

KARABÜK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



MTM406 ENDÜSTRİYEL İLETİŞİM SİSTEMLERİ

DERS NOTLARI

No :
Ad Soyad :
Öğretim :

Doç. Dr. Raif BAYIR
Arş. Gör. Emel SOYLU

2014

İçindekiler

VERİ İLETİŞİMİ	2
OSI REFERANS MODELİ	2
ENDÜSTRİYEL İLETİŞİM PROTOKOLLERİNE GİRİŞ	8
ENDÜSTRİYEL AĞLAR.....	12
RS-232	12
RS-485	14
MODBUS	15
DEVICENET	46
PROFIBUS	76
CAN-BUS	91
MPI HABERLEŞME SİSTEMİ.....	111
USB	118

VERİ İLETİŞİMİ

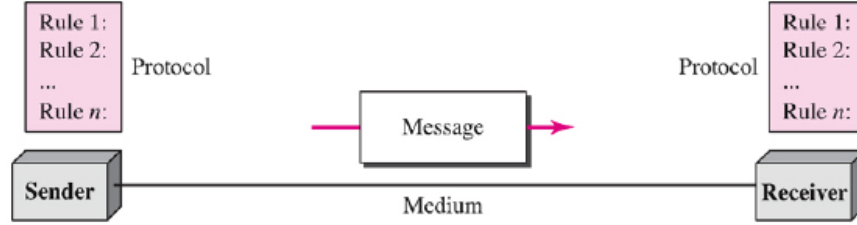
Veri iletişimi (data communications) bilginin herhangi bir ortam (hava, kablo vb.) kullanılarak iki cihaz arasında transfer edilmesidir.

Veri iletişiminin etkinliği aşağıdaki 4 temel parametreye dayanır.

- Doğru hedef (Delivery): Verinin sadece doğru hedefe ulaşmasıdır.
- Doğruluk (Accuracy): Verinin kaynağından çıktığı şekliyle iletilmesidir.
- Zaman (Timeliness): Verinin zamanında hedefe ulaşmasıdır. Gerçek zamanlı iletişimde (audio, video) çok önemlidir.
- Gecikme değişimi (Jitter): Paketlerin hedefe ulaşma süresindeki değişimdir.

Veri iletişim sistemi 5 elemandan oluşur:

- Message (mesaj): iletilen bilgidir (ses, görüntü, metin, sayı, resim)
- Sender (gönderici): veriyi ileten cihazdır (pc, g), workstation, video camera)
- Receiver (alıcı): veriyi alan cihazdır (pc, workstation, televizyon)
- Transmission medium (iletim ortamı): verinin gönderen ve alan cihaz arasında iletilmesini sağlayan fiziksel yoldur (twisted pair wire, coaxial cable, fiber optic cable, radio waves)
- Protocol (protokol): veri iletişimini başlatır, yönetir, sonlandırır.



Protokol, lisanları ayrı olan iki olgu arasında, belirli bir düzeyde, iletişimi sağlamak amacıyla gerekli kuralları ortaya koyan bir çevirmendir. Protokol, iletişimi sağlamak amacıyla ortaya konan kurallara uyan iki veya daha fazla orandaki Çevirmeni kullanan tarafları kapsar. Bu Protokol genellikle farklı dil kullanan iki veya daha fazla sistemin ortak bir iş yapmalarını gerektiren düzenler arasında yapılır. Kısaca bu tarifleri yaptıktan sonra Açık Protokol'ün ne olduğunun tanımına geçebiliriz.

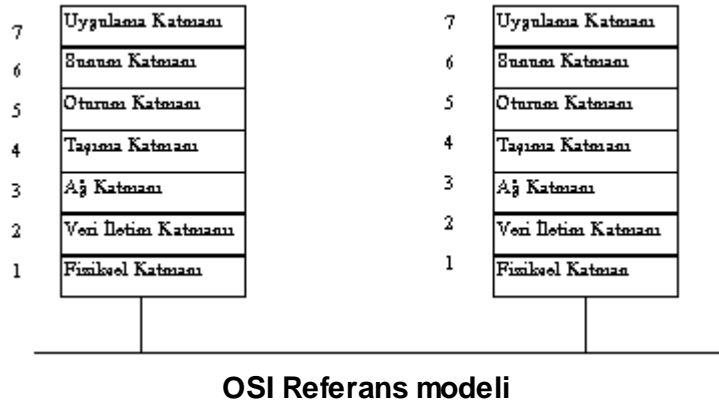
Elektronik haberleşmede kullanılan protokoller o kadar çoktur ki, ayrı dili taşıyan iki sistem arasında iletişim kurmak zorunluluğu olduğunda var olan güçlükler kendini bariz bir şekilde hemen ortaya koyar. Hele teknik bilgilerin alış-verişinde bu daha da vehim bir duruma girer.

- Bilginin iki nokta arasında iletilmesi için analog veya sayısal sinyale çevrilmesi gerekir.
- Sayısal-sayısal çevirmede sayısal veri sayısal sinyale dönüştürülür.
- Analog-sayısal çevirmede analog veri sayısal sinyale dönüştürülür.
- Çevirme işleminden elde edilen sinyal paralel veya seri olarak iki nokta arasında iletilir.

OSI REFERANS MODELİ

Bilgisayarlar arası iletişimin başladığı günden itibaren farklı bilgisayar sistemlerinin birbirleri arasındaki iletişim daima en büyük problemlerden birisi olmuş ve bu sorunun üstesinden gelebilmek için uzun yıllar boyunca çeşitli çalışmalar yapılmıştır. 1980'li yılların başında Uluslararası Standartlar Organizasyonu (International Standards Organization-ISO) bilgisayar sistemlerinin birbirleri ile olan iletişimde ortak bir yapıya ulaşmak yönünde çabaları sonuca bağlamak için bir çalışma başlatmıştır. Bu çalışmalar sonucunda 1984 yılında Açık Sistem Bağlantıları (Open Systems Interconnection-OSI) referans modeli ortaya çıkarılmıştır. Bu model sayesinde değişik bilgisayar firmalarının ürettikleri bilgisayarlar arasındaki iletişimi bir standarda oturtmak ve farklı standartlar arası uyumsuzluk sebebi ile ortaya çıkan iletişim sorununu ortadan kaldırmak hedeflenmiştir. OSI referans modelinde, iki bilgisayar sistemi arasında yapılacak olan iletişim problemini çözmek için 7 katmanlı bir ağ sistemi önerilmiştir. Bir baksa deyişle bu temel problem 7 adet küçük probleme parçalanmış ve her bir problem için ayrı ayrı bir çözüm yaratılmaya çalışılmıştır. Bu 7 katmanın en altında yer alan iki katman yazılım ve donanım, üstteki beş katman ise genelde yazılım yolu ile çözülmüştür. OSI modeli, bir bilgisayarda çalışan uygulama

programının, iletişim ortamı üzerinden başka bir bilgisayarda çalışan diğer bir uygulama programı ile olan iletişiminin tüm adımlarını tanımlar. En üst katmanda görüntü ya da yazı şeklinde yola çıkan bilgi, alt katmanlara indikçe makine diline dönüşür ve sonuç olarak 1 ve 0 lardan ibaret elektrik sinyalleri halini alır. Aşağıdaki şekilde OSI referans modeli katmanları ve bir yerel ağ üzerindeki durumu gösterilmektedir:



OSI katmanlarının tanımlanan temel görevleri:

7-Uygulama

Kullanıcıya en yakın olan katmandır. Spreadsheet, kelime işlemci, banka terminali programları vs. bu katmanın parçalarıdır.

6-Sunum

Bu katmanda gelen paketler bilgi haline dönüştürülür. Bilginin karakter set cevrimi veya değiştirilmesi, şifreleme vs. görevlerini bu katman üstlenir.

5-Oturum

İki bilgisayar üzerindeki uygulamaların birbirini fark ettiği katmandır.

4-Taşıma

Bu katman gelen bilginin doğruluğunu kontrol eder. Bilginin taşınması esnasında oluşan hataları yakalar ve bunları düzeltmek için çalışır.

3-Ağ

Bağlantıyı sağlayan ve ulaşılmak istenen bilgisayara giden yolu bulan katmandır. Yönlendirme protokolleri bu katmanda çalışır.

2-Veri iletim

Bu katman fiziksel katmana ulaşım stratejisini belirler. Fiziksel adresleme, ağ topolojisi, akis kontrolü vs. bu katmanın görevlerindendir. Köprü cihazları bu katmanda çalışır.

1-Fiziksel

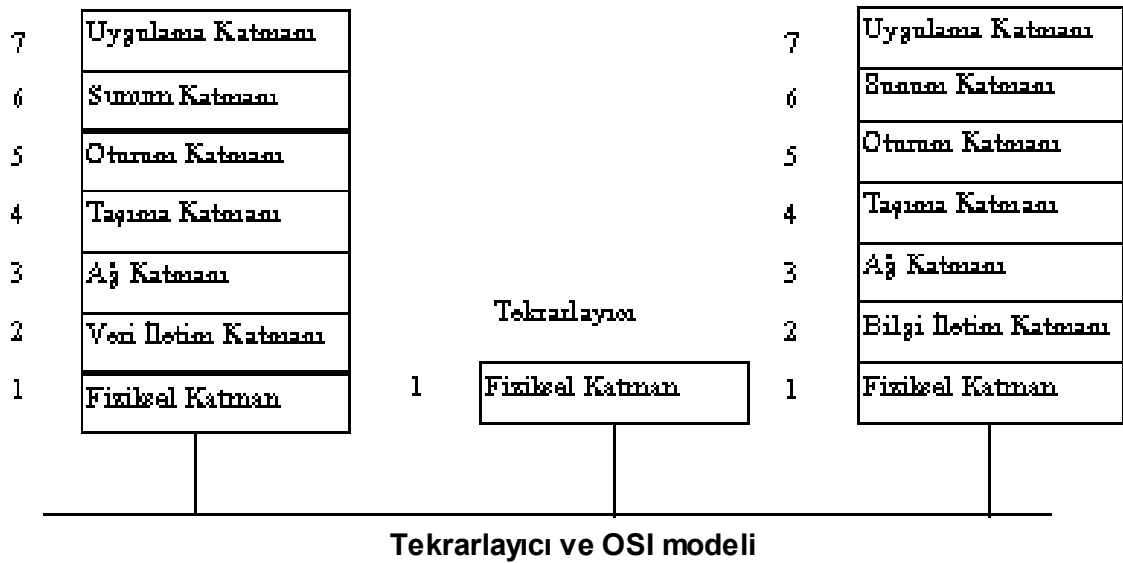
Bu katman ağın elektriksel ve mekanik karakteristiklerini belirler. Modülasyon teknikleri, çalışma voltajı, frekansı vs. bu katmanın temel özelliklerindendir. OSI referans modeli bir ağ uygulaması değildir. OSI sadece her katmanın görevini tüm detayları ile tanımlar. Bu modeli bir gemi ya da ev projesine benzetebiliriz. Nasıl aynı gemi planını alıp farklı firmalar gemi yapabilirse OSI modeli de böyledir. Nasıl aynı gemi planından iki farklı firma gemi ürettiğinde en azından kullanılan çiviler farklı yerlere çakılırsa, OSI modeli de gerçekleştiren firmadan firmaya farklılık gösterebilir.

Bağlantı Aygıtları

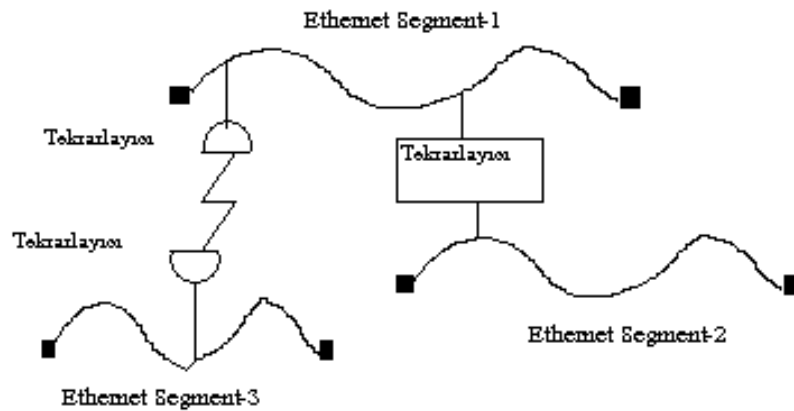
Bilgisayar ağı erişiminde genel olarak dört tip bağlantı aygıtı kullanılır: tekrarlayıcı (repeater), köprü (bridge), yönlendirici (router) ve geçityolu (gateway). Tekrarlayıcılar tamamen protokol bağımsız olarak fiziksel katmanda çalışır ve fiziksel genişleme amaçlı kullanılırlar. Geleneksel köprüler aynı protokolü kullanan Yerel Ağlar arasında temel veri düzeyinde bağlantı sağlar. Buna karşılık, geleneksel yönlendiriciler değişik tipteki ağ protokollerini idare edebilecek şekilde programlanabilirler ve böylelikle aynı geniş ağ alanı üzerinde farklı tipteki Yerel Ağları ve bilgisayar sistemlerini destekleyebilirler. Geçityolları daha karmaşık olup, işlem yoğunluklu protokol cevrimi yaparak uygulamalar arasında işletilebilirliği (interoperability) sağlarlar.

Tekrarlayıcı (Repeater)

Tekrarlayıcılar aşağıdaki şekilden de görüleceği gibi fiziksel katmanda çalışan cihazlardır.



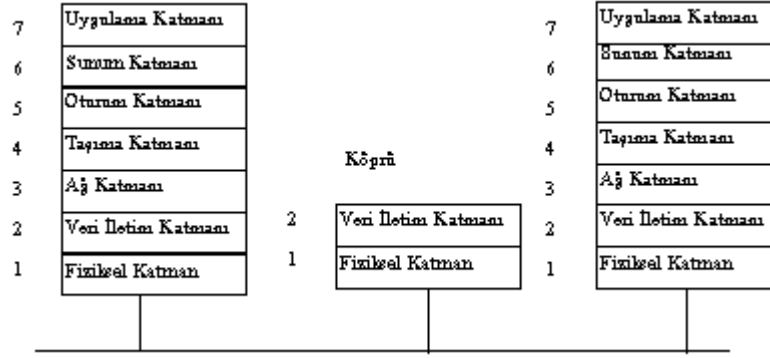
Tekrarlayıcının temel görevi bir fiziksel ortamdaki (kablo, fiber-optik, radyo dalgası vs.) sinyali alıp kuvvetlendirip bir diğer fiziksel ortama vermektir. Ağların fiziksel büyüklük sınırlarını daha da genişletmek amacı ile kullanılan bu cihazlar ile kuramsal olarak bir bilgisayar ağı sonsuza kadar genişletilebilir. Ancak çeşitli bilgisayar ağlarındaki tasarım sınırlamaları nedeni ile gerçekte bu genişleme belli sınırlar içinde kalmaktadır. Aşağıda tekrarlayıcıların bir ağ üzerinde nasıl kullanıldıklarını göstermektedir.



Temelde bir ağın genişletilmesi amacı ile kullanılan tekrarlayıcılar çok kolay kurulmaları, çok az bakım gerektirmeleri ve fiyatlarının ucuz olması sebepleri ile çok popüler cihazlardır.

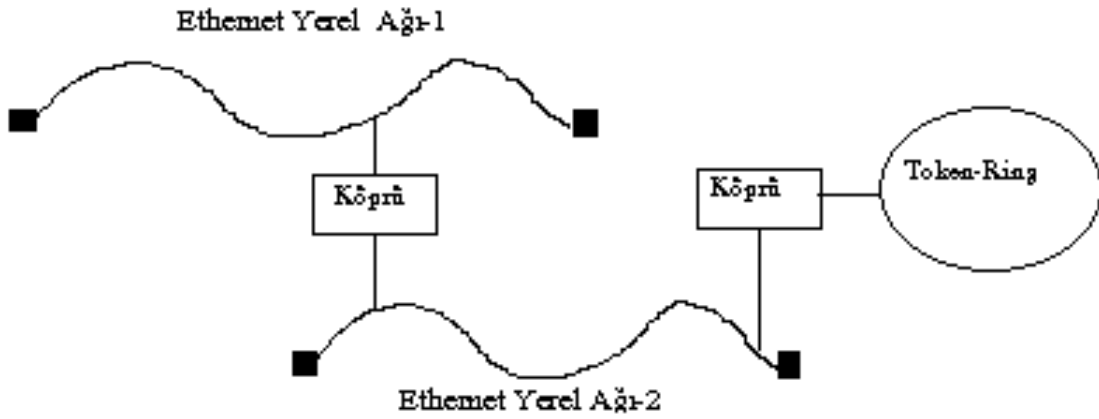
Köprü (Bridge)

Modern, protokol-şeffaf köprüler aşağıdaki şekilde görüldüğü gibi OSI referans modelinin veri iletim (data link) katmanında çalışırlar



Köprü ve OSI modeli

Köprü cihazları temelde bağımsız iki ağın (farklı ağ teknolojilerini kullanabilirler-Ethernet ve Token-Ring gibi) birbirine bağlantısı için kullanılırlar. Aşağıdaki şekilde iki Ethernet ve bir Token-Ring ağının birbirlerine köprüler vasıtası ile yapılan bağlantısı gösterilmektedir. Bir köprü bağladığı alt ağlar üstündeki tüm trafiği yürütür. Her paketi okur, paketin nereden geldiğini ve nereye gittiğini görmek için MAC (Media Access Control)-katman kaynağını ve yerleşim (destination) adresini inceler. Bu süzme yeteneği mesajları yayınlamak ya da yerel veri trafiğinin diğer ağ üzerine geçmesini engellemek için etkili bir yol sağlar. Bazı köprüler adres süzmenin ve protokol tipine bağlı süzgecin de ötesine gider.



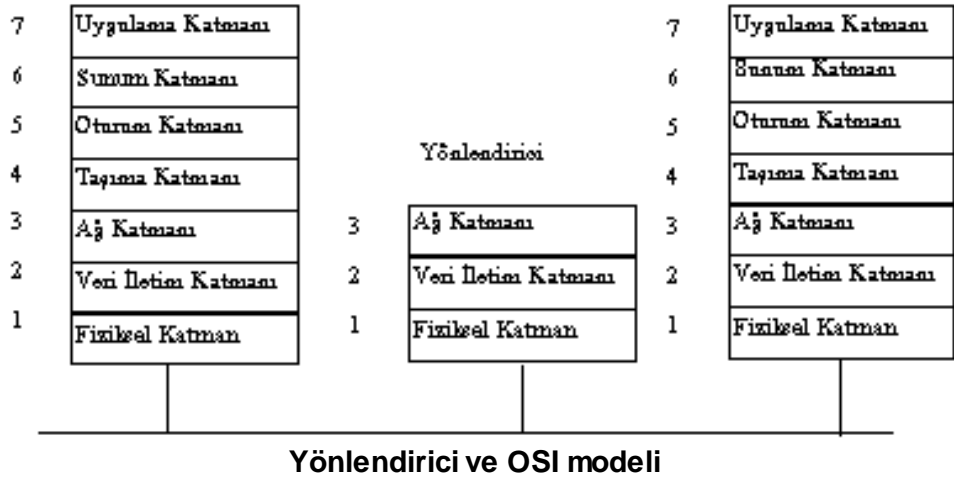
uygulaması

Bir Köprü

Bir köprü, DECnet, TCP/IP, XNS gibi farklı iletişim protokollerini kullanarak, protokol uyumluluğunu göz önüne almadan ağlar arasında fiziksel bağlantı sağlayabilse de, bu uygulamalar arasında işletilebilirliğini garanti etmemektedir. Bu, OSI referans modelinin yüksek katmanlarında işleyen ve farklı işletim ortamları arasında çevrim yapabilen standalone protokol çeviricilerini gerektirmektedir. Köprülü ağlar, protokol çevrimlerinin olmadığı, güvenlik gereksinimlerinin en az olduğu ve gereken tek şeyin basit yönlendirme olduğu durumlarda başarılıdır.

Yönlendirici (Router)

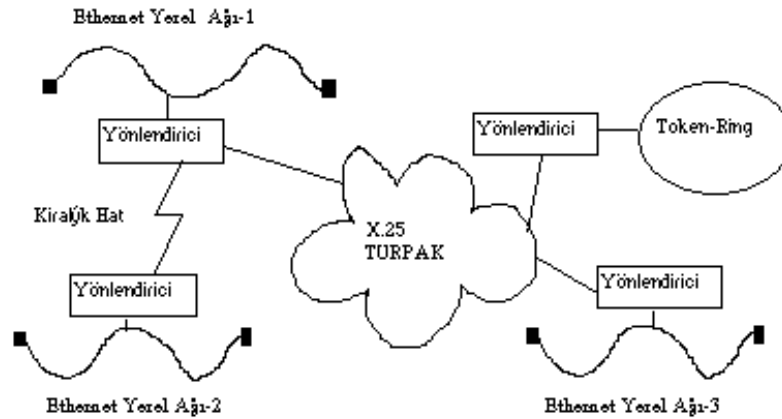
Yönlendiriciler aşağıdaki şekilde görüldüğü gibi OSI referans modelinin ağ (network) katmanında çalışırlar.



