2025	10	5	1	Murat
22.07.2025	15	6	1	Murat
22.07.2025	20	8	2	Remzi

Başlangıç : 21.07.2025 Op Name :

Bitiş : 25.07.2025

Record Date	Flow Value	Pressure Value	Operator ID	Operator Name
23.07.2025	10	5	1	Murat
22.07.2025	15	6	1	Murat

Projemizi çalıştırdığımızda , MSSQL tbl_Report tablosuna yandaki gibi Operator_Name sütununun eklendiğini görebilirsiniz.

Managment Studio ile tabloya yeni bir satır ekleyiniz ve Operator_Name sütununun satırlarında ekrandaki gibi doldurunuz.

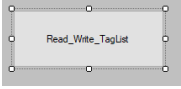
Projemizi çalıştırınız. Tarih aralığı seçip List HMI_Buttonuna tıkladığımızda satırların geldiğini görürsünüz.

Burada hml_TextBox_OpName şimdilik boş

hml_TextBox_OpName içerisine "mu" yazınız ve List buttonuna tıklayınız.

Belirtilen tarih aralığındaki , içerisinde "mu" geçen Operatörler listelenecektir.

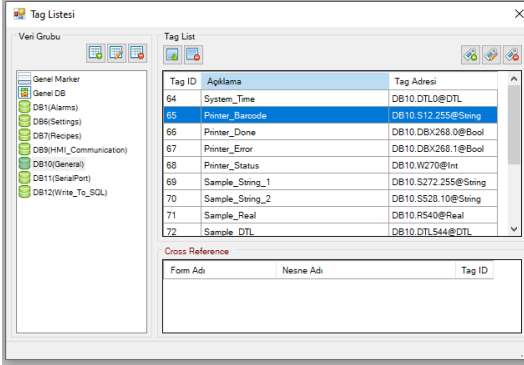
Taglara Kod İle Erişim : İsterseniz C# Kod ekranını açar ve HMI_Connect TagList de tanımladığınız taglara kod ekranından rahatlıkla erişebilirsiniz. Bu sayede C# .NET Kütüphanesini projenizde rahatlıkla kullanabili ve NikCom_HMI ı C# kodları ile birlikte rahatlıkla kullanabilirsiniz.



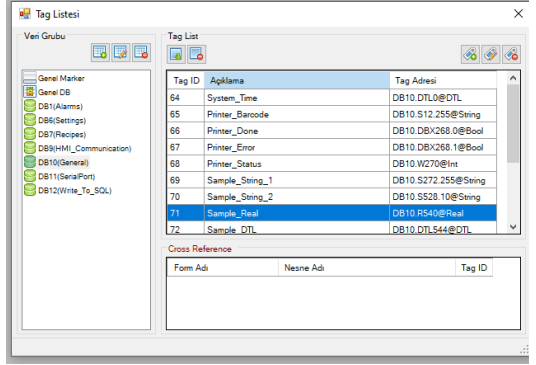
Standart bir Button kontrolü tanımlayınız Properties ekranında Text özelliğini bulunuz ve Read_Write_TagList yazınız.
Daha sonra çift tıklayıp kod satırında Click olayına ulaşınız.

<pre>1 reference private void button_Read_Write_TagList_Click(object sender, EventArgs e) { } }</pre>	<p>Kod satırındaki Click olayı yandaki gibidir.</p>
---	---

Aşağıda seçili tag adresi; 65 ID li tag adresi ile işlem yapacağız. Bu tagın değerini kod satırından hem okuyup hemde kod satırından bu adrese değer göndereceğiz.



Aşağıda seçili tag adresi; 71 ID li tag adresi ile işlem yapacağız. Bu tagın değerini kod satırından hem okuyup hemde kod satırından bu adrese değer göndereceğiz.



Tag Adresinden Değer alma ve Değer yazma kodları aşağıdaki gibidir. Lütfen Dikkatlice inceleyiniz.

```
1 reference
private void button_Read_Write_TagList_Click(object sender, EventArgs e)
{
    //-----Write-----
    //Write To TagList
    //Aşağıdaki Gibi Tag ID ile ulaşabilirsiniz
    hmi_Connect_General.TagList_Deger_Yaz(65, "ABC12345");

    //veya Tag ID nin Tag Adresi ilede Erişilebilir.
    //65 nolu Tagın Adresi DB10.S12.255@String Adresidir , 10 nolu DataBlock , 12. byte dan başlar , 255 karakterlik string
    hmi_Connect_General.TagList_Deger_Yaz("DB10.S12.255@String", "ABC12345");

    //Aynı İşlemi Real Tipte Yapalım-----

    // Tag ID ile Erişim
    hmi_Connect_General.TagList_Deger_Yaz(71, 86.18);

    //Tag Adresi ile Erişim
    hmi_Connect_General.TagList_Deger_Yaz("DB10.R540@Real", 86.18);

    object value;
    //-----READ-----
    //Read to TagList
    //Aşağıdaki Gibi Tag ID ile ulaşabilirsiniz
    value = hmi_Connect_General.TagList_DegerAl(65);
    MessageBox.Show(value.ToString());

    //veya Tag ID nin Tag Adresi ilede Erişilebilir.
    //65 nolu Tagın Adresi DB10.S12.255@String Adresidir , 10 nolu DataBlock , 12. byte dan başlar , 255 karakterlik string
    value = hmi_Connect_General.TagList_DegerAl("DB10.S12.255@String");
    MessageBox.Show(value.ToString());

    //Aynı İşlemi Real Tipte Yapalım-----

    // Tag ID ile Erişim
    value = hmi_Connect_General.TagList_DegerAl(71);
    MessageBox.Show(value.ToString());

    //Tag Adresi ile Erişim
    value = hmi_Connect_General.TagList_DegerAl("DB10.R540@Real");
    MessageBox.Show(value.ToString());
}
```

İçindekiler

NIKCOM HMI KULLANIM KLAVUZU.....	1
Giriş ve Amaç :.....	2
Kurulum.....	3
Nikom_HMI .dll Projenize Nasıl Yüklenir.....	3
Component Listesi ve Kısa Açıklamalar :.....	7
PLC ve DataBlock Ayarları.....	8
COMPONENTLER.....	9
HMI_Connect :.....	10
Tag Tanımlama :.....	11
Reçete Tanımlama :.....	14
Dil Listesi Tanımlama :.....	15
Security :.....	16
HMI_Label :.....	17
Text Düzenle :.....	18
Dinamik Özellik :.....	18
Diğer Özellikler :.....	20
HMI_Button :.....	21
Dinamik Özellik :.....	22
Olay (Event) Yönetimi :.....	22
Security :.....	25
HMI_Triger :.....	25
HMI_NumericBox :.....	29
Diğer Özellikler:.....	31
HMI_TextBox :.....	32
HMI_ComboBox.....	33
HMI_TextList_SelectBox :.....	35

HMI_DateTime :	37
HMI_MessagePanel :	39
HMI_ImageList :	41
HMI_Connect_Status	45
HMI_PanelForm	46
HMI_Security.....	49
HMI_Serial.....	50
HMI_AlarmList.....	52
HMI_IOList.....	58
HMI_RecipeList.....	61
HMI_SQL.....	67
Connection String.....	71
Event Olay Yönetimi İle SQL İşlemleri :	74
Select Sorgusu	76
Ekleme Sorgusu	85
Silme Sorgusu	88
Genel Sorgu	91
Where Editör Koşul İçerisinde {Dinamik} Kullanımı	95