Bir köprü sadece paketlerin kaynağını ve gittiği yerin adresini kontrol ederken bir yönlendirici çok daha fazlasını yapar. Bir yönlendirici ağın tüm haritasını tutar ve paketin gittiği yere en iyi yolu belirleyebilmek için tüm yolların durumunu inceler.

Yönlendirici farklı fiziksel yapıda olan ve farklı protokolleri çalıştıran yerel ya da geniş alan ağlarının birbirleri ile olan bağlantısında başarı ile kullanılabilir.

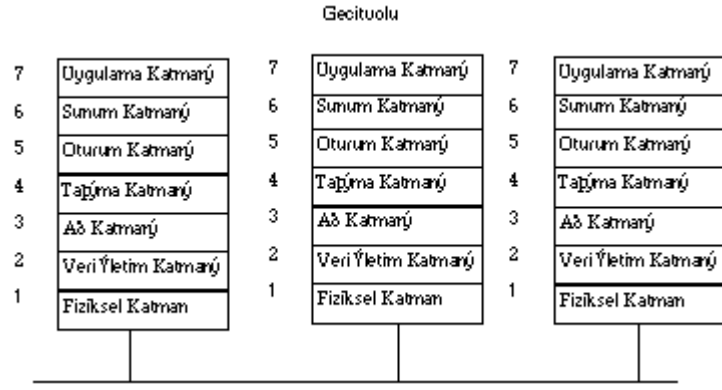
Bir yönlendirici, OSI referans modelinin ağ katmanında genel olarak tanımlanan protokollerle, yerel bölge ağlarını geniş bölge algarına bağlar. Bu özellikleri sayesinde örneğin yönlendirici TCP/IP kullanarak bir Ethernet ağının X.25 paket ağına bağlamasını sağlar. Eski yönlendiriciler protokol bağımlı olduklarından, kuruluşların ağ işletim ihtiyaçlarını karşılamak için birden fazla yönlendirici gerekebilir. Yeni yönlendiriciler ise, birden fazla ve değişik protokolü aynı anda idare edebilmektedirler.



Yönlendiriciler paketleri iki istasyon arasındaki en iyi yolu gösteren yönlendirme tablosuna göre ilerleterek ağ üzerindeki yolları en iyi şekilde kullanırlar. Yönlendiriciler kendi yönlendirme tablolarını oluşturduklarından, ağ trafiğindeki değişikliklere hemen ayak uydururlar ve böylelikle veri yükünü dengelerler. Aynı zamanda, yönlendiriciler ağdaki değişiklikleri tespit ederler ve aşırı yüklü ve işlemeyen bağlantıları önlerler.

Geçityollari (Gateway)

Geçityollari köprü ve yönlendiricilerin yeteneklerinin de ötesine geçerler. Aşağıdaki şekilden de görülebileceği gibi OSI referans modelinin üst katmanlarında işlerler.



Geçityolu ve OSI modeli

Geçityolları sadece farklı noktadaki ağları bağlamakla kalmaz aynı zamanda bir ağdan taşınan verinin diğer ağlarla uyumlu olmasını da garanti ederler. Bu bir serverda, minibilgisayarda ya da ana bilgisayarda bulunan protokol çevirim yazılımıyla yapılır. İnternet protokolleri farklı ağlar arasındaki veri iletimini, geçityollarıyla bağlı altağlardan oluşmuş otonom sistem (Autonomous System, AS) gruplarını birbirine bağlayarak yapar. Yani İnternet, her biri merkezi olarak yönetilen ağ ya da altağlar serisi olan AS serisinden oluşmaktadır. Her AS diğer AS'lere bağlantı sağlayan geçityolu sunar. Geçityolları tüm farklı ağları birlikte tutan bir yapıştırıcıdır. İnternet protokolleri altağların nasıl birbirine bağlı olduğunu ve bağlantı araçlarının nasıl çalıştığını tanımlar.

ENDÜSTRİYEL İLETİŞİM PROTOKOLLERİNE GİRİŞ

Otomasyon sisteminde sensör, aktüatör, valf, röle vb. elemanlar, PLC ve bilgisayarlarla haberleşmektedir. Böylece bilgi akışı sağlanarak veri toplama (data acquisition), veri işleme, veri analizi işlemleri gerçekleştirilmektedir

Veri iletişimi

Dijital Sinyal

Elektronikte tüm veri haberleşme standardı temel olarak “1” ve “0” rakamlarından oluşmaktadır. Bu sistem ile oluşturulabilen anlamlı en küçük veri yığını BIT (İkili Değer) olarak adlandırılır. BIT sistemi ile herhangi bir anahtarlama elemanının AÇIK (1) veya KAPALI (2) konumları tanımlanarak örneğin pnömatik bir valfi kumanda eden röle elemanının durumu belirlenebilir.

PLC tarafından kumanda edilen bir pnömatik valf grubunun açma/kapama bilgileri de “1” ve “0” rakamlarından oluşan bilgiler olarak transfer edilir. Bir valfin açılıp kapanması aynı zamanda o valfe gerekli enerjinin gönderilmesi veya voltajın kesilmesi anlamını da taşır. Eğer valf 24V DC voltaj ile çalışıyorsa “0” konumu “0 V” veya düşük (LOW) sinyal, “1” konumu “24 V” veya yüksek (HIGH) sinyal olarak değerlendirilir.

Tüm elektronik haberleşme sistemi elbette ki sadece açık/kapalı kombinasyonundan oluşmamaktadır. İçerik olarak sadece iki olasılıktan daha başka bilgilerin kullanıldığı durumlarda “0” ve “1” rakamlarının çeşitli kombinasyonlarından oluşan veri yığınları oluşturulur. 8 BIT ile oluşturulan BYTE ve iki BYTE ile oluşturulan WORD veri yığınları bunlara örnektir.

Sadece “1” ve “0” rakamlarından oluşan 8 parçalık veri yığını (8 BIT = 1 BYTE) 0-255 aralığında bir değer içerebilmektedir. Bu rakam, 8 parçalık veri paketi ile ifade edilebilecek bilginin alt ve üst sınırlarını ve verinin çözünürlüğünü belirler. Örneğin 0..6 bar aralığında basınç regülasyonu yapabilen elektro pnömatik oransal basınç kontrol valfimizin 8 BIT kontrol sistemi olduğunu kabul edelim. Regülatörümüzün 2.59 bar basınç üretebilmesi için; Pset değeri $255 \times 2,59 / 6,0 = 110$ olarak sisteme gönderilmelidir. Ancak öncelikle onluk düzendeki bu değer ikili düzene çevrilmeli ve 8 basamaklı bir sayı olan “0110 1110” rakamı olarak cihaza gönderilmelidir.

Analog Sinyal

Doğrudan elektriksel olmayan basınç, sıcaklık veya pozisyon gibi değişkenleri çeşitli algılama elemanları ile ölçerken elde edilen veriler genellikle akım veya voltaj cinsinden analog değerler olarak ölçülür. Analog veri, belirli bir değer aralığında süreklidir ve teorik olarak sonsuz sayıda değer alabilir. Dijital veri ise analog verinin aksine aynı ölçme aralığı içinde sadece belli değerleri alabilir ve kesintili ölçümler verir. Kullandığımız kontrol sisteminin ve ölçüm cihazının BİT değeri, ölçümü yapılan değerın çözünürlüğünü ve dolayısıyla ölçüm yapabileceğimiz minimum değer aralığını tayin eder.

Veri aktarım ortamları

Elektronik sistemlerde veri kablolar üzerinden aktarılır. Kablolar sinyal iletimi için gerekli iletişim ortamını sağlar. Haberleşme amacıyla kullanılacak kablo seçilirken aşağıdaki kriterler esas alınır.

- Aktarılabilecek verinin miktarı (aktarım hızı)
- Aktarım yöntemi (voltaj seviyesi)
- Aktarım mesafesi (kablo uzunluğu)
- Elektromanyetik gürültü (EMC güvenliği)
- Mekanik tasarım kriterleri

Elektronik veriler fiziksel olarak yukarıda belirtilen ortamlar üzerinden taşınır. Ancak taşınan verinin aktarım yöntemi farklı olabilir. Sinyallerin asimetrik veya simetrik olarak kablolar üzerinden voltaj

değeri cinsinden iletildiği durumlarda sinyal iletim mesafesi oldukça sınırlıdır. Optik haberleşme ve fiber-optik kablo kullanımı durumunda sinyal mesafesi oldukça artacaktır

Veri aktarımı yöntemleri

Veri aktarımı temel olarak iki alternatif yöntemle gerçekleştirilir. Gönderilecek sinyaller paralel veya seri olarak taşınır. Bu ayırım yapılırken taşınacak verinin kaç adet kablo ile yapılacağı ile ilgili herhangi bir sınıflandırma yapılmaz.

Paralel haberleşme

Verici ve alıcı taraflarında 8 adet bağlantı noktası bulunmaktadır. Bu tür bir bağlantı ile 8 BIT / 1 BYTE büyüklüğünde bir veri aktarımı yapılabilir. Sinyal aktarıldıktan belli bir süre sonra kesilerek bir başka sinyal moduna geçilmesi mümkündür.

Verici bölümü, herhangi bir PLC ünitesinin 8 adet çıkış verebilen çıkış modülüdür. Bu 8 adet vericinin karşılığında 8 adet alıcı mevcuttur. Konvansiyonel paralel kablolama olarak adlandırılabilir olan bu yöntemde PLC ünitesinin her bir çıkış noktası ayrı bir selenoid valfi kumanda eden röle kontaklarını kumanda eder. Bu durum, özellikle çıkış sayısı arttıkça problem teşkil etmeye başlar. Herhangi bir verici ucunda sinyal olduğunda 24V DC enerji kumanda edilen valfi çalıştırarak belirli bir işlevi yerine getirecektir. Tüm bobinlerin ortak bir geri dönüş (-) hattı mevcuttur. Tüm çıkışlar PLC tarafından her çevrim sonunda tekrar değerlendirilerek gerektiği taktirde değiştirilecektir. Valflerin enerjide kalma süresi yapılan programın içeriğine ve gönderilen sinyalin tipine ve süresine bağlıdır.

Seri haberleşme

Seri veri aktarımı durumunda da esasında bilgi paralel veri yongaları (BIT) olarak üretilir. Ancak bu bilgi tek bir veri hattına indirgenerek iletilecektir. Bu amaçla iletmek istediğimiz bilgi kodlanarak tek bir hat üzerinden iletilebilecek şekle getirilmelidir. Bu işleme paralel-seri data dönüşümü adı verilir. Bu dönüşüm ve iletim işleminin hızı (BIT/s = BAUD) aynı zamanda veri aktarım hızını belirler.

Seri haberleşme yönteminde sadece kontrol bilgilerinin yanısıra start/stop bilgileri, kontrol verileri ve bazı özel uygulamalarda güvenlik bilgileri de aynı hat üzerinden taşınır. Böylece birden fazla kablo kullanımının önüne geçilir. Ancak tek bir veri hattı üzerinden taşınabilecek verinin miktarı ve hızı sınırlı seri haberleşme kaba bir ifadeyle KODLAMA ve KOD ÇÖZME esasıyla verici ve alıcı konumundaki cihazların aynı ortak dili konuşarak anlaşmasıdır. Verinin kodlanması, gönderilmesi, çözülmesi ve kontrol işlevinin yerine getirilmesi belirli bir süre gerektirir. İlk bakışta bu dezavantajdan dolayı seri haberleşme, paralel haberleşmeye göre yavaş gibi görünebilir.

Ancak söz konusu gecikme, kodlama -kod çözme işlemleri kullanılan BUS sisteminin mimarisine ve kullanıcıların sayısına bağlı olmakla birlikte milisaniye mertebesinde (0.1-2ms) olduğu için pratik olarak ihmal edilebilir.

Haberleşme standartları

Tüm dünyada kabul edilen ve uluslararası standart haline gelmiş bazı haberleşme ara yüzleri vardır. Bütün üreticiler bu standartlar üzerinden haberleşme yaparak farklı marka ürünlerin birbirleriyle haberleşmelerine olanak tanır. RS , "Recommended Standard" yani tavsiye edilen standart kelimelerinin kısaltmasıdır.

RS232 standardı

RS232 genellikle iki nokta arasında asimetrik haberleşme yapıldığı durumlarda kullanılmaktadır. +3/+15V aralığı yüksek (HIGH) sinyal, -3/-15V aralığı düşük (LOW) sinyal olarak kabul edilir. Bu standart genellikle kişisel bilgisayarlarda kullanılmaktadır.

RS485 standardı

RS485 simetrik ve çok noktalı bağlantıya ihtiyaç duyulan uygulamalarda kullanılır. Simetrik haberleşmede iki veri hattı arasındaki diferansiyel gerilim ölçülür. Sinyal aktarımı yapılan bir noktadan gelen voltaj farkı negatif olduğunda sinyal yüksek (HIGH), pozitif olduğunda düşük (LOW) olarak kabul edilir. RS485 ile teknik olarak 32 alt sisteme bağlantı yapılabilir. Bundan daha fazla noktaya bağlantı yapılması tavsiye edilmemektedir.

Otomasyonun karmaşıklaşmasıyla birlikte kullanılan tahrik/kontrol ve algılama elemanlarının sayısı artmış ve buna bağlı olarak konvansiyonel terminal kablo bağlantılı projeler hem işçilik bakımından yüksek maliyetli, hem de operasyon esnasında hata bulma / giderme açısından verimsiz olmaya başlamıştır. Daha az kablo ile daha fazla veri taşıyabilen ve veriyi sadece iletmekle kalmayıp durum/konum bilgisini de aynı hat üzerinden gönderebilen seri haberleşme protokolleri bu ihtiyaca cevap olarak karşımıza çıkmaktadır.

Haberleşme protokolleri

Verici ve alıcı arasında hızlı, güvenli ve anlaşılabilir veri iletişimi gerçekleştirilebilmesi için kullanılan yazılım ve donanıma göre bazı unsurların belirlenmesi gerekir. Haberleşme ile ilgili kuralların belirlenmesi ile haberleşme protokolü belirlenmiş olur. Herhangi bir protokolda bilgi aktarımı ve kontrolü ile ilgili tüm detaylar belirlenmiş ve sabitlenmiştir. Alıcı ve verici cihazların belirli bir protokol üzerinden haberleşecek şekilde üretilmiş olması gerekmektedir.

Protokol, kontrol edilen sistemin tüm elemanlarının haberleştiği ve birbiriyle anlaştığı ortak konuşma dili olarak değerlendirilebilir. Örneğin PLC belirli bir işlevi yerine getirmek üzere oluşturduğu çıkış bilgilerini kodlayarak kullanılan protokol üzerinden diğer çevre birimi aygıtlarına gönderir. Kullanılan protokole ve kodlamaya uygun elektronik donanımı bulunan valf grubu gönderilen kodlanmış veriyi çözümledikten sonra içeriğine göre bazı silindirleri hareket ettirir. Daha sonra silindirler veya pozisyon ölçer cihazlardan gelen konum bilgileri aynı mantık ile kodlanarak geri gönderilir ve PLC bu geri besleme bilgilerini çözüp işleyerek kontrol işlemlerini devam ettirir.

Fieldbus (Alansal Veriyolu) sahada yani fabrika içi ve dışındaki alanlarda kullanılan tüm protokollerin genel adıdır.

Fieldbus protokolu dünyadaki otomasyon sistemlerinde yaklaşık % 80 'lik bir pazara sahip olan 140 şirketin biraraya gelmesi ile oluşturulan "Fieldbus Foundation" tarafından desteklenmektedir. Özellikle dağıtılmış proses kontrol uygulamaları için tasarlanmıştır. Teknolojisi fiziksel katman, haberleşme çatısı ve kullanıcı katmanından meydana gelmektedir.

Haberleşme protokolleri açık ve kapalı sistem olmak üzere ikiye ayrılmaktadır.

Açık sistem haberleşme protokolü: Markaya bağlı olmaksızın aynı amaç için üretilmiş tüm cihazların birbiriyle haberleşebilmesidir. Bu türde, Omron firmasının profibus ürünleri ile Phoenix firmasının profibus ürünleri birbiriyle sorunsuz haberleşebilir.

Kapalı Açık sistem haberleşme protokolü: Her üreticinin sadece kendi ürünleri arasında haberleşme sağlayacak şekilde geliştirdiği haberleşmedir. Mitsubishi Melsec, Omron Phoenix Contact Interbus-S Compobus gibi.

Açık sistem haberleşme protokollerine genel bakış

Interbus-S Protokolü: Interbus-S protokolu, Phoenix Contact tarafından geliştirilen ve özellikle Almanya'da çok yaygın olan açık mimarili ve DIN normlarına göre standartlaştırılmış bir BUS

sistemidir. Sistem kapalı halka topolojisi ile haberleşir. Veri iletişimi çift yönlü olarak gerçekleştirilir ve asıl haberleşme hattı alt seviye gruplara ayrılarak ölçeklenebilir.

Profibus Protokolü: Profibus geniş kapsamlı üretim ve proses otomasyonu için tasarlanmış üreticiden bağımsız açık saha hat protokolüdür. Üretici bağımsız oluşu ve açıklığı uluslararası standartlar olan EN 50170, EN 50254 ve IEC 61158 üzerine kurulmuştur. 650'ye yakın üyesi bulunan ve birçok araştırma enstitüsü tarafından desteklenen Profibus, farklı üreticilerin cihazları arasında haberleşme sağlayan ve bunu yaparken herhangi özel bir arabirime ihtiyacı olmayan bir veri yolu olmakla birlikte, yüksek hızlı kritik uygulamalar veya kompleks haberleşme işlemleri gibi kullanım alanlarında yaygın olarak uygulanan bir veri yolu sistemidir.

Modbus Protokolü: Modbus seri iletişim protokolü, master/slave ilişkisine sahip bir protokoldür. Sadece bir master düğümü (aynı zamanda olmak koşulu ile) ve maksimum 247 olmak üzere çeşitli sayılarda slave düğümü aynı bus (yol) üzerinde bağlanabilir. Modbus haberleşmesi her zaman master tarafından başlatılır. Slave düğümler Master düğüm tarafından bir istek emri almadığı sürece asla veri iletimi yapmaz. Slave düğümler kendi aralarında hiçbir şekilde haberleşemezler. Master düğüm aynı zaman içerisinde sadece tek bir Modbus iletişimi kurabilir.

CANBus Protokolü: Bosch firması tarafından geliştirilen (seri) veri yolu sistemi olan Controller Area Network protokolu, özellikle otomotiv sektörüne yönelik akıllı ağ, sensör ve aktüatörler için tasarlanmış ve kısa bir zamanda bu çalışmalarda standart hale gelmiştir. Multimaster yani bütün CAN noktalarının data iletebildiği ve birkaçının da eş zamanlı olarak istekte bulunabildiği veri yolu sistemi olan CANBus hiçbir abone ya da kullanıcı için herhangi bir adreslemeye sahip olmamakla birlikte öncelikli mesajın iletilmesi şeklinde veri iletir. Bu veri iletişim protokolu en sık Otomotiv ve Medikal endüstrisinde kullanım alanı bulmaktadır.

DeviceNet Protokolü: Allen-Bradley tarafından geliştirilen akıllı sensör ve aktüatörler için tasarlanmış endüstriyel iletişim protokolu olan DeviceNet "Open DeviceNet Vendors Association" adı verilen üretici bağımsız bir kuruluş tarafından günümüzde gelişimini sürdürmektedir. DeviceNet ile limit anahtar, fotoelektrik sensör, barkod okuyucu ve motor starterleri gibi düşük seviyeli aygıtlara bağlanılabilir ve PC veya PLC gibi daha üst seviyeli aygıtlarla haberleşme sağlanabilir.

AS-i Arayüzü: AS-i (Aktüatör -Sensör Arayüzü) olarak adlandırılan sistem en alt seviye otomasyon düzeyinde oldukça basit bir altyapıyla tahrik ve algılama elemanları üzerinde uygulanmaktadır. Paralel kablolamaya göre en basit alternatif olan sistem 11 farklı firma tarafından finanse edilen bir araştırma grubu tarafından geliştirilmiştir. Henüz tam anlamıyla standartlaşmaya gidilememiş olmakla beraber bu konudaki çalışmalar devam etmektedir. Yeni geliştirilen ürünlerle AS-i Bus üzerinden analog veri haberleşmesi de yapılmaya başlanmıştır AS-i son derece basit, ucuz ve aynı derecede güvenli bir sistemdir. Sistem merkezi kontrolör (master) ve buna bağlı maksimum 31 alt düzey kontrol sistemi bağlanabilir. AS-I denetleyicisi doğrudan ana otomasyon sistemine bağlanabildiği gibi bir başka sistemin alt sistemi olan mantıksal komponent grubu olarak ta bulunabilir. Özel veri dönüştürücüleri ile Profibus DP sinyalleri AS-i formatına çevrilerek kullanılabilir. AS-i sisteminde temel olarak algılayıcılardan gelen sinyaller işlenir ve aktüatörleri kumanda eden valfler kontrol edilir. Bu amaca hizmet eden standart giriş/çıkış modülleri vardır.

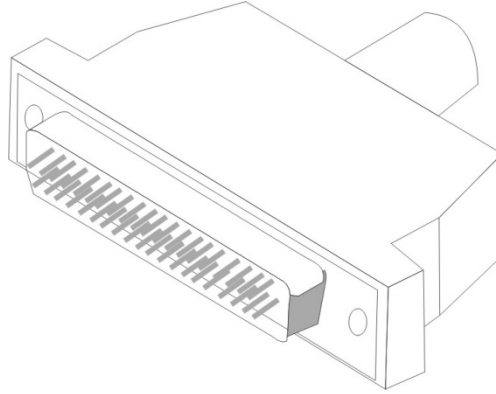
ENDÜSTRİYEL AĞLAR

Bilindiği gibi ağlar temel olarak kaynakların ortak kullanımı fikrinden ortaya çıkmıştır. Endüstriyel Ağlarda da aynı şekilde kaynakları ortak kullanabilir ve çeşitli üniteler arasında gerekli haberleşmeyi sağlayabiliriz. Günümüzde Endüstriyel Ağlar üzerinde kullanılmak üzere bir çok protokoller geliştirilmiştir. Bunlardan en popüler olarak kullanılanlarından bazılarını Modbus, Profibus, DeviceNet i sayabiliriz.

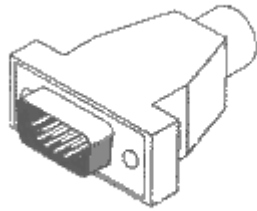
Genellikle endüstriyel ağların kurulumunda seri haberleşme standartları kullanılmaktadır. Bu standartlardan en yaygın olarak kullanılanlar RS232 standardı ve RS485 standardıdır. Bu standartlar ile ilgili kısa açıklamalar aşağıdaki gibidir.

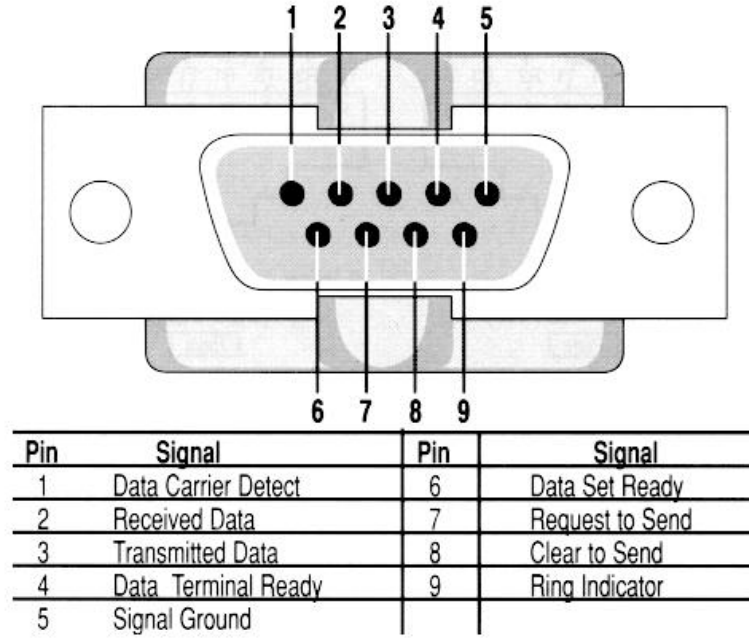
RS-232

Seri iletişim gerçekleştiren cihazlar arasında bu tür asenkron iletişimi sağlamak amacıyla geliştirilen bir haberleşme standardıdır. RS-232 standardı ilk olarak 1962 yılında çıkmıştır. Daha sonra 1969 yılında RS-232c standardı ortaya çıkmıştır. RS-232D standardı ise 1987 yılında RS-232C standardının geliştirilmesiyle ortaya çıkmıştır. Aynı zamanda bu standart EIA-232-D olarak da adlandırılmaktadır.



RS-232C standardında en çok kullanılan konektör DB25 ve 25 pinlidir konektör idi, ancak günümüzde 9 pinli DB 9 yaygın olarak kullanılmaktadır.





RS-232 EIA (Electronic Industries Association) tarafından geliştirilmiş bir standarttır. +15V ve -15V arasında iki voltaj seviyesi kullanarak 15 metreye kadar haberleşme için geliştirilmiştir. Modem, klavye ya da terminal gibi kısa mesafelerdeki birimlere sayısal veri aktarmak için kullanılır.

Veri genelde 8 bitlik karakterler halinde iletilir (ASCII karakterlerinin 8 bit tanımlanması nedeniyle). Ancak tercihe bağlı olarak 7 ve 5 e kadar daha düşük bitler halinde de iletilebilir. İletim seri yapılıdır (bitler ardışıl gönderilir). İletim standartta hem senkron hem asenkron olacak şekilde farklı farklı tanımlanmıştır. Asenkron (asynchronous) olması halinde gönderici ve alıcının koordine olması gerekmez.

Gönderici belli bir formatta hazırlanan veriyi hatta aktarır. Alıcı ise devamlı olarak hattı dinlemektedir, verinin gelişini bildiren işareti aldıktan sonra gelen veriyi toplar ve karakterleri oluşturur.

RS-232'de, eksi voltaj seviyesi "1", artı voltaj seviyesi "0" anlamındadır. Hattın boş olduğu eksi voltaj seviyesi ile ifade edilir. Veri gönderileceği voltajın artı değere çekilmesi (0, başlangıç biti) ile ifade edilir ve ardından karakter serisi gönderilir. Her bit için voltaj (1 için -3,-25 V ; 0 için +3,+25 V arası) belli bir süre aynı seviyede tutulur. Gönderici ve alıcı birimler bu süreye göre ayarlanmıştır. Her karakterin sonuna bir bitiş biti "1" eklenir.

RS-485

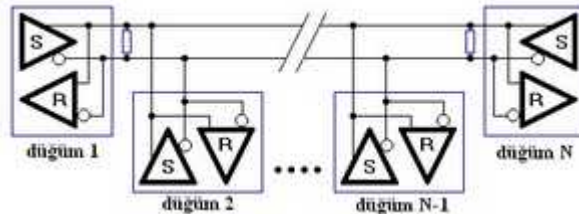
RS-485 yine bir seri haberleşme standartıdır. Sadece iki kablo ile haberleşme imkanı sağlar. Bu kablolardan biri Tx ucuna diğeri Rx ucuna bağlanmalıdır. RS485 arayüzü üzerinden çeşitli protokoller örneğin Modbus profibus gibi ile haberleştiğimiz yapabiliriz.. Rx ve Tx üzerindeki gerilimler arasındaki farka bakılarak sinyal durumları hesaplandığından ve gürültü bu iki adet kabloya aynı miktarda bineceğinden gürültülü sinyal ile normal sinyal arasında taşıdığı bilgi açısından bir fark olmamaktadır. ancak RS-232'de referans gnd (ground yani bildiğimiz sıfır voltaj temel referans) olduğundan gürültü kabloya bindiğinde sinyalin değerini artırıp azalttığından sinyalin taşıdığı bilgi değişim gösterebilmektedir.

Bu sayede RS-485 ile daha uzun mesafelerde güvenli haberleşme imkanı sağlanmıştır. Yine RS-485 ile ilgili ayrıntılı bilgi Bilgisayar Kontrol dersi modüllerinde verilmiştir.

RS485 EIA tarafından tanımlanmış çok yönlü bir seri haberleşme standartıdır. Birden fazla cihazın birbirleriyle haberleşmesi gereken veri işleme, ve kontrol uygulamalarında yoğun bir şekilde kullanılır.

RS232 nin en temel problemi sinyal hattı üzerindeki gürültüden kolay etkilenir olmasıdır. RS232 protokolü alıcı ve verici arasındaki data ve handshake line voltajlarını ortak bir toprak hattı kullanarak karşılaştırır. Toprak hattındaki herhangi bir voltaj artışı felaket sonuçlar doğuracaktır. Bu yüzden RS232 tetikleme seviyesi +/-3volta ayarlanmıştır. Bu nedenle mesafe arttığında gürültü hızla artar. RS485 standartında ise sinyal referansı için ortak sıfır kullanılmaz. Bu sebeple RS485 alıcı ve verici ünite arasındaki voltaj seviye farkı bir problem oluşturmaz. RS485 sinyalleri değişkendir ve her bir sinyal Sig+ ve Sig-hatları üzerinde iletilir. RS485 alıcısı sinyal hattı üzerindeki kesin voltaj seviyesi yerine iki hat arasındaki voltaj farkını karşılaştırır. Bu sayede bir çok haberleşme sorunun temeli olan toprak döngüsü önlenmiş olur.

RS485 in network yapısı data işleme ve kontrol uygulamalarında yoğun bir şekilde kullanılmasının ana nedenidir. 12 kohm giriş direnci ile networke 32 cihaza kadar bağlantı yapılabilir. Daha yüksek giriş direnciyle bu sayı 256 ya kadar çıkarılabilir. RS485 tekrarlayıcıları ile bağlanabilecek cihaz sayısı birkaç bine, haberleşme mesafesinde birkaç kilometreye çıkabilir. RS485 bunun için ayrıca bir donanım istemez yazılım kısmında RS232 den zor değildir.



Yukarıdaki resim RS485 network yapısını göstermektedir. N kadar düğüm çok noktalı RS485 networküne bağlanmıştır. Hattın iki ucundaki R dirençleri 100 ohm seçilerek yansıma önlenmiş olur böylece daha yüksek hız ve daha uzun mesafeye erişilmiş olur.

RS485 in belli başlı teknik özellikleri

Maksimum sürücü sayısı : 32

Maksimum alıcı sayısı : 32

Çalışma şekli : Half Duplex

Network Yapısı : Çok noktalı bağlantı

Maksimum Çalışma Mesafesi : 1200 metre

12 m kablo uzunluğunda maksimum hız : 35 Mbps
1200 m kablo uzunluğunda maksimum hız : 100 kbps
Alıcı giriş direnci : 12 kohm
Alıcı giriş duyarlılığı : +/-200 mvolt
Alıcı giriş aralığı : -7...12 volt
Maksimum sürücü çıkış voltajı : -7...12 volt
Minimum sürücü çıkış voltajı (yük bağlı durumda) : +/-1.5 volt

Endüstride Yaygın olarak Kullanılan Endüstriyel Bus Protokolleri.

Endüstride kullanılan Bus protokollerinden bazılarını Modbus, Profibus, DeviceNet, Canbus olarak sayabiliriz. Biz bu Modülümüzde kullanımı tamamen açık olan ve herhangi bir ücrete tabi olmayan Modbus ın RTU protokolünü anlatacağız.

MODBUS

Modbus 1979 yılında Modicon firması tarafından geliştirilmiş, sunucu istemci tabanlı olarak endüstride kullanılan akıllı cihaz olarak tabir edilen aygıtların haberleşmesi için geliştirilmiş bir protokoldür. Modbus protokolü tamamen açık ve günümüzde birçok endüstri kuruluşlarındaki yapıda kullanılan bir protokoldür. Farklı yapılardaki cihazlar Modbus ile haberleşebilmektedir.

Bazı verilere göre günümüzde dünya çapında 7 Milyon endüstri kuruluşu sistemleri içerisinde Modbus ı kullanmaktadırlar.

Modbus çoklu sahip/köle (master/slave) sistemlerini izlemek ve aygıtları programlamak için, akıllı cihazları (PLC, inverter vb.), algılayıcılar ve diğer aygıtları birbirleri ile haberleştirmek için veya alan içerisindeki cihazları uzaktan bilgisayar yada insan makine ara yüzleri ile kontrol edebilmek için kullanılabilir. Modbus cihazlardan verilerin alındığı ve bu verilerin bir merkezde toplanabildiği bir endüstriyel ağ sistemidir.

Standart bir Modbus ağında bir adet sahip (Master) ile birlikte 247 adet Köle (Slave) cihaz bulunabilir. Bir önceki cümleden de anlaşılacağı üzere köle cihazlar 247 adet olabilir adresleri de 1 ile 247 arasında verilebilir. Modbus ağında sahip cihaz diğer köle cihazdan aldığı verilere göre yine köle cihazları kontrol ettirebilir, diğer bir deyişle köle cihazlar içerisinde veri alıp bunlara veri yazılmasını sağlayabilir.

Modbus açık bir protokoldür. Bunun anlamı farklı üreticiler bu protokole uygun cihazları herhangi bir ücret ödemeksizin üretebilir ve bu protokolü kullanabilirler. Bu nedenle günümüzde endüstride yaygın olarak kullanılmaktadır.

Modbus ın esnekliği yüksektir. Sadece akıllı cihazlarla (PLC, Mikrobilgisayar) değil de gelişmiş sensörlere de direkt olarak uygulanabilmektedir.

Modbusın Çalışması :

Modbus seri haberleşme standardını kullanır. Orijinal olarak Modbus RS232 seri haberleşme standardını kullanmaktadır. Ancak bu sadece kısa mesafeler için geçerlidir. Uzun mesafeler için de RS485 seri haberleşme standardını da kullanabilmektedir.

Modbus da veriler birler ve sıfırlardan oluşan seriler halinde taşınmaktadır. Her bir bit bir voltaj seviyesi olarak aktarılır. Bu voltaj seviyeleri yani bitler çok hızlı bir şekilde

cihazlar arasında iletilir. Bu hız BAUD olarak adlandırılır. Örneğin 9600 baud (bit/saniye).

ModBus Komutları

Bobin Durumu Oku (Fonksiyon Kodu : 01)

Bu komut verilen adresteki köle (Slave) aygıtındaki bağımsız her bir bobin durumunu okumak için kullanılır. Kullanım şeklini aşağıdaki örnekte inceleyelim.

Örnek :

Bu örnekte 17 nolu adrese sahip cihazdan 17 inci bobinden itibaren 16 adet bobinin durumunu sorgulayan komutu Modbus RTU formatında yazalım. Burada CRC program tarafından otomatik olarak hesaplanarak komut sonuna eklenmelidir.

İstek : (Modbus Master Tarafından gönderilen)

11	01	0011	000F	CRC
İstek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Bobin Oku komut Kodu	Okunması istenen ilk bobinin Data Adresi (11Hex= 17. Bobin)	Durumu Okunacak Bobin Adedi (F Hex= 16 Adet Bobin)	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt)

11	01	02	CD6B	CRC
İstek Yapılan Aygıt Adresi	Bobin Oku Kodu	(16 Bobin / 8 Bit = 2 byte	CD : 17 ile 24 Arası Bobinlerin Durumu 6B :25 ile 32 Nolu Bobinlerin durumu	dönüşsel artıklık denetim

Bobinlerin Durumu :

Bobin	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Durum BIN	0	1	1	0	1	0	1	1	1	1	0	0	1	1	0	1
HEX	6				B				C				D			

Gelen Yanıt bilgisinde "CD6B" Hexadecimal olarak okunduğunda 17 inci çıkış bobini "1" , 22 inci bobin ise "0" olarak görülmektedir.

1. Giriş Durumu Oku (Fonksiyon Kodu : 02)

Bu fonksiyon her bir ayrı ayrı girişlerin durumlarını ON/OFF okumak için kullanılır. Giriş adresleri Modbus için #10001 den başlar. Aşağıdaki örnekte #10197 nolu adresten başlayarak 22 bitlik giriş durumu okutulmaktadır.

Örnek :

Bu örnekte ise 196 nolu girişten itibaren 22 adet girişin durumunu istek yapıyoruz. Okunması istenen ilk girişin data adresi hexadecimal olarak "00C4" olarak belirtilmiş, bunun onluk karşılığı 196 dır, ancak Modbus için girişler 10001 den başlamaktadır. İstek bilgisi gönderilirken bu adres bilgisi 10001+196=10197 olarak gönderilecektir.

İstek : (Modbus Master Tarafından gönderilen)

11	02	00C4	0016	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Giriş Oku komut Kodu	Okunması istenen ilk girişin Data Adresi (C4Hex = 196. giriş => #10197)	Durumu Okunacak Giriş Adedi (16 Hex = 22 Adet Giriş) 10197 den 10218 a kadar	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt)

11	02	03	ACDB35	CRC
Istek Yapılan Aygıt Adresi	Bobin Oku Kodu	(22 Giriş / 8 Bit = 3 byte)	AC : 10197 ile 10204 Arası Giriş Durumu DB : 10205 ile 10212 Nolu Giriş durumu 35 : 10213 ile 10218 Nolu Giriş Durumu	dönüşsel artıklık denetim

Bobinlerin Durumu :

AC: Ayrık Girişler 10197 – 10204 (**1010 1100**)

DB: Ayrık Girişler 10205 – 10212 (**1101 1011**)

35: Ayrık Girişler 10213 – 10218 (**0011 0101**)

Tabloya göre bu durumda 10197 nolu giriş “0” yani off durumdadır, 10204 nolu giriş ise “1” yani on durumdadır. Tabloya bakıldığında değeri 35 olan girişlerde ise 6 bitin kullanıldığı görülmektedir, bu altı bitin geriye kalan iki biti ise “0” olarak geri dönecektir.

2. Tutucu Registerleri Oku (Fonksiyon Kodu : 03)

Bu komut istek yapılan cihazın analog çıkış tutucu registerlerindeki verileri istek yapmak için kullanılır. Modbus için tutucu registerler adresleri #40001 den başlar.

Örnek: yine 17 nolu slave cihazın 107 nolu yani Modbus için #40108 nolu registerinden itibaren üç adet registerini istek yapan Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	03	006B	0003	CRC
Istek Yapılan Aygıt Adresi (11Hex = 17 Adres Nolu Aygıt)	Tutucu Register oku komutu	Okunması istenen ilk register Adresi (6B Hex = 107. nolu register=> #40108)	Durumu Okunacak registerAdedi (3 Hex = 3 Adet register) 40108 den 40110 a kadar	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt)

11	03	06	AE41 5652 4340	CRC
Istek Yapılan Aygıt Adresi	Tutucu register oku komutu	Veri byte larının sayısı 3 register => 3 x 2 byte= 6 byte)	AE41 : 40108 Nolu registerin içeriği 5652 : 40109 Nolu registerin içeriği 4340 : 40110 Nolu registerin içeriği.	dönüşsel artıklık denetim

3. Giriş Registerleri Oku (Fonksiyon Kodu : 04)

Bu komut ile giriş registerlerin durmuu istek yaptırabiliriz. Modbus için giriş register adresleri # 30001 den başlar.

Örnek: yine 17 nolu slave cihazın 8 nolu yani Modbus için #30009 nolu registerin durumunu istek yapan Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	04	0008	0001	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Giriş Register oku komutu	Okunması istenen ilk register Adresi (8 Hex= 8 nolu register=> #30009)	Durumu Okunacak registerAdedi (1 Hex= 1 Adet register) sadece 3009 nolu register	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt)

11	04	02	000A	CRC
Istek Yapılan Aygıt Adresi	Giriş registeri oku komutu	Veri byte larının sayısı 1 register => 1 x 2 byte= 2 byte)	30009 nolu registerin içindeki bilgi.	dönüşsel artıklık denetim

4. Sadece Bir bobin durumu değiştir (Fonksiyon Kodu : 05)

Bu komut ile ayrı bir bobinin durumunu değiştirebilirsiniz. Modbus için bobinler #1 den başlar.

Örnek: 17 nolu slave ünitenin 173 nolu bobinin ON yapmak için gerekli Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	05	00AC	FF00	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Bobin içeriği değiştir fonksiyon kodu.	Bobinin data adresi (AC hex= 172 => 173 Modbus için)	FF00 nun anlamı bobini ON konuma getir. OFF konum için bu bilgi 0000 olmalıdır.	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt)

11	05	00AC	FF00	CRC
Istek Yapılan Aygıt Adresi	Bobin içeriği değiştir fonksiyon kodu	Durumu değiştirilen bobin adresi	Durumu FF00 yani ON olarak değiştirildi mesajı	dönüşsel artıklık denetim

5. Sadece Bir Register durumunu değiştir (Fonksiyon Kodu : 06)

Bu komut ile bir registerin durumunu değiştirebilirsiniz. Modbus için bobinler #40001 den başlar.

Örnek: 17 nolu slave ünitenin 40002 nolu register içeriğini 0003 yapmak için gerekli Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	06	0001	0003	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Register içeriği değiştir komutu	1 hex=1 dec (40001+1=40002 nolu register)	40002 nolu registerin içeriğini 0003 hex yap.	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt) register içeriği yazıldıktan sonra aşağıdaki yanıt alınır.

11	06	0001	0003	CRC
Istek Yapılan Aygıt Adresi	Register içeriği değiştir fonksiyon kodu	Durumu değiştirilen register adresi 40002	40002 nolu register içeriği 0003 yapıldı.	dönüşsel artıklık denetim

6. Birden fazla Bobin içeriği değiştir (Fonksiyon Kodu : 15=0F hex)

Bu komut ile birden fazla bobinin durumunu değiştirebilirsiniz. Modbus için bobinler #1 den başlar.

Örnek: 17 nolu slave ünitenin 20 nolu bobin den itibaren 10 adet değerini değiştirmek için gerekli Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	0F	0013	000A	02	CD01	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	15 nolu Fonksiyon Bobinlerin içeriği değiştir komutu	13 Hex =20 nolu bobinden itibaren.	A hex=10 Dec 10 adet bobin datası değişecek.	10 adet bobin => 10 bobin/8 =2 byte değişecek.	CD :20 den 27 ye kadar bobinlere 01 : 27 den 29 a kadar bobinlerin durumu	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt bobin içeriği yazıldıktan sonra aşağıdaki yanıt alınır.

11	0F	0013	000A	CRC
Istek Yapılan Aygıt Adresi	Bobinlerin içeriğini değiştir fonksiyonu kodu.	Başlangıç bobin adresi	A hex : 10 adet bobin durumu değişti.	dönüşsel artıklık denetim

Yukarıdaki yanıt geldiğinde;

CD : 20 -27 arası bobinler (1100 1101) binary olarak değiştirilir.

01 : 27-29 arası bobinler (0000 0001) binary olarak değiştirilir.

7. Birden fazla Register'e Değer atamak (Fonksiyon Kodu : 16=10 hex)

Bu komut ile birden fazla registre değer atayabiliriz. Modbus için registerle #40001 den başlar.

Örnek: 17 nolu slave ünitenin nolu bobinin içeriğini değiştirmek için gerekli Modbus RTU komutunu yazalım.

İstek : (Modbus Master Tarafından gönderilen)

11	10	0001	0002	04	000A	0102	CRC
Istek Yapılan Aygıt Adresi (11Hex= 17 Adres Nolu Aygıt)	Fonksiyon kodu	İlk register adresi (40002)	2 adet register değışecek	2 adet register=> 2x2= 4 byte toplam	Register 40002 ye yazılacak değer	Register 40003 e yazılacak değer.	dönüşsel artıklık denetim

Yanıt : (Slave cihazdan alınan yanıt) register içeriğine yazıldıktan sonra aşağıdaki yanıt alınır.

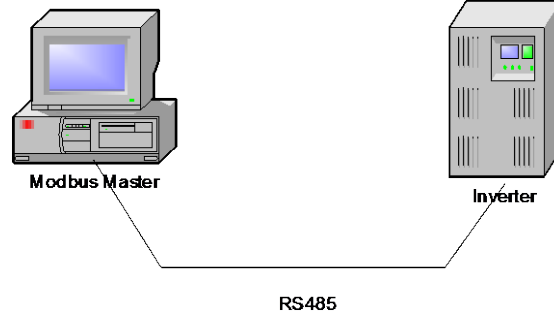
11	10	0001	0002	CRC
Istek Yapılan Aygıt Adresi	Fonksiyon Kodu	Başlangıç register adresi	2 adet register değeri değışti.	dönüşsel artıklık denetim

Modbus İle Modbus uyumlu Cihaz Haberleşmesi:

Modbus komutlarını ve yapısını öğrendik. Bu komutları ve uygulamaları kuracağımız farklı sistemlerde uygulayalım. Öncelikle Modbus ile Bilgisayar ve diğer modbus uyumlu cihazları haberleştirelim.

Modbus ile İnverter haberleşmesi:

Bu uygulamada aşağıdaki sistemi kuracağız.



Öncelikle bağlantıları yapacağız. Bilindiği üzere Modbus bağlantılarında RS485 Bağlantısı kullanarak çok daha uzun mesafelerde bağlantı yapılabilir. Bu amaç ile bilgisayardan USB/RS485 dönüştürücü modülü kullanacağız.



Şekil USB/RS485 dönüştürücü

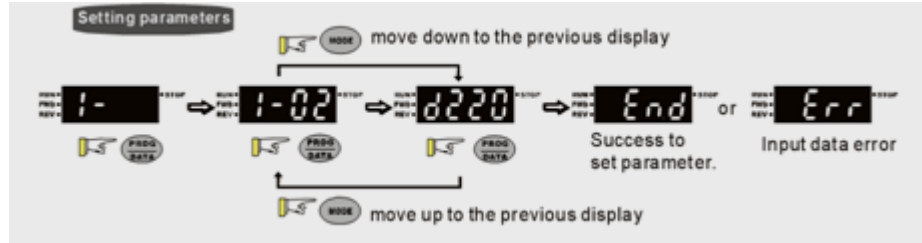
RS485 bağlantı uçlarından D+ ve D-Uçlarını kullanıyoruz. Bu kablounun diğer ucunu ise inverterin yine D+ ve D-Uçlarına çaprazlayarak bağlamalıyız. Yani USB/RS485 dönüştürücünün D+ ucunu inverterin D-bağlantı noktasına D-ucunu da inverterin D+ ucuna çaprazlayarak bağlıyoruz.



USB/RS485 dönüştürücü

İnverter Parametreleri:

Bu uygulamada kullanacağımız inverterin Delta VFD-S1 modeli. Modbus haberleşmesi yapılacak inverterin Modbus adresinin ve diğer parametrelerinin ayarlamasını üzerindeki panelden yapacağız. Kullanıcı klavuzunda bu ayarlamalar ve menü kullanımı verilmiştir.



Delta Ac Motor Sürücü Parametre Ayarlama Menüsünün kullanımı

Şimdi Modbus ayarlanmasını adım adım inceleyelim:

Chapter 5 Parameters | VFD-S Series

Pr.	Explanation	Settings	Factory Setting	NOTE
9-00	Communication Address	d1 to d254	d1	
9-01	Transmission Speed	d0: Baud Rate 4800 bps d1: Baud Rate 9600 bps	d1	
9-01	Transmission Speed	d2: Baud Rate 19200 bps d3: Baud Rate 38400 bps	d1	
9-02	Transmission Fault Treatment	d0: Warn and Keep Operating d1: Warn and Ramp to Stop d2: Warn and Coast to Stop d3: Keep Operating without Warning	d0	
9-03	Time-out Detection	d0: Disable d1 to d20: time setting (1 sec increment)	d0	
9-04	Communication Protocol	d0: 7,N,2 (Modbus, ASCII) d1: 7,E,1 (Modbus, ASCII) d2: 7,O,1 (Modbus, ASCII) d3: 8,N,2 (Modbus, ASCII) d4: 8,E,1 (Modbus, ASCII) d5: 8,O,1 (Modbus, ASCII) d6: 8,N,2 (Modbus, RTU) d7: 8,E,1 (Modbus, RTU) d8: 8,O,1 (Modbus, RTU)	d0	

9 Nolu parametrenin alt parametreleri ve anlamları Klavuzdan alındı

1. Adım



düğmesine bir kere bastıktan sonra yön tuşları ile 9 nolu parametreyi bulalım. 9 nolu parametre içerisinde bağlantı ayarlarını yapabiliriz.



Delta Ac Motor Sürücü Modbus Parametreleri

2. Adım: Aygıt adresi verme

Prog/data düğmesine tekrara basarak 9 nolu parametrenin alt parametrelerini ayarlayalım. Klavuzda belirtildiği üzere 9.00 parametresi ile cihaza bir d1 ile d2543 arasında bir aygıt adresi verebiliriz. Biz bu seçenekte aygıtımıza d3 parametresini ayarlayacağız. 9.00 ifadesini gördükten sonra tekrar “prog/data” düğmesine basıyoruz. Ve karşımıza gelen göstergede aygıt numarasını d3 olarak ok tuşu ile ayarlıyoruz.



Delta Ac aygıt adresi verme

Adres verme işlemini de prog/data tuşu ile onaylıyoruz.

3. Adım : Baud hızının ayarlanması



Delta Ac Motor sürücü bağlantı boud hızının set edilmesi 9.01 nolu parametre.

Ok tuşları ile 9.01 nolu parametreyi seçerek yine prog/data tuşuna basıyoruz. Elimizdeki klavuzda ki verilere göre bu parametre aşağıdaki değerlere karşılık gelen baud hızında haberleşmektedir. Biz bu baud hızlarından 38400 Baud rate yani d3 ü seçeceğiz.

Veri	Değer (Boud Rate)
d0	4800
d1	9600
d2	19200
d3	38400



Delta Ac Motor sürücü boud hızı d3 yani 38400 olarak seçildi.

Seçimi kabul etmek için yine prog/data düğmesine basıyoruz.

4. Adım: İletişim Protokolünün seçimi.

Bu işlem için menüde 9.04 nolu parametreyi bulup prog/data ile bu parametreyi ayarlıyoruz.



Delta Ac Motor iletişim protokolü ayarlama 9.04 nolu parametre.

Klavuzdan baktığımızda bu parametreyi d7 yaptığımızda Modbus RTU modunda <8,E,1> 8 bit çift parite 1 stop biti şeklinde ayarlanmış oluyoruz.



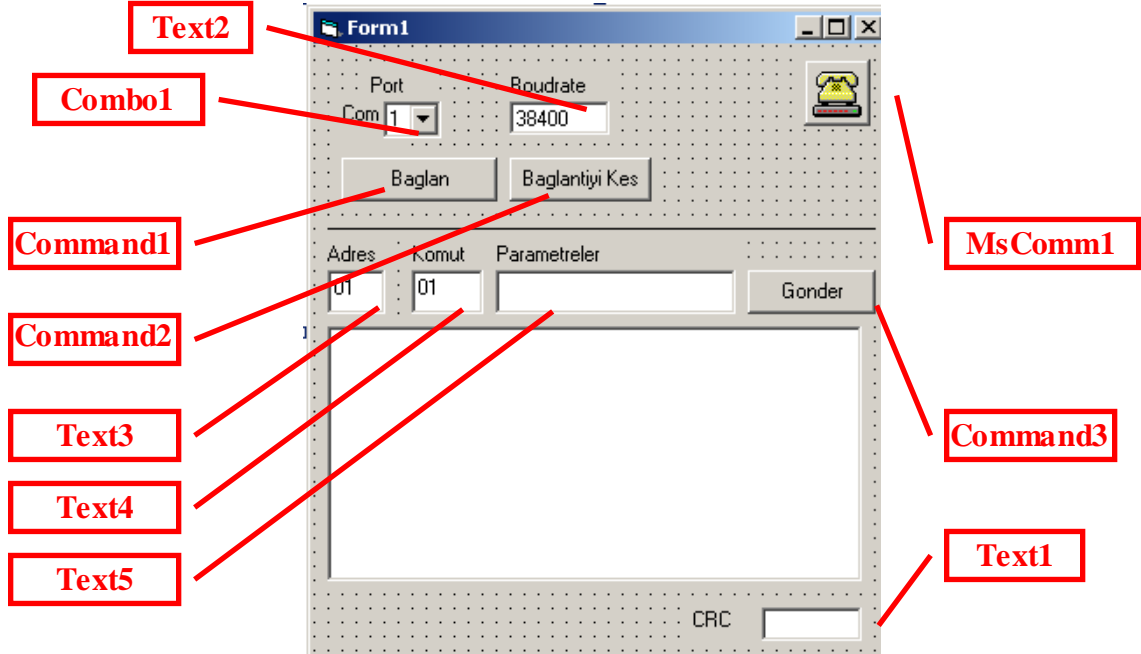
Delta Ac Motor iletişim protokolü ayarlama 9.04 nolu parametre.

Bu parametre ayarını sonlandırmak içinde yine prog/data düğmesine basıyoruz. Ve son olarak da Mode düğmesine basarak ana menüye dönüş yapıyoruz.

Parametre ayarlarını yaptığımız inverteri kontrol etmek amacı ile bilgisayarımızda visual basic 6.0 kullanarak bir program yazacağız. Bu programı diğer uygulamalarda da kullanacağız.

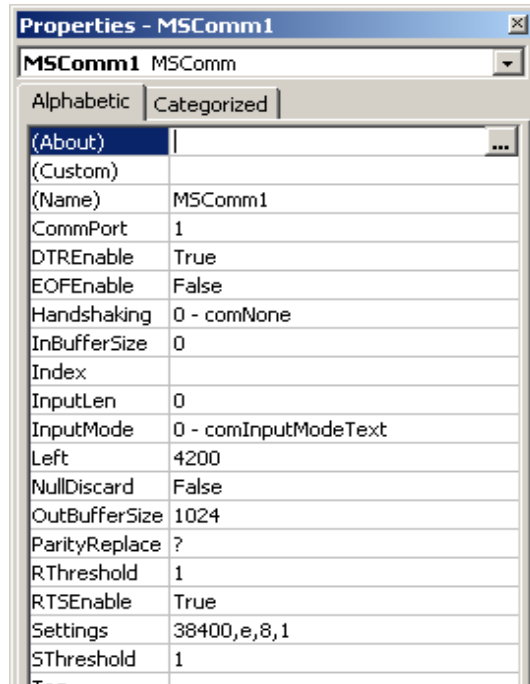
Visual Basic ile Modbus Master Programı yapımı:

Herhangi bir Modbus uyumlu cihaza Modbus rtu komut yapısı kullanmak amacı ile Visual Basic 6.0 kullanarak bir arayüz ve kodlarını oluşturalım. Öncelikle aşağıdaki formu oluşturalım. Seri iletişimde MsComm nesnesini kullanacağız.



Modbus için hazırlanan arayüz

Form üzerindeki MsComm nesnesinin özellikleri şunlardır.



Mscomm1 Nesnesinin Özellikleri

Form Load Olayı :

Bu bölümde Form yüklendiği anda butonların durumları ve ekranda bulunan combo kutusunun elemanları belirlenmektedir.

```
Private Sub Form_Load()  
  
Command2.Enabled = False  
Combo1.AddItem 2  
Combo1.AddItem 3  
Combo1.AddItem 4  
  
End Sub
```

Command1 Click Olayı :

Command1 nesnesi üzerinde bağlan yazısı görülmektedir. Bu düğmeye bakıldığında aşağıdaki kodda belirlenen ve diğer nesnelerden gelen port numarası ve Boud hızına göre seri iletişim kurulmaktadır.

```
Private Sub Command1_Click()  
  
MSComm1.CommPort = Val(Combo1.Text)  
MSComm1.Settings = Text2.Text + ",e, 8, 1"  
MSComm1.PortOpen = True  
Command1.Enabled = False  
Command2.Enabled = True  
End Sub
```

Command2 Click Olayı

Command2 düğmesi üzerinde "bağlantıyı kes" yazmaktadır. Bu düğmeye basıldığında seri iletişim bağlantısı kesilecektir.

```
Private Sub Command2_Click()  
  
Command2.Enabled = flase  
Command1.Enabled = True  
MSComm1.PortOpen = False  
End Sub
```

Command3 Click Olayı:

Bu kısımda da "gönder" düğmesine basıldığında yapılması istenenler kodlanmıştır. Buna göre girilen hexadecimal parametreler birbirine eklenerek Modbus komut satırı oluşturulmaktadır. Ayrıca Bu kısımda girilen komuta göre otomatik olarak CRC hesaplanması da yapılmıştır.

```

Private Sub Command3_Click()
Dim verigir As String
Dim uz, msb, reg_crc, verigir(20) As Integer
verigir = Text3.Text + Text4.Text + Text5.Text
uz = Len(verigir)
uz1 = (uz / 2) - 1
ReDim veri(uz1 + 2) As Byte
ii = -1
For i = 1 To uz Step 2
    ii = ii + 1
    veri(ii) = Val("&H" & Mid(verigir, i, 2))
Next
    Text6.Text = Text6.Text + " "
    ii = 0
    jj = (uz / 2)
    reg_crc = &HFFFF
Do Until jj = 0
    reg_crc = reg_crc Xor veri(ii)
    ii = ii + 1
    For j = 0 To 7
        If (reg_crc And &H1) Then
            msb = reg_crc And &H8000
            reg_crc = ((reg_crc And &H7FFF) \ 2)
            If msb Then reg_crc = reg_crc Or &H4000
            reg_crc = reg_crc Xor &HA001
        Else
            msb = reg_crc And &H8000
            reg_crc = (reg_crc And &H7FFF)
            reg_crc = (reg_crc \ 2)
            If msb Then reg_crc = reg_crc Or &H4000
        End If
    Next
    jj = jj - 1
Loop
veri(uz1 + 1) = reg_crc And &HFF
For i = 0 To 7
    msb = reg_crc And &H8000
    reg_crc = (reg_crc And &H7FFF)
    reg_crc = (reg_crc \ 2)
    If msb Then reg_crc = reg_crc Or &H4000
Next
veri(uz1 + 2) = reg_crc And &HFF
Text1.Text = Hex(veri(uz1 + 1)) + Hex(veri(uz1 + 2))
MSComm1.Output = veri
End Sub

```

MsComm1 OnComm olayı :

MsComm OnComm olayı bilgisayarın seri portuna herhangi bir bilgi geldiğinde aktif olacaktır. Bu bölümde gelen bilgi text6 kutusuna hexadecimal olarak yazdırılacaktır.

```

Private Sub MSComm1_OnComm()
Dim i As Integer
Dim ResStrLen As Integer
Dim ResStr As String
Select Case MSComm1.CommEvent
Case comEvReceive:
ResStr = MSComm1.Input
ResStrLen = Len(ResStr)

For i = 1 To ResStrLen
If Len(Hex(AscB(Mid(ResStr, i, 2)))) < 2 Then
Text6.Text = Text6.Text & ",0" & Hex(AscB(Mid(ResStr, i, 2)))
Else
Text6.Text = Text6.Text & "," & Hex(AscB(Mid(ResStr, i, 2)))
End If
Next

End Select
End Sub

```

Modbus RTU protokolü ile Inverter parametrelerini değiştirme:

Sistem kurulumu ve programlar hazırlandıktan sonra şimdi de inverterin sırası ile frekans bilgisini değiştirerek motorun ileri geri ve durdurma işlemlerini komutlar ile sağlayalım.

A .Frekans parametresini değiştirme :

Kullandığımız Delta inverterin kullanım kılavuzunda hangi register alanına hangi bilgiyi gönderdiğimizde ne tür değişiklikler olacağı belirtilmiştir. Örnek olarak öncelikle Frekans bilgisini değiştirelim. Kılavuzda ileri geri ve durdurma işlemleri için değiştirilecek register adresi 2000H olarak verilmiştir. Frekans değişiklikleri içinde 2001H adresli registeri değiştirmek yeterli olacaktır. Register adresleri ve hangi bilgide ne değişiklikler olacağı kılavuzda bulunan aşağıdaki tabloda verilmiştir.

Chapter 5 Parameters | VFD-S Series

Content	Address	Functions	
Command	2000H	Bit 0-1	00: No function 01: Stop 10: Run 11: Jog + Run
		Bit 2-3	Not used
		Bit 4-5	00: No function 01: FWD 10: REV 11: Change direction
		Bit 6-15	Not used
	2001H	Freq. command	
	2002H	Bit 0	1: EF (external fault) on
		Bit 1	1: Reset
		Bit 2-15	Not used

Şekil Inverter 2000H ile 2002H nolu registerlerin değer listesi

İlk olarak frekans registerini değiştirelim. Bunun için Modbusun 10 nolu komutunu kullanalım. Komut düzeni aşağıdaki gibi olacaktır.

Bilgi	Açıklama
03	Komut gönderilen cihazın Hexadecimal olarak adresi
10	Birden fazla registre birden değer atamak için kullanılan Hexadecimal komut. Decimal karşılığı 16
2001	Register adresi (Bu adresi Kullanıcı klavuzundan elde ettik)
0001	Bir adet Register değişecek
02	1 adet register toplam byte olarak (2 byte)
0DAC	Registere yazacağımız verinin Hexadecimal karşılığı (0DAC Hex= 3500 Decimal)
9BCE	CRC bilgisi. Bu veri otomatik olarak programımız tarafından hesaplanarak inverttere gönderildi.

Şimdide bağlantıları yaptıktan sonra visual basic de hazırladığımız programı çalıştırarak komutu gönderelim.

Inverter 2001H adresine 0DACHex=3500 yani 35 herz bilgisi gönderme

Hatırlayacağınız üzere 10Hex (16 Dec) komutu çoklu olarak register içerisine bilgi yazdırabildiğimiz komuttu. Yukarıdaki örneğimizde bir önceki bölümde adres bilgisi olarak 03 şeklinde ayarladığımız inverterin kullanıcı kılavuzundaki listeden aldığımız üzere 2001 nolu registerine hexadecimal olarak "0DAC" yani 3500 Herz bilgisini yazdık. Şimdide aşağıdaki tabloda gelen yanıtı inceleyelim.

Bilgi	Açıklama
03	Adres
10	Komut
2001	Register adresi hexadecimal
0001	1 Byte Değiştirildi
5A2B	CRC Kontrol Bilgisi Hexadecimal

Görüldüğü üzere gönderilen komut karşılığında istenilen yanıt ulaştı. Şimdide inverterin göstergesine baktığımızda frekans bilgisini okuyalım.

B. İnverterin RUN konuma alınması

Kılavuzdan Register adres listesine bakıldığında 2000H registerindeki datalar ile inverterin RUN, STOP, JOG + RUN Moduna alınabildiğini ayrıca FWD "ileri", REV "geri" şeklinde motoru sürebileceğini görebiliriz.

2000H	Bit 0-1	00: Fonksiyonu Yok
		01 : Dur
		10 : Run
		11: Jog+ Run
	Bit 2-3	Kullanılmaz
	Bit 4-5	00: Fonksiyonu Yok
		01: İleri
		10: Geri
		11: Yön Değiştir
	Bit 6-15	Kullanılmaz

Şimdiki örneğimizde İnverteri Motoru ileri yönde RUN konumuna aldıralım. Bunun için 1 nolu biti ve 4 nolu biti 1 yapmamız yeterli olacaktır. Aşağıdaki tabloda bit düzenini ve Hexadecimal olarak değerini görelim.

Bit Numarası	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Değer Binary	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Değer Hexadecimal	0012 Hex															

Şimdide Modbus komutunu hazırlayalım.

Bilgi	Açıklama
03	Komut gönderilen cihazın Hexadecimal olarak adresi
06	Sadece bir registeri değiştireceğiz
2000	Register adresi (Bu adresi Kullanıcı klavuzundan elde ettik)
0012	Registerin değeri 12H olacak.
03E5	CRC Hazırladığımız program tarafından hesaplanacak.

Programımızı kullanarak inverterdeki registeri değiştirelim.

The screenshot shows a software window titled 'Form1' with the following elements:

- Port:** A dropdown menu showing 'Com 4'.
- Boudrate:** A text box containing '38400'.
- Buttons:** 'Baglan' and 'Baglantiyi Kes'.
- Adres:** A text box containing '03'.
- Komut:** A text box containing '06'.
- Parametreler:** A text box containing '20000012'.
- Gonder:** A button to send the command.
- Text Area:** Displays the command string '.03,06,20,00,00,12,03,E5'.
- CRC:** A field at the bottom right showing '3E5'.

Gönder Tuşuna basıldığında inverterin FWD ileri ve RUN konumuna geçtiğini göreceksiniz.

C. İnverter Frekans Bilgisinin Okunması :

Modbus Protokolünü kullanarak inverterin frekans bilgilerini ve RUN konuma geçişini kontrol edebildiğimiz gibi o anda set edilen frekans bilgilerini ve ayrıca Akım gerilim gibi bilgileri de okuyabiliriz. Kılavuzdan faydalanarak bu bilgilerin hangi register adresinde olduğunu görebiliriz. Bi sonraki örneğimizde inverterden daha önceden set edilmiş frekans değerini okutalım. Frekans bilgisi uygulamada kullandığımız inverterin 2102H nolu adresinde tutulmaktadır. Şimdi uygun Modbus komutunu hazırlayalım.

Bilgi	Açıklama
03	Aygıt Adresi
03	Komut Yalnız bir register oku
2103	Hexadecimal olarak okumak istenilen register adresi
0001	Kaç adet registerin okunacağı
7FD4	CRC

Şimdi de hazırladığımız veriyi programımızı kullanarak invertere gönderelim ve cevabı inceleyelim.

Şekil: Frekans Bilgisinin inverterden okunması.

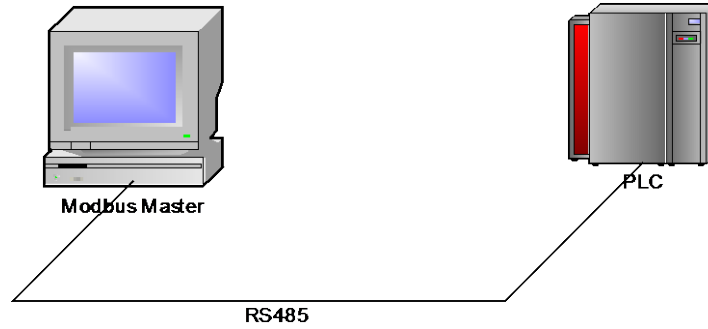
Bilgi	Açıklama
03	Aygıt Adresi
03	Komut
02	Inverterden gelen bilgi uzunluğu 2 byte (1 register)
0DAC	Register içeriği. (0DACH=3500)
C569	CRC

Yukarıdaki örnekten de anlaşılacağı gibi inverterde o anda set edilen frekans bilgisinin 3500 yani 35 Hertz olduğunu okumuş olduk. Bu veriler ışığında ekrana programlama üzerinde değişiklik yaparak gelen bilgiyi hexadecimal olarak değil de hertz bilgisi olarak da yazmak mümkün olabilecektir.

Görüldüğü üzere Modbus üzerinden Modbus uyumlu inverter cihazı ile haberleşmiş olduk. Bu uygulamada örnekleri arttırmak da mümkündür. İnverterdeki registerlerin değerlerini değiştirerek motoru ileri veya geri yönde istenilen hızlarda hareket ettirebileceğimiz gibi o anki hareket yönünü hızını ve motor tarafından çekilen akımı da ekranda gösterebiliriz. Bu sayede bir Scada sistemi de tasarlamamız mümkün olacaktır.

Modbus ile PLC haberleşmesi:

Bu uygulamamızda aşağıdaki sistemi kuracağız.

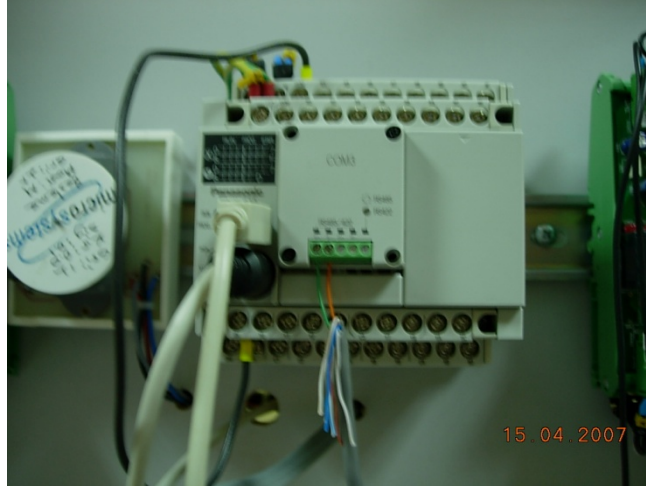


Bu uygulamamızda yine bilgisayarı Modbus Master olarak Modbus uyumlu NAIS PLC ile haberleştireceğiz. Yine visual basicte inverter ile haberleşmek amacı ile oluşturduğumuz vbasic programını kullanacağız. Bu haberleşmede de yine USB/RS485 dönüştürücüyü kullanacağız. PLC üzerinde de RS485 bağlantı modülü kullanacağız.



NAIS PLC RS485 arayüzü

Bilgisayarımızın USB ucuna bağladığımız USB/RS485 dönüştürücünün RS485 çıkışındaki D+ ucunu PLC RS485 ara yüzündeki S+ ucuna bağlıyoruz, D-ucunu ise yine PLC RS485 ara yüzünün S-ucuna bağlıyoruz. Ara yüzü PLC ye bağlıyoruz ve PLC katalogundaki bilgiler ışığında PLC güç bağlantılarını da yapıyoruz.

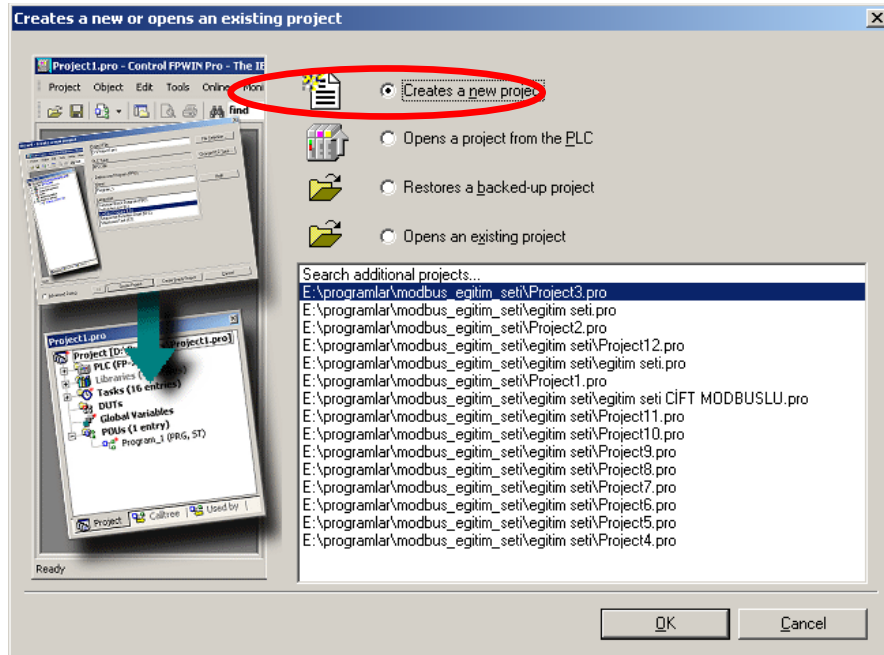


PLC RS485 arayüzü bağlantısı

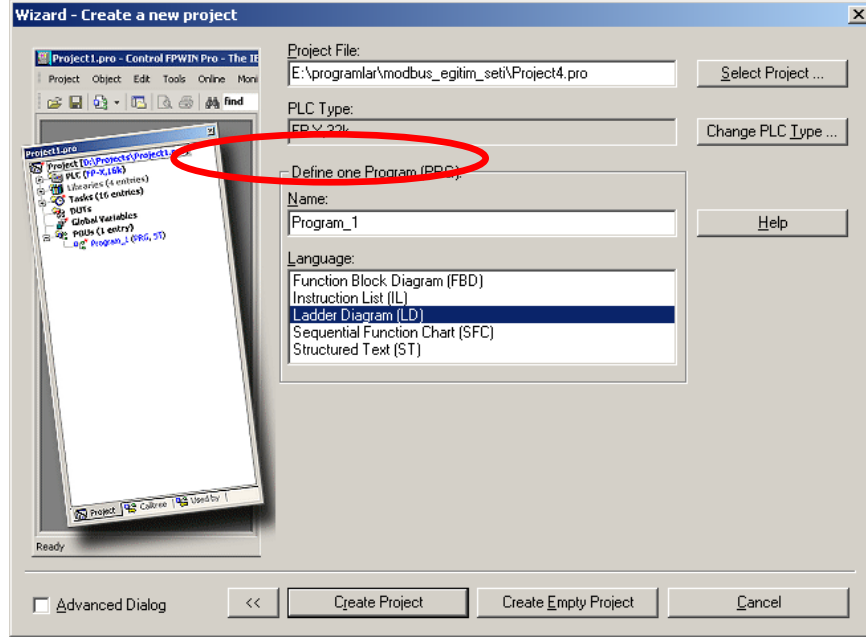
Bağlantı hazırlandıktan sonra artık PLC nin Modbus ile ilgili ayarlarını yapacağız. NAIS PLC yi programlamak için kullandığımız FPwin pro programını bilgisayarımıza kuruyoruz. Ayrıca PLC ile bilgisayarımızı USB kablosu ile bağlıyoruz. Bu kablo PLC içerisine program yazmak ve konfigürasyon yapmak için verilen kablodur.

PLC Konfigürasyonu

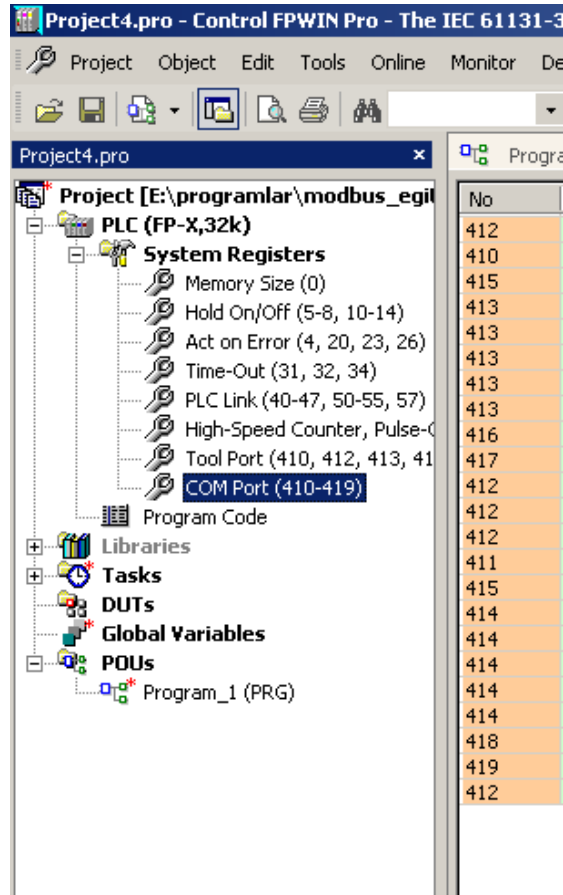
PLC nin Modbus haberleşmesi yapabilmesi için öncelikle bazı konfigürasyonların yapılması gerekiyor. Bunun için önce FPwin programını başlatalım.



Karşımıza gelen pencereden yeni bir proje seçeneğini seçiyoruz.

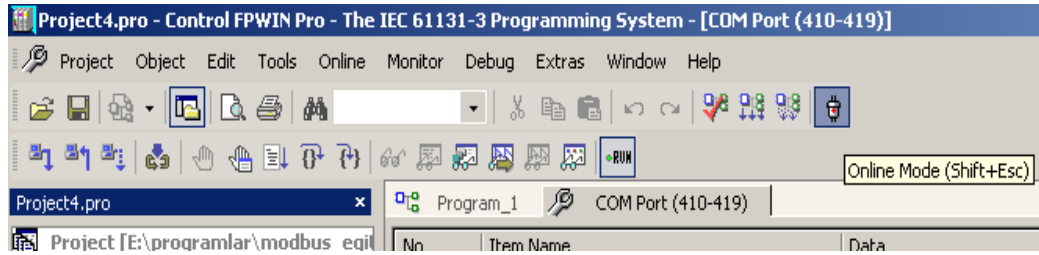


PLC tipini seçerek Create Project düğmesine basıyoruz. Karşımıza gelen proje tasarım ekranında sağ taraftaki Project gezgin penceresinden sistem registeri seçerek Com port ayarlarına geliyoruz ve parametreleri aşağıdaki şekilde yapıyoruz.

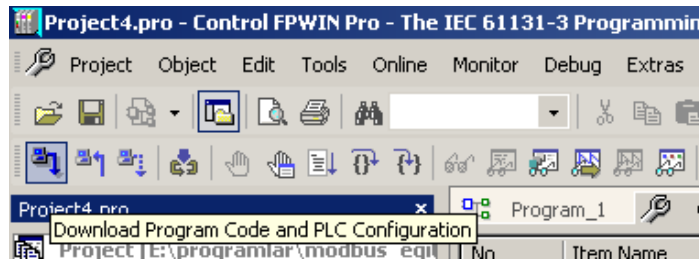


No	Item Name	Data	Time...	Range	Additional
#12	COM port 1 communication mode	Modbus RTU Master/Slave		MEWTOCOL-...	The PLC c
#10	COM port 1 unit number	1		1 to 99	Station n
#15	COM port 1 baud rate	38400	baud	115200	Specifies
#13	COM port 1 sending data length	8 bits		8 bits	Selects th
#13	COM port 1 sending parity check	With-Even		None	Selects th
#13	COM port 1 sending stop bit	1 bit		1 bit	Specifies
#13	COM port 1 sending header	No-STX		No-STX	Selects th
#13	COM port 1 sending terminator/reception d...	CR		CR	Selects th
#16	COM port 1 data register starting address ...	0		0 to 32762	Specifies
#17	COM port 1 receive buffer capacity	0		0 to 2048	Specifies
#12	COM port 1 modem connection	Disable		Disable	Specifies
#12	COM port 2 port selection	Internal USB port		Internal USB ...	
#12	COM port 2 communication mode	MEWTOCOL-COM Slave [...]		MEWTOCOL-...	The MEW
#11	COM port 2 unit number	1		1 to 99	Station n
#15	COM port 2 baud rate	9600	baud	115200	Specifies
#14	COM port 2 sending data length	8 bits		8 bits	Selects th
#14	COM port 2 sending parity check	With-Odd		None	Selects th
#14	COM port 2 sending stop bit	1 bit		1 bit	Specifies
#14	COM port 2 sending header	No-STX		No-STX	Selects th
#14	COM port 2 sending terminator/reception d...	CR		CR	Selects th
#18	COM port 2 data register starting address ...	2048		0 to 32762	Specifies
#19	COM port 2 receive buffer capacity	0		0 to 2048	Specifies
#12	COM port 2 modem connection	Disable		Disable	Specifies

Görüldüğü üzere haberleşme türünü Modbus RTU mod seçtik ve adresini 1 olarak verdik. Com Baud hızını da 38400 olarak verdik. 8 bitlik çift pariteyi seçtikten sonra artık bu konfigürasyonu PLC ye yazabiliriz. Bağlantıda da herhangi bir problemimiz yok ise PLC ile online moda geçiş yapabiliriz.



Online moda geçtiğimize göre artık konfigürasyonu PLC ye yazdırabiliriz. Bunun için Download program code düğmesini kullanıyoruz.



Program yazımı işlemi bittikten sonra PLC ile Modbus haberleşme yapmaya hazırız demektir.

Program ile Test Etme

Programı PLC ye yazdıktan ve bağlantıları yaptıktan sonra artık PLC ile Bilgisayarımızı Modbus RTU Protokolü ile haberleştirebiliriz. Bu amaçla öncelikle visual basic te daha önce hazırladığımız programı başlatıyoruz. Ve aşağıdaki komut dizisini gönderiyoruz.

Komut dizesini çözümleyelim

01: Adres Bilgisi

02 : Giriş Oku komutu

0000 : Okunacak ilk giriş adresi

0008 : 8 Hex = 8 Decimal adet girişi oku

79CC : CRC (programımız tarafından otomatik oluşturuldu)

Şimdide cevap bilgisini yorumlayalım

01: Adres Bilgisi

02 : Komut Bilgisi

01: 01 Hex = 1 Decimal yani 1 byte lık bilgi gönderildi

00 : Girişlerin hepsi sıfır (0000 0000)

A1C6: Cihaz tarafından üretilen CRC

Şimdide PLC üzerideki Y0 ve Y1 bobinlerini aktif edelim, yani ON konumuna getirelim. Bunun için Modbus ın 15 nolu komutunu yani 0F hex nolu komutunu kullanacağız.

Komut Dizesi

01 :Aygıt adresi
 0F : Modbusun birden fazla bobin konumu değıştir komutu
 0000 : Bobin başlangıç adresi
 0008 : 0008 Hex yani 8 decimal adet bobini değışecek
 01 : 1 byte lık veri göndeirlecek
 03 : Gönderilen bilgi (0000 0011) Y0 ve Y1 Bobinlerine 1 gönderiliyor.
 BE94 : Program tarafından hesaplanan CRC

Cihazdan gelen yanıt

01: Aygıt adresi
 0f : Komut
 0000 :başlangıç bobin adresi
 0008 : 8 adet bobin durumu değıştirildi.
 540D : Aygıt tarafından gönderilen CRC

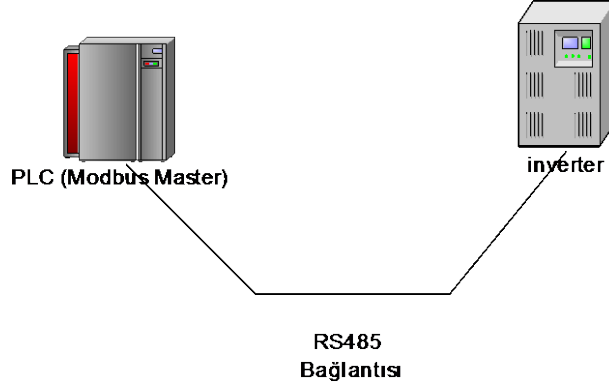
PLC üzerindeki bobinlerin durumunu gösteren LED ler baktığımızda Y0 ve Y1 Bobinlerinin aktif olduğunu göreceksiniz.



Bu bölümde basit olarak Modbus haberleşme ile ilgili bilgileri aldık ve komutlar ile birlikte çeşitli cihazları bilgisayar ile haberleştirdik. Bu komutları çok daha farklı cihazları da haberleştirmek mümkündür. Modbus haberleştirmede sadece bir bilgisayarı master olarak kullanabileceğimiz gibi Modbus fonksiyonlarına sahip PLC gibi akıllı cihazları da kullanabiliriz. Bir sonraki öğrenme faaliyetinde bunun üzerinde örnekler yapacağız.

Modbus Master Olarak PLC kullanımı :

Bu uygulamamızda farklı olarak bilgisayar yerine bu defa akıllı cihazımız olan PLC yi Modbus master olarak kullanacağız.



Daha önceki bölümde anlatıldığı gibi PLC ve inverter ayarlarını aynen koruyoruz. Ayarlar aşağıdaki gibi olacak.

Inverter için ;

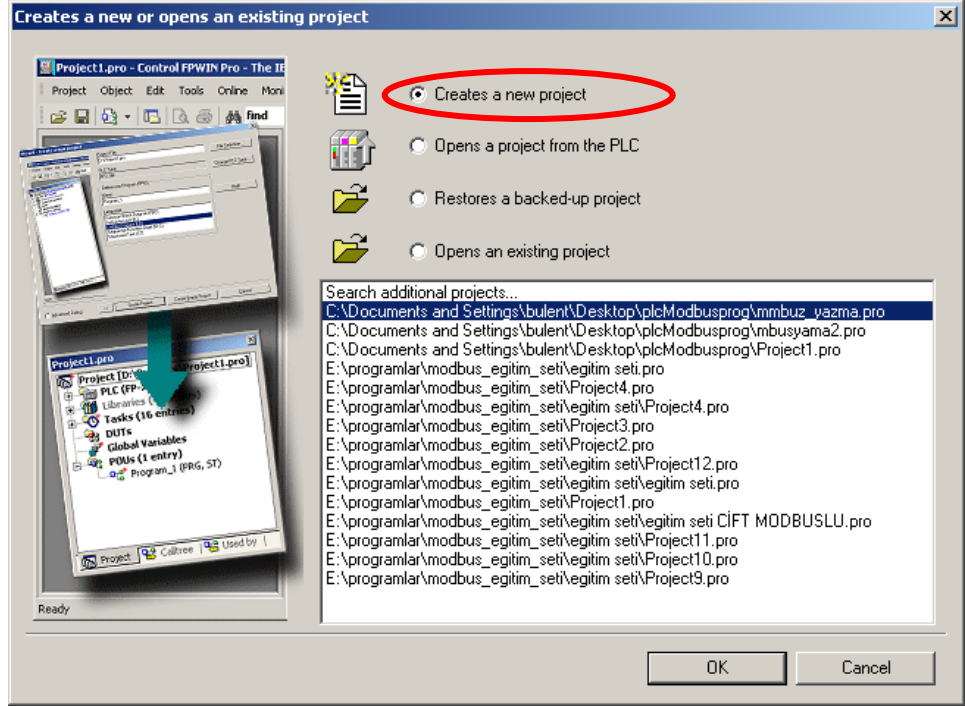
Modbus Adres : 03
Haberleşme Hızı : 38400 (baudrate)
Parite : Even
Stop bit :1 Bit
Data Uzunluğu : 8 Bit

PLC için :

Modbus Adres :01
Haberleşme Hızı: 38400 (baudrate)
Parite : Even
Stop bit : 1 Bit
Data uzunluğu : 8 Bit

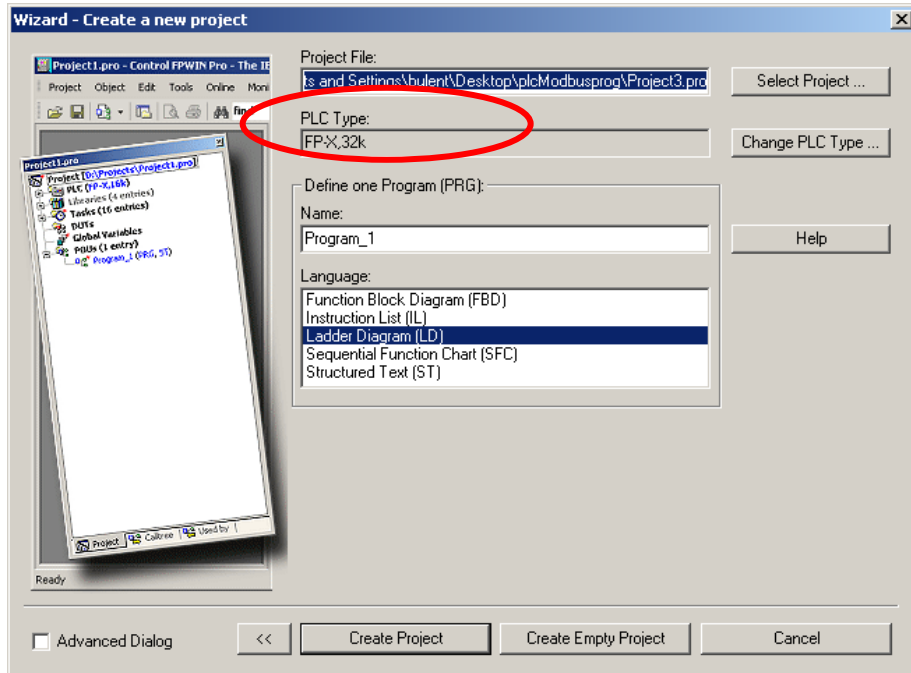
PLC Programı :

PLC yi programlamak için daha önceden de anlatıldığı gibi Panasonic NAIS firmasının PLC programlama arayüzü olan FpWinPro yu kullanıyoruz. Yeni bir Proje oluşturalım .

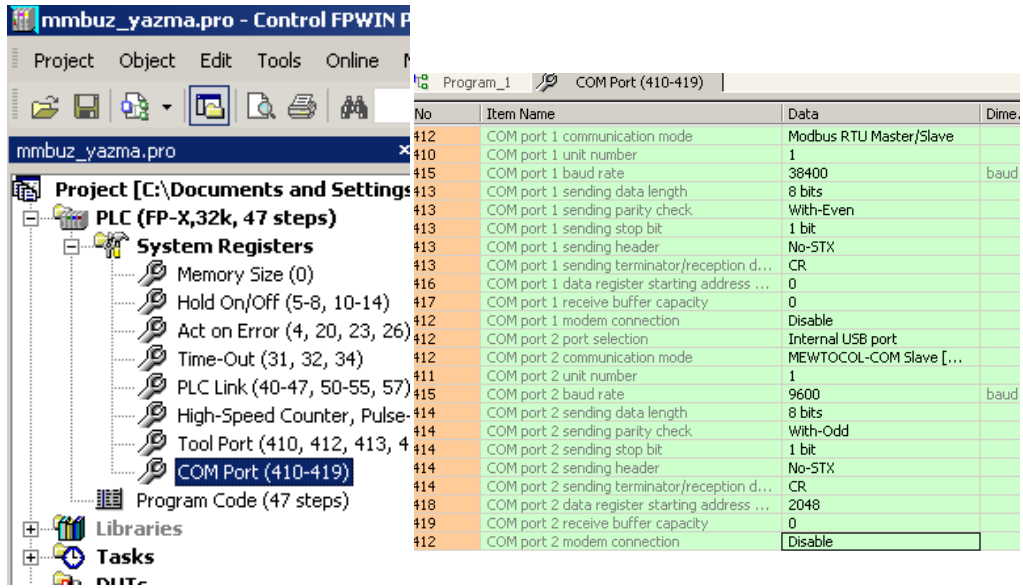


Yeni bir proje oluşturma penceresi

Bi sonraki pencerede PLC tipini seçiyoruz.



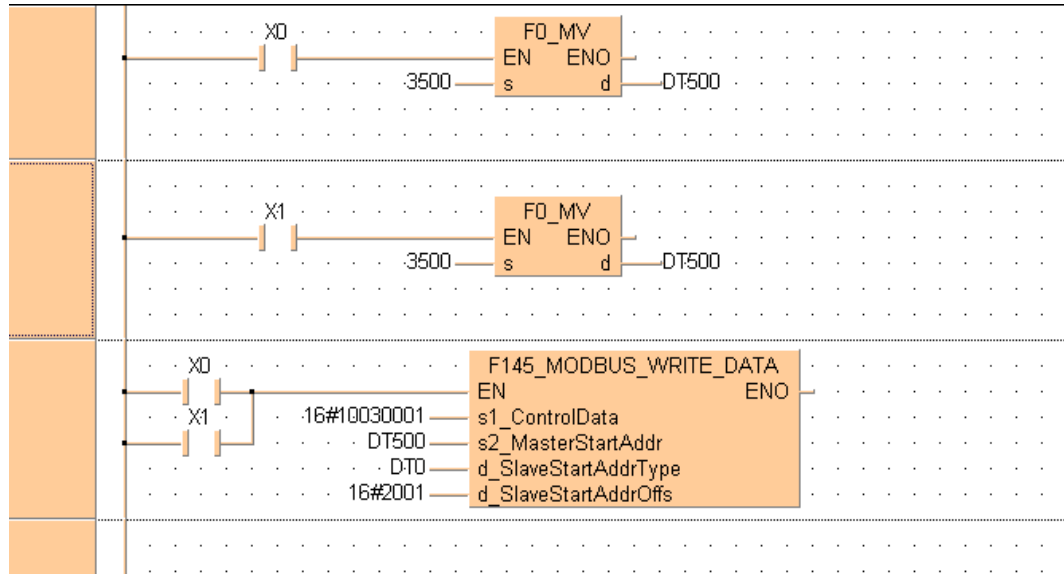
Create Project düğmesi ile projeyi başlatıyoruz. Ardından seri haberleşme portunun ayarlarını yapıyoruz.



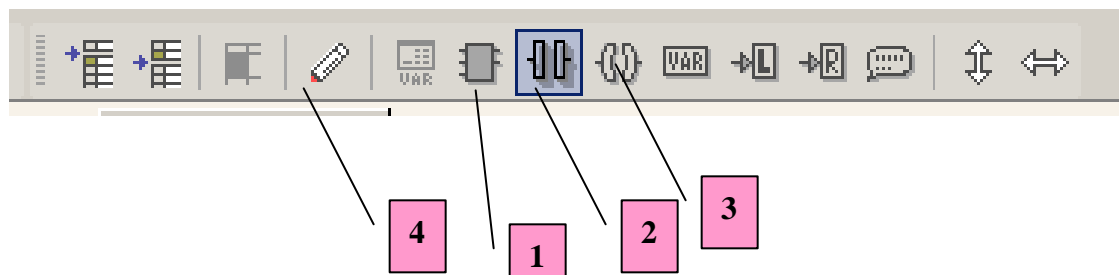
Ayarları Tamamladıktan sonra Ladder diyagramı ile PLC için gerekli programı yazalım.

PLC Ladder Programı :

Programımızda PLC üzerindeki X0 girişi aktif edildiğinde inverterdeki frekans parametresi 35 Herz olacak X1 aktif edildiğinde ise 40 Herz olacak şekilde Modbus üzerinden bu bilgileri PLC ile invertere göndereceğiz.



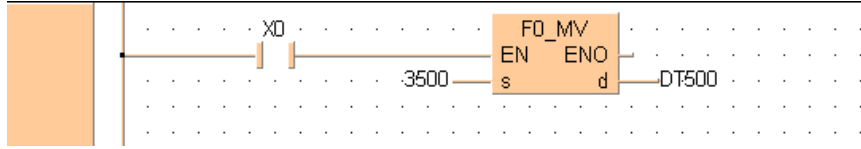
Şimdi satır satır Ladder diyagramını inceleyerek elemanları nasıl eklediğimizi anlayalım.



PLC ladder Programlama için yukarıdaki şekildeki araç kutusunu kullanacağız.

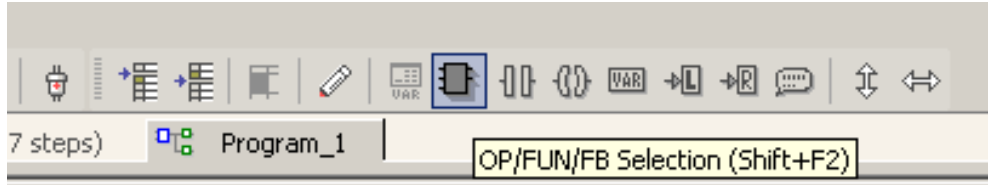
Araç Kutusundaki kullanacağımız araçlar ve işlevleri aşağıdaki gibidir :

1. Ladder içerisinde fonksiyon eklemek istediğimizde kullanılır.
2. Ladder içerisinde giriş eklemek için kullanılır.
3. Ladder içerisinde çıkış eklemek için kullanılır.
4. Ladder içerisinde bağlantı hattı eklemek için kullanılır.

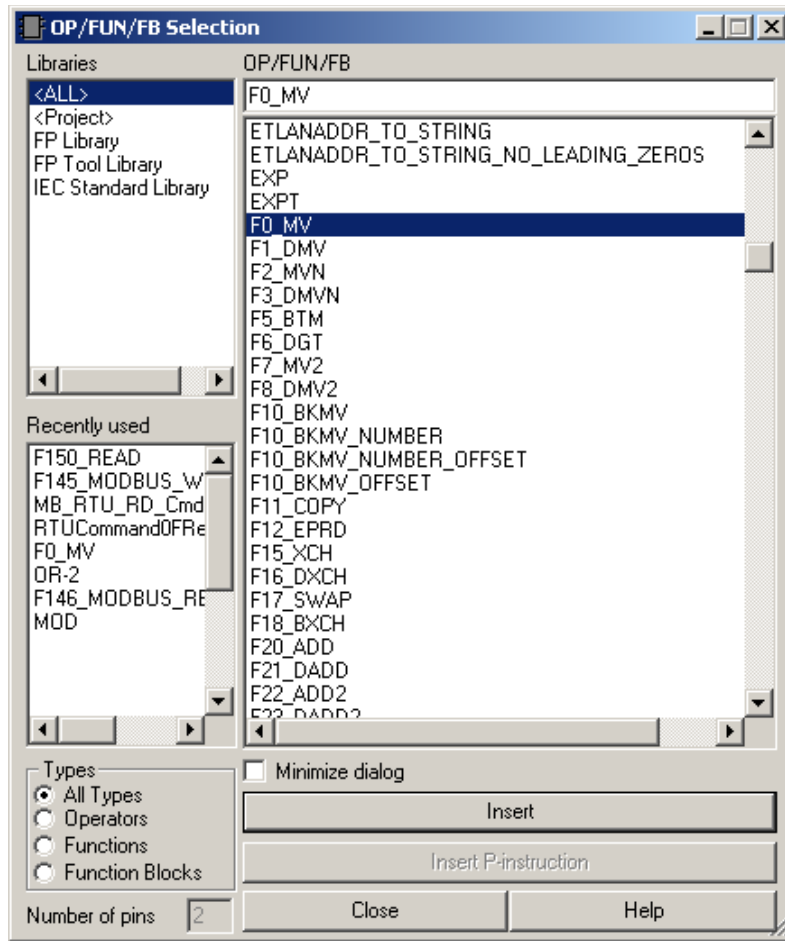


İlk satırımızda X0 girişini ve F0_MV fonksiyonunu ekliyoruz. Buradaki F0_MV fonksiyonu s ucuna verilen 1 word luk bilgiyi DT500 registerine aktarıyor. Buradaki 3500 Bilgisi inverterimiz için Herz bilgisi olacaktır.

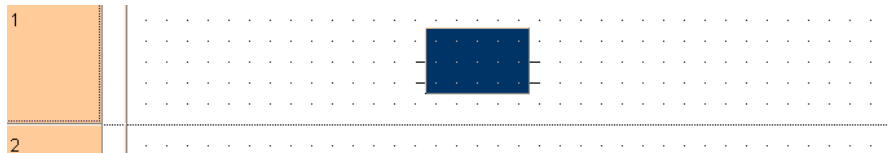
Şimdi adım adım F0_MV fonksiyonunu nasıl eklediğimizi görelim .



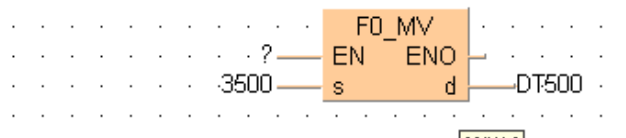
Araç kutusundan fonksiyon eklemeyi seçiyoruz.



İkinci adımda Karşıma çıkan fonksiyon seçim penceresinden F0_MV yi bulup inser ile ladderimize ekliyoruz.

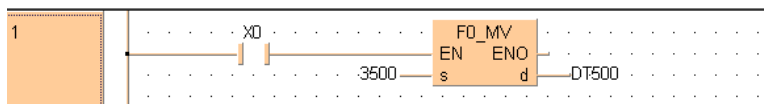


Mouse ile fonksiyonumuzu istenilen yere tıklayarak ekliyoruz. Daha sonra parametrelerini değiştiriyoruz.

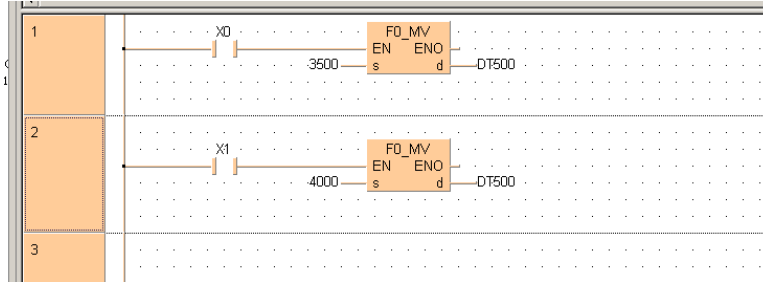


Klavye ile parametreleri belirtilen şekilde yapıyoruz.

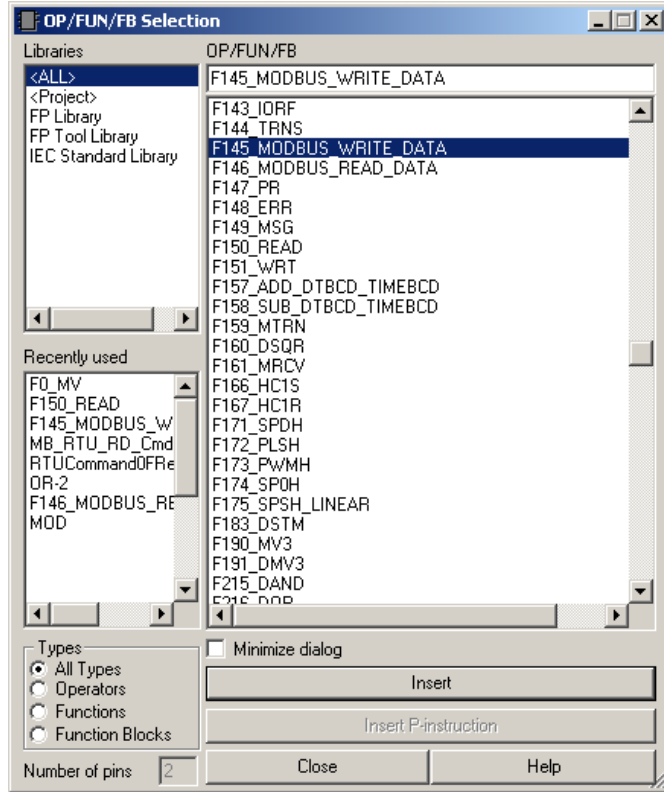
Daha sonraki adımda da X0 girişini ekliyoruz ve kalem ile bağlantıları yapıyoruz.



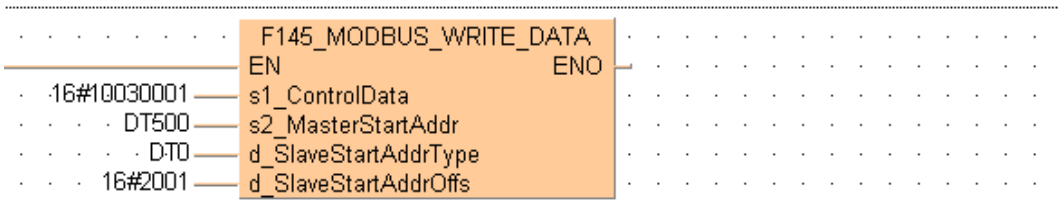
Benzer adımları tekrarlayarak diğer F0_MV fonksiyonlu satırı oluşturuyoruz.



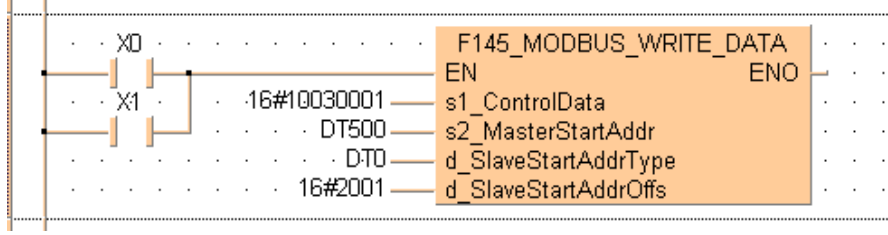
Son olarak da F145_MODBUS_WRITE_DATA Fonksiyonunu kullanacağız.



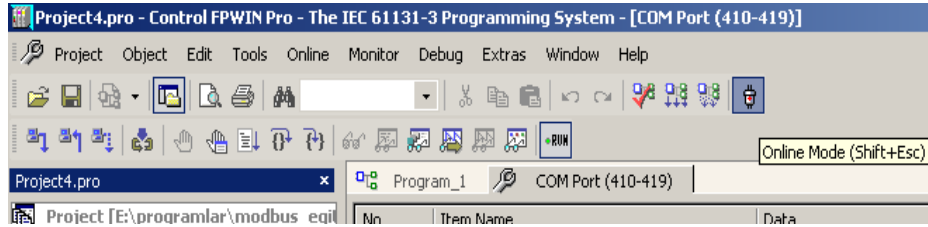
Fonksiyon seçme penceresinden F145_MODBUS_WRITE_DATA fonksiyonunu seçerek ekle düğmesine bastık. Fonksiyonu program arayüzünde seçilen yere ekleyerek parametrelerini aşağıdaki şekilde ayarlayalım.



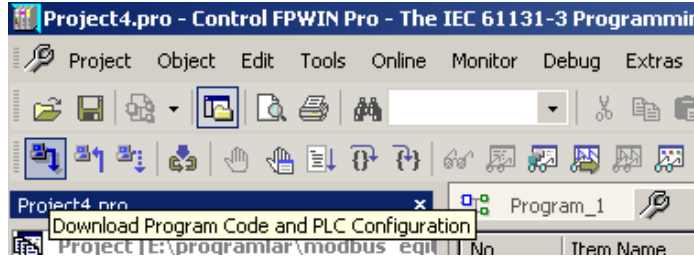
Fonksiyondaki parametrelerin açıklamalarını çalıştırma işleminden sonra anlatılacaktır. Fonksiyonu EN ucuna ise X0 ve X1 girişlerini şekildeki gibi bağlayalım.



Şimdi Programımızı PL ye yükleyelim. İlk olarak online moda geçelim.



İkinci basamakta ise programımızı PLC ye yüklüyoruz.



Programın Test Edilmesi :

Öncelikle PLC üzerindeki X0 a 24 Volt vererek aktif hale getiriyoruz. Inverter üzerindeki frekans göstergesi 35.0 oluyor.



Daha sonra X1 e 24 volt veriyoruz görüldüğü üzere inverter üzerindeki frekans bilgisi 40.0 oluyor.



DEVICENET

DeviceNet Nedir?

DeviceNet; tek bir ağ ile limit anahtarları, fotoelektrik sensörleri, vana manifoldları, motor starterleri, süreç sensörleri, panel göstergeleri, operatör arabirimleri gibi endüstriyel aygıtları kontrol etmek, bu aygıtların kurulum maliyetlerini ve kurulum için harcanan zamanı azaltmak için CAN (Controller Area Ağ (network)) mimarisi üzerine inşa edilerek geliştirilmiş açık, düşük maliyetli dijital, multidrop bir ağ (network) teknolojisidir.

DeviceNet, endüstriyel ortamlarda güvenilirlik ve performans gerekliliklerini karşılamak için tasarlanmıştır. Kararlı bir yapıya sahiptir ve kendini ispatlamış bir ağ (network) teknolojisidir.

DeviceNet'e Genel Bakış

DeviceNet kendi veri bağı katmanı için CAN (Controller Area Network) ve üst ağ (network) katmanları için CIP (Common Industrial Protocol) kullanır. DeviceNet ODVA tarafından yönetilen açık ağ (network) standardıdır ve dünya çapında uluslararası standartlar birliği tarafından kabul edilmiştir. DeviceNet diğer CIP tabanlı ağlar (network) (EtherNet/IP ve ControlNet) ile dikişsiz köprüleme ve yönlendirme özelliğine de sahiptir.

DeviceNet tüm dünyada birçok üretici tarafından desteklenmektedir. 700'ün üzerinde üretici kimliği ODVA (Open Devicenet Vendors Association) (Açık Devicenet Sağlayıcı Birliği) tarafından yayınlanmıştır. Gerçek şudur ki birçok üretici dünya üzerindeki en iyi sınıf ürünleri kullanıcılarına sunmak için DeviceNet'i kendi ürünlerinde uygulamak için seçmiştir ve bu üreticiler en uygun ürün desteği için coğrafi kapsam ve uygulama alanındaki uzmanlığı temel almışlardır.

ODVA'nın yetkili test servis sağlayıcılarının birinden uygunluk testinden geçen ve ODVA'dan resmi uygunluk bildirisi alan uygunluk testi yapılmış DeviceNet ürünleri açık ve çoklu üretici sistemine ve spesifikasyona uyulması için ODVA tarafından sertifikalandırılır.

DeviceNet endüstriyel kontrolörlerle I/O aygıtlarını birbirine bağlayan ve bu aygıtlar arasında iletişim ağı olarak hizmet veren dijital, multidrop bir ağıdır. Her aygıt ve kontrolör ağda bir düğümdür. DeviceNet çok yönlü iletişim hiyerarşisini ve mesaj sıralama önceliğini destekleyen üretici-tüketici ağıdır. DeviceNet sistemleri master-slave ya da eşler arası (peer to peer) iletişimi kullanan dağıtık kontrol mimarisi çalıştırabilecek şekilde düzenlenebilir. DeviceNet sistemleri I/O ve açık mesajlaşmayı destekleyerek ayarlama ve kontrol için tek noktalı bağlantı sunar. DeviceNet aynı zamanda ağdan güç elde etme gibi eşsiz bir özelliğe sahiptir. Bu, aygıtların sınırlı güç gereksinimlerini direk olarak ağdan almalarını sağlayarak bağlantı noktası ve fiziksel boyutun azalmasını mümkün kılar. Bunun sonucunda bağlantı sayısının artmasından kaynaklanan yüksek kablolama maliyetini ve kablolama hatalarını giderir. Farklı üreticilerin ürettiği bileşenlerin güvenilir bir şekilde birbiriyle değiştirilebilmesini sağlarken, aynı zamanda kablolama ve endüstriyel otomasyon aygıtlarının kurulumu için harcanan zamanı ve maliyeti azaltır. Farklı üreticilerin ürettiği bileşenlerin güvenilir bir şekilde birbiriyle değiştirilebilmesini sağlarken, direk bağlanabilirlik aygıtlar arasında geliştirilmiş iletişim sağlar. Çok önemli olan aygıt seviyesi hata teşhis ve tespitlerini, kolaylıkla erişilemez yerlerde ve katı kablolama yapılmış I/O arabirimler arasında bile mümkün kılar.

DeviceNet teknolojisi geniş ölçüde Amerika Birleşik Devletleri, Avustralya, Yeni Zelanda, Japonya ve Çin'de kullanılmaktadır. Avrupa'da da hızlı bir şekilde kullanımı yaygınlaşmaktadır.

DeviceNet otomotiv sektörü ağı için ISO standardı olan ve tarım, medikal, inşaa kontrolü, denizcilik sektörü, uzay sanayi gibi hemen hemen tüm endüstrilerde kullanılan Controller Area Network (CAN) olarak bilinen standardı temel almıştır.

DeviceNet hiyerarşik doğaya sahip ağ (network) iletişimleri (communications) için ISO standardı olan OSI (Open Systems Interconnection) modelini takip eder. Bu modeli takip eden çalışma ağları fiziksel uygulamalardan protokollere, yönetimden iletim kontrolüne ve çalışma ağları boyunca akan bilgi verilerine kadar gerekli tüm fonksiyonları tanımlarlar.

Günümüzde DeviceNet resmi GENELEC standardıdır (EN50325).

Ağ (network) Boyutu	64 Döğüme Kadar
Ağ (network) Uzunluğu	Seçilebilir Uçtan Uca Ağ (network) Uzaklığı Hız Değişimi 125 Kpbs: 500 m 250 Kpbs: 250 m 500 Kpbs: 100 m
Veri Paketleri	0-8 byte
Veri Yolu Topolojisi	Çizgisel (trunk hattı/drop hattı); güç ve sinyaller aynı kablo üzerinde taşınır.
Veri Yolu Adresleme	Multicast (tekten çokluya) ile peer to peer; Multimaster (çoklu efendi) ve Master/Slave (Efendi/Köle) özel durumlarda; Polled ve Change of State (durum değişimi) istisna durumlarda;
Sistem Özellikleri	Güç altında aygıt ekleyip çıkarma

Tablo 1 : DeviceNet'in Özellikleri ve Fonksiyonları

DeviceNet'in temel teknolojisi Allen-Bradley tarafından geliştirilmiş ve Mart 1994'te tanıtılmıştır. Daha sonra Nisan 1995'te teknoloji ODVA'ya (Open DeviceNet Vendor Association) aktarılmıştır.

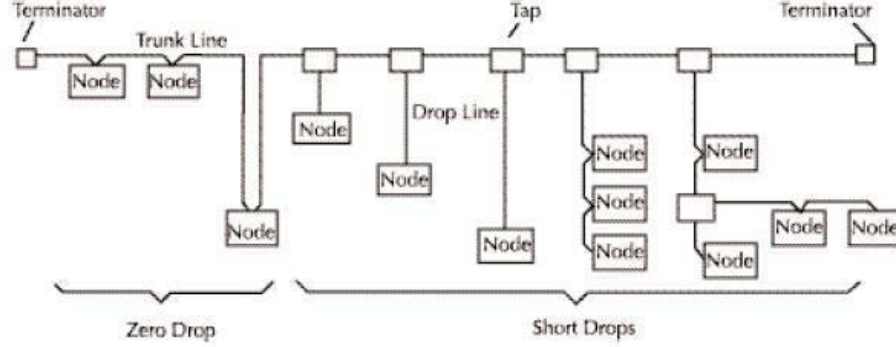
DeviceNet'in OSI Modeline Göre İncelenmesi

1. Fiziksel Katman

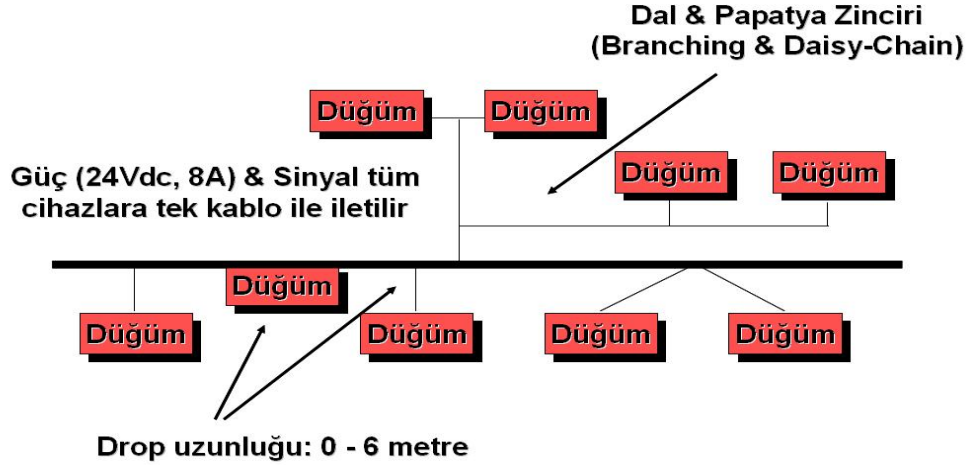
DeviceNet sinyal ve gücü taşımak için trunk-hattı/drop-hattı topolojisini ve bu topoloji içinde ayrı çift burgulu iletim yolunu kullanır. Bu topolojinin mümkün olan varyantları Şekil 1 ve Şekil 2'de gösterilmiştir. Trunk hattı veya drop hatlarının herhangi biri için kalın ya da ince kablo kullanılabilir. Ağlar için uçtan uca değişik uzunluklar ile kablo kalınlığı ve veri iletim hızı tablo 1'de gösterilmiştir.

DeviceNet drop hattı boyunca döğümler arası dal ve papatya zinciri modeli kullanır.

Trunk üzerindeki düğümler arası uzaklık maksimum 6 metre olmalıdır ve repeater (tekrarlayıcılar) ile bu uzunluk 4 km'ye çıkarılabilir. Her DeviceNet ağında en fazla 64 düğüm olabilir. Fakat efendi tarayıcı (master scanner) birinci düğüm numarası kullanır ve 63. düğüm ise varsayılan düğüm numarasıdır. Sonuç olarak diğer aygıtlar için 62 düğüm numarası mevcuttur.



Şekil 1: Kalın ve İnce Kablolar Trunk ve Drop Hatları İçin Kullanılabilirler

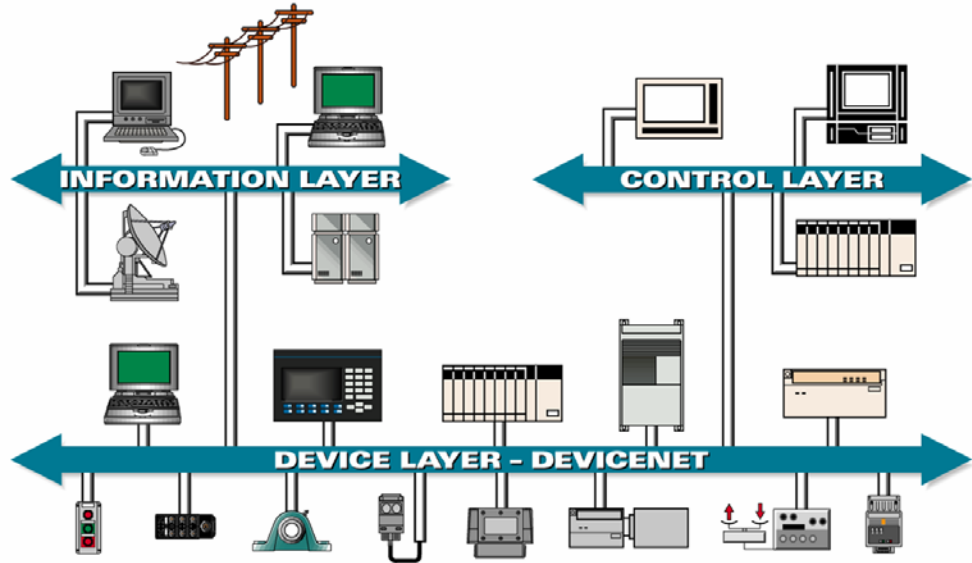


Şekil 2: Hem Sinyal Hem de Güç Tek Bir Kablo İle İletilir

VERİ ORANLARI	125 KBPS	250 KBPS	500 KBPS
Kalın Trunk Uzunluğu	500 m (1.640 ft)	250 m (820 ft)	100 m (328 ft)
İnce Trunk Uzunluğu	100 m (328 ft)	100 m (328 ft)	100 m (328 ft)
Yassı Trunk Kablo	380 m (1250 ft)	200 m (656 ft)	75 m (246 ft)
Maksimum Nokta Uzaklığı	6m (20 ft)	6m (20 ft)	6m (20 ft)
Kümülatif Nokta Uzaklığı	156 m (512 ft)	78 m (256 ft)	39 m (128 ft)

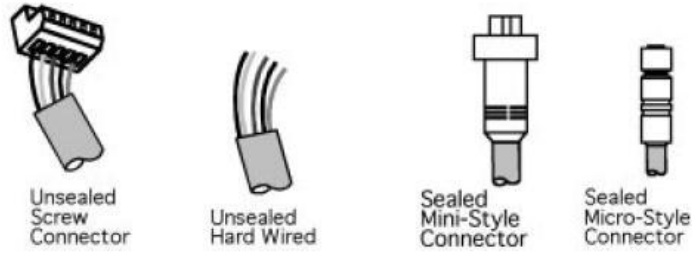
Tablo 2: Kablo Kalınlıkları ve Veri İletim Hızları

DeviceNet aygıt düzeyine yerleştirilmiş bir ağ (network) teknolojisidir (Şekil 3). DeviceNet kullanılan aygıtların izole edilmiş ve edilmemiş fiziksel katman tasarımlarının ikisini de destekler. Opto-izole tasarım seçeneği dışarıdan güç beslemeli aygıtların (AC sürücüler, starterler ve selenoid valfler) aynı veri yolu kablosunu paylaşmalarına izin verir. DeviceNet spesifikasyonu gerekli parçalar, kablo problemlerinden korunma ve örnekler ile ilgili ek bilgiler içermektedir.

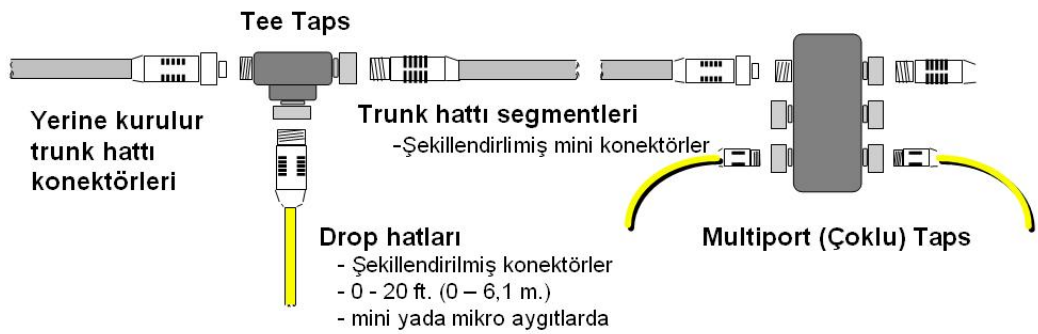


Şekil 3: DeviceNet Aygıt Düzeyine Yerleştirilmiştir

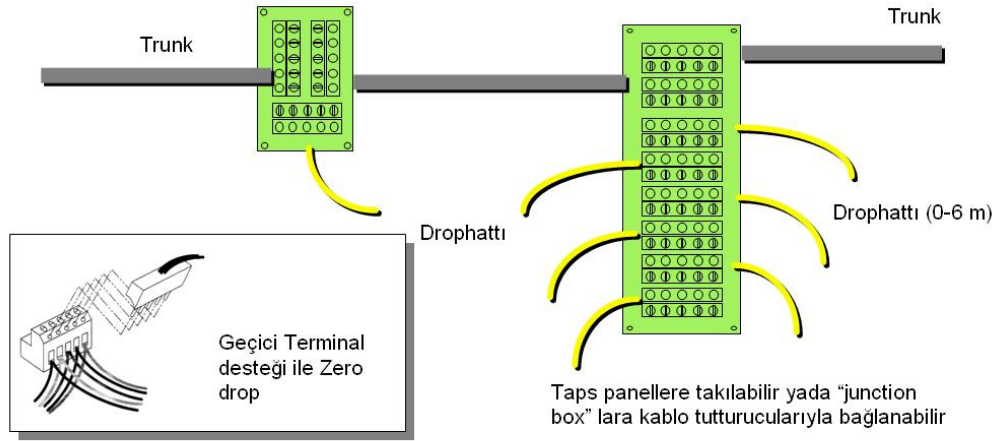
Farklı çeşit konektör tipleri DeviceNet üzerinde kullanılabilir. (Şekil 4) Mühürlü ve Mühürsüz konektörlerin ikisi de kullanılabilir. Büyük (mini) ve küçük (mikro) boy takılabilir mühürlü konektörlerde kullanılabilir. Mühürlü konektörler gerektirmeyen ürünler için açık tip konektörler kullanılabilir. Eğer takılıp çıkartılabilir bağlantı gerekli değil ise direk kabloya Vidalı ya da kısaçalı bağlantı uygulanabilir.



Şekil 4: DeviceNet Açık ve Mühürlü Tip Konektörleri Kullanılabilir



Şekil 5: DeviceNet Mühürlü Tip Bağlantı Konektörler



Şekil 6: DeviceNet Açık Tip Bağlantı Konektörleri

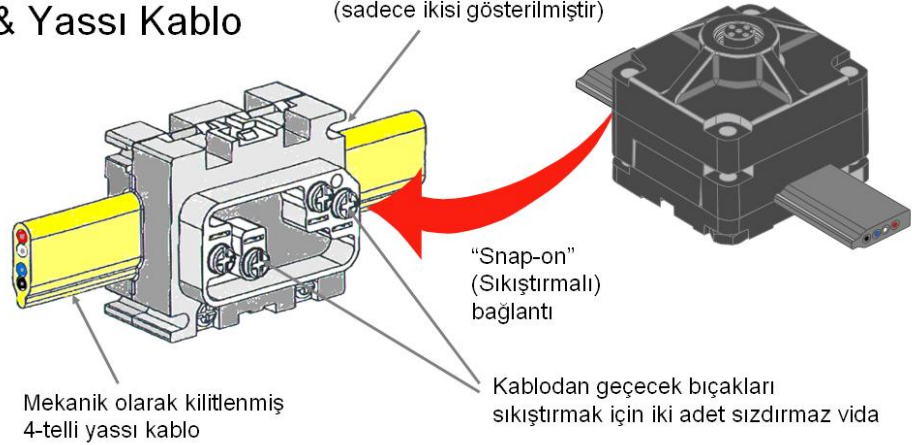
Düğümler güç kesilmeden ağı monte edilip çıkartılabilir. DeviceNet'in eşsiz özelliklerinden biride ağ (network) üzerinde herhangi bir noktaya güç ünitesi eklenebilme yeteneğidir. Bu özellik fazla güç kaynağı eklenmesini mümkün kılar. Trunk hattının akım büyüklüğü 8 amperdir.

IDC: Insulation Displacement Connectors

IDC Modül Tap & Yassı Kablo

Bütün montajı yapmak için dört adet yuva (sadece ikisi gösterilmiştir)

Üstten Görünüş



Şekil 7: İzolasyon Değiştirme Modüllü Yassı Medya

2. Veri Bağı Katmanı

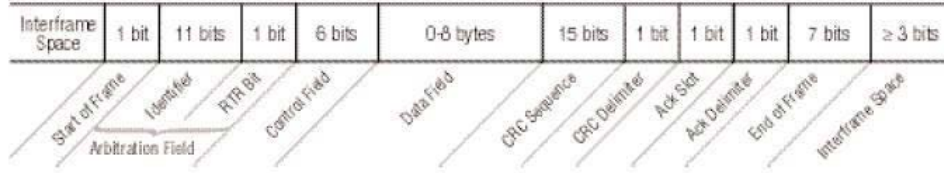
DeviceNet veri bağı katmanı CAN spesifikasyonları ve CAN kontrol cipleri kullanılarak tanımlanır. CAN spesifikasyonu dominant (lojik 0) ve resesif (lojik 1) olarak adlandırılan iki veri durumu ile tanımlanır. Herhangi bir verici veri yolunu dominant duruma geçirebilir. Veri yolu, hiç bir vericinin dominant durumda olmadığı zaman sadece resesif olabilir. Bu olgu CAN tarafından çalıştırılan veri yolu arabulucu planı ile çalışır duruma gelir.

CAN tarafından tanımlanan bazı çerçeve türleri:

- Veri çerçevesi

- Aşırı yük çerçevesi
- Uzak çerçeve (remote frame)
- Hata çerçevesi

Veriler DeviceNet üzerinde veri çerçevesini kullanarak hareket ettirilirler. Diğer çerçevelerin herhangi biri DeviceNet üzerinde kullanılmaz ya da istisna durumlar için kullanılırlar. DeviceNet veri çerçeve formatı aşağıdaki şekillerde gösterilmiştir.



Şekil 8: DeviceNet Veri Çerçevesi 1

Veri Çerçevesi



Tanımlayıcı ile 0 ile 8 byte arası veri yayını

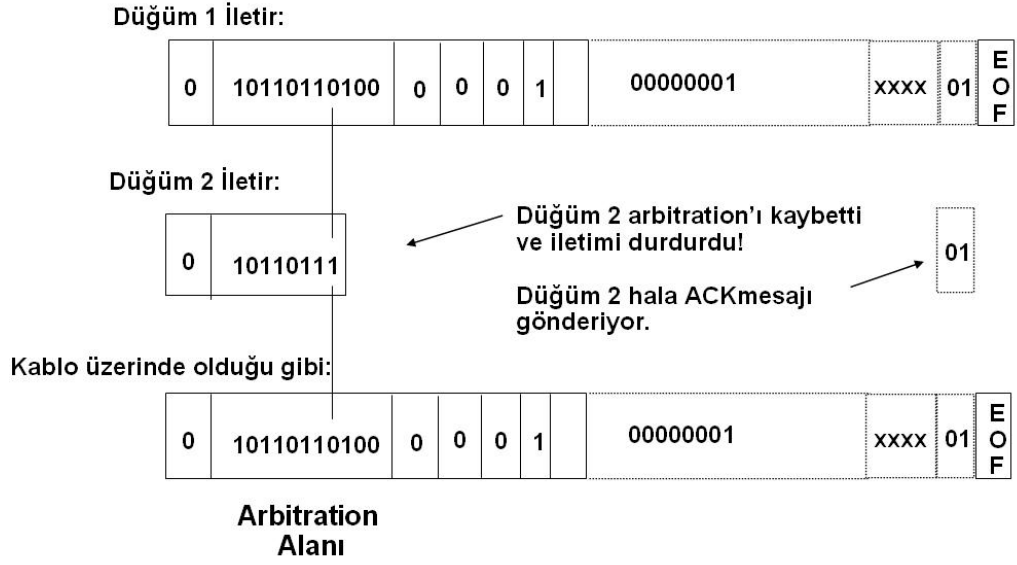
Şekil 9: DeviceNet Veri Çerçevesi 2

CAN taşıyıcı duyarlı bir ağ (network)tur. Ve düğüm diğer düğümler iletim yapmadığı zaman mesaj göndermeye çalışabilir. Bu özellik peer to peer bağlantılarına has olan bir yeteneği mümkün kılar. Eğer

İki ya da daha fazla CAN düğümü eş zamanlı olarak ağ (network)'a erişmeyi denerse zararlı olmayan (non-destructive) akıllı veri (bit-wise) arabulucu mekanizması potansiyel çakışma problemini veri ya da bant genişliğinde bir kayıp olmadan çözer.

Non-Destructive Arbitration (Zararlı Olmayan Çözüm)

- Ethernet'e çok benzer, her düğüm ağ (network) boş iken iletim için denemeler yapar.
- Ethernet'ten farklıdır, çarpışma (collision) yoktur. Kazanan düğüm kendi mesajı tamamlanana kadar iletime devam eder.
- Bu mekanizma bilgi ve zaman kaybının olmayacağını garanti eder.
- Tanımlayıcının değeri çözüm (arbitration) boyunca öncelik sırasını belirler.
- Sistemde hiçbir tanımlayıcı birbirine benzemez.



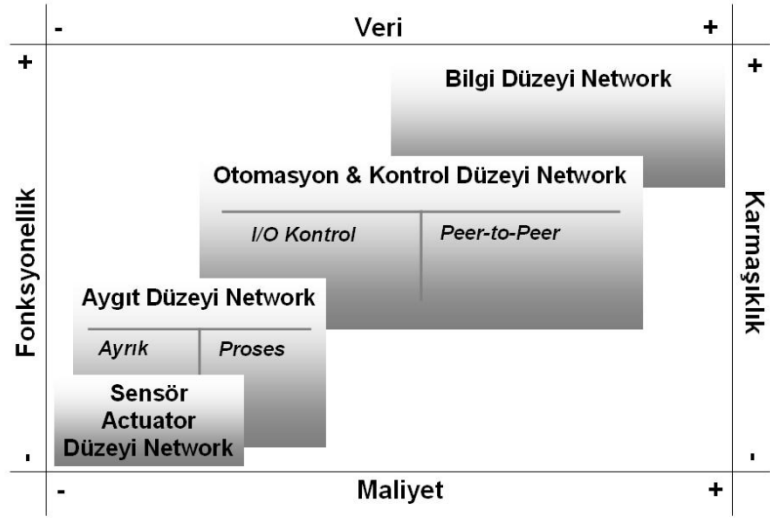
Şekil 10: Non-Destructive Arbitration (Zararlı Olmayan Çözüm)

CAN ağ (network) üzerindeki tüm alıcılar başlangıç çerçeve biti ile resesiften dominant'a kadar ifade edilmiş geçişleri senkronize ederler. Tanımlayıcı ve RTR (Remote Transmission Request -DeviceNet tarafından kullanılmayan) biti ile birlikte hakem (arabulucu) alanını oluştururlar. Hakem alanı ortam erişim sırasını kolaylaştırmak için kullanılır. Aygıt iletim yaptığı zaman, aynı zamanda her gönderdiği biti aynı olup olmadığında emin olmak için monitörler (tekrar geri alır). Bu sistem eş zamanlı iletimlerin tespit edilmesine izin verir. Eğer bir düğüm hakem alanını gönderirken resesif bit iletim dominant bit alırsa, iletim yapmayı durdurur. Eş zamanlı iletim yapan tüm düğümler arasındaki uzlaşmanın galibi en düşük numaralı 11-bit tanımlayıcı ile taktır. CAN aynı zamanda DeviceNet tarafından kullanılmayan 29-bit tanımlayıcı alan ile veri çerçeve formatını belirler.

Kontrol alanı 4-bit uzunluk alanı ve iki sabit bit içerir. Bu uzunluk veri çerçevesi içinde bayt sayısını gösteren 0'dan 8'e kadar bir sayı olabilir. 0–8 byte uzunluğu, sıkça değişen küçük miktarlarda I/O verisi ile çalışan düşük seviyeli aygıtlar (low-end) için idealdir. Sekiz byte basit aygıtlarda tanı (teşhis) verisi göndermek ya da hız referansı ve sürücünün hızlanma oranını göndermek için yeterlidir. DeviceNet aynı zamanda düğümlere en az protokol kullanımı ile daha fazla miktarda iletim yolu sağlayan parçalama protokolünü de tanımlar.

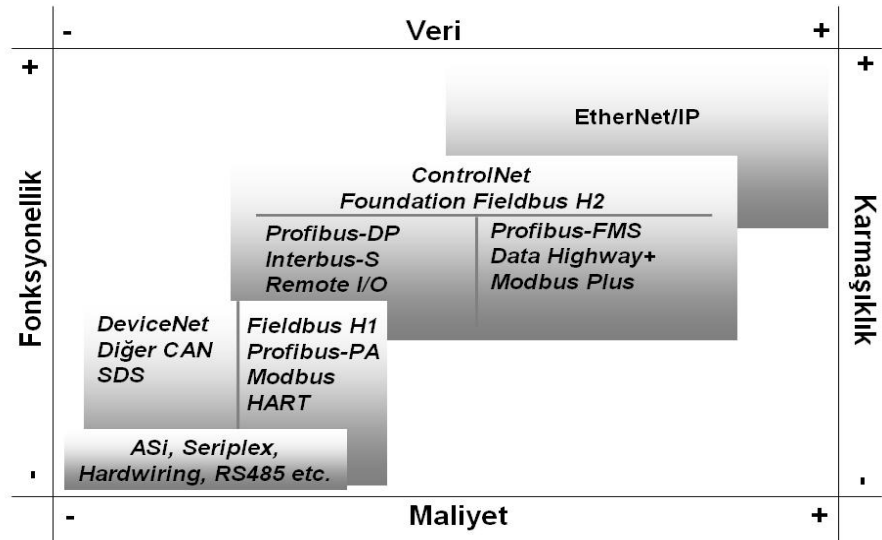
CRC (Cyclic Redundancy Check) döngüsel artıklık kontrolü anlamına gelir. CRC alanı CAN kontrolörleri tarafından çerçeve hatalarını saptamak için kullanılır. CRC kendinden önce gelen bitlerden hesaplanır. ACK slot içindeki dominant bit ileticiden başka en az bir alıcının iletimi duyduğu anlamına gelir. CAN çeşitli hata belirleme ve CRC' yi içeren, otomatik tekrar deneme yapan hata sınırlama metotları işleterek çok güçlü bir hata kontrolü sağlar. Bu genellikle uygulamaya açık metotlar hatalı düğümlerin ağ (network)u bozmalarını engeller.

3. Ağ ve Ulaşım Katmanları



Şekil 11: Ağ Katmanı Yerleşimi

DeviceNet aygıtlar ile bağlantının ilk olarak bilgi değişimi amacıyla kurulmuş olmasını gerektirir. Bağlantı kurmak için her DeviceNet ürünü herhangi bir Bağlanmamış Mesaj Yöneticisi (Unconnected Message Manager (UCMM)) ya da bağlanmamış grup 2 portu kullanacaktır. Bu iki durumda bazı kullanılabilir CAN tanımlayıcılarını rezerve ederek fonksiyonlarını gerçekleştirirler. Herhangi bir UCMM ya da bağlanmamış grup 2 portu açık mesajlaşma bağlantısı kurmak için kullanılır. Bu bağlantı daha sonra bilgileri bir düğümden diğerine taşımak için ya da ek I/O bağlantısı kurmak için kullanılır. Bir kez I/O bağlantısı kurulduktan sonra I/O verileri ağ (network) üzerindeki aygıtlar arasında taşınabilir. Bu noktada tüm DeviceNet protokolleri I/O mesajları 11-bit CAN tanımlayıcısı içerisinde bulunur. Geriye kalan her şey veridir.



Şekil 12: Ağ (network) Yerleşimi

11-bit CAN tanımlayıcısı bağlantı kimliği (ID) tanımlaması için kullanılır. Tek olmayan bağlantı kimlikleri (ID) çok sıkı bir şekilde, üretici – tüketici uyumluluğunda gerekli olan tam avantajı sağlamak kontrol edilir DeviceNet 11-bit CAN tanımlayıcısını dört farklı gruba böler. Tanımlana ilk üç grup iki alan içerir; ilk 6 bitlik alan MAC kimliği (ID)

ve diğeri de mesaj kimliği (ID) içindir. Birleşmiş (kombine) alanlar bağlantı kimliğini (ID) tanımlar. Dört mesaj offline iletişim için kullanılır.

Aygıtlar, istemci, sunucu ya da ikisi birden olabilirler. İstemci ve sunucular üretici, tüketici ya da ikisi birden olabilirler. Tipik istemci (aygıt) ve bağlantısı istek üretir, cevap alır. Tipik sunucu (aygıt) ve bağlantısı istek alır, cevap üretir. DeviceNet bu model üzerinde çeşitli varyasyonları sağlar.

Bazı bağlantılarda istemci veya sunuculardan herhangi biri sadece mesaj alabilir. Bu bağlantılar döngüsel ve durum değişimi mesajları için varış yeri olabilir. Benzer şekilde bazı bağlantılar da, istemci veya sunucudan herhangi biri sadece mesaj üretebilir. Bu bağlantılar döngüsel ve durum değişimi mesajları için kaynak olabilir.

Döngüsel ya da durum değişimi bağlantıları bant genişliği ihtiyacını büyük ölçüde azaltabilirler. DeviceNet sistemi içindeki düğümler kendi tanımlayıcılarını yönetmekten sorumlu olacak şekilde tasarlanırlar. Bu tanımlayıcılar tüm mesafe boyunca dağıtılırlar. Tüm düğümler ortam giriş kodu kimliğine (Media Access Code ID (MAC ID)) rağmen kendilerini geçerli kılan tam mesafe mesaj önceliğine sahiptirler. Kopyalanmış (duplicate) MAC ID algoritması sayesinde CAN tanımlayıcılarının merkezi bir aygıtı ihtiyaç duyulmadan ya da her ağ (network) kayıt edilmeden eşsizliği garantilenir. Şekil 13'te 11-bit CAN tanımlayıcısı içindeki DeviceNet paylarını göstermektedir.

IDENTIFIER BITS											HEX RANGE	IDENTITY USAGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						000 - 3ff	Message Group 1
1	0	MAC ID					Group 2 Message ID				400 - 5ff	Message Group 2
1	1	Group 3 Message ID			Source MAC ID						600 - 7bf	Message Group 3
1	1	1	1	1	Group 4 Message ID (0 - 2f)						7c0 - 7ef	Message Group 4
1	1	1	1	1	1	1	X	X	X	X	7f0 - 7ff	Invalid CAN Identifiers
10	9	8	7	6	5	4	3	2	1	0		

Şekil 13: DeviceNet İçindeki 11-bit CAN Tanımlayıcısı

Buradaki önemli konu kopyalanmış düğümlerin bulunmasıdır. Çünkü DeviceNet CAN tanımlama alanı içinde aygıt adresini kullanır. Bu sistem kopyalanmış adresli düğümleri bulmak için bir mekanizma sunar. Kopyalanmış adreslerin oluşmasını engellemek onların geçekleştikten sonra yerini bulmaktan daha iyidir (Diğer CAN tabanlı ağ (network)larda bazı şeyler hesap içine alınmaz.) Düğüm yönetimi konusunda diğer anahtar faydalardan biri de kullanıcının herhangi bir zamanda, sistemin kurulumu hakkında hiçbir bilgiye sahip olmadan düğüm ekleyip silebilmesi ve var olan düğümler arasında fazladan peer to peer mesajları ekleyebilmesidir. Merkezileştirilmiş kayıt sistemi takılmamalı ya da yapılmamalıdır. Düğümler hangi kimliklerin (ID'lerin) hala kullanımda olduğunu bilinceye kadar araç basitçe iki aygıt arasında I/O bağlantısı ekleme, öncelik düzeyini belirleme, veri yolu (sınıf, durum, özellik) ve üretim (tetikleme) başlatma için (döngüsel, poll (anket) durum değişimi) istekte bulunmalıdır.

4. Üst Katmanlar: Genel Endüstri Protokolü (Common Industrial Protocol (CIP))

DeviceNet daha üst katmanlar için CIP diye adlandırılan Genel Endüstri Protokolü'nü

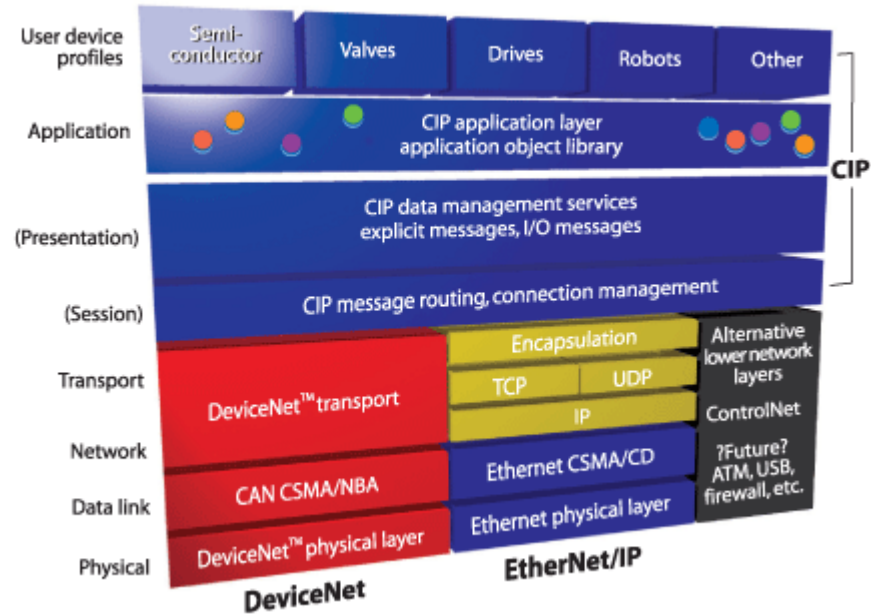
(Common Industrial Protocol) kullanır. CIP kesinlikle nesne yönelimlidir. Her nesne özelliklere (veri), servislere (emirler) ve davranışlara (olaylara tepki) sahiptir. İki farklı nesne CIP sipesisfikasyonu içine tanımlanmıştır: iletişim nesneleri ve uygulama özgül nesneleri. Aynı zamanda üretici üründe özgül nesneleri spesifikasyon değil fonksiyonellik gerektiren durumlar için tanımlayabilir (Şekil 14). Verilen ürün tipleri için minimum genel nesne takımı yeterli olacaktır. Aygıt tipleri ve üretici farklılıklarına rağmen aygıtlar arası işlerlik kullanıcı yararınadır. CIP tabanlı ağ (network)lar genel uygulama katmanını üzerine yapılandığından beri hangi ağ (network) aygıtı ev sahipliği yaparsa yapsın uygulama verileri aynı kalmaktadır. Uygulama programlarının sisteme hangi ağ (network) aygıtının bağlandığını bilmelerine gerek yoktur. Şekil 15'te CIP içerisinde OSI modeli ve DeviceNet tanımlamaları gösterilmiştir.

AC Sürücü Profili (Örnek)

Spesifikasyon içinde yayınlanmış	Üretici A	Üretici B	Üretici C
	start/stop	start/stop	start/stop
	fwd/rev	fwd/rev	fwd/rev
	accel/decel	accel/decel	accel/decel
	power calc.	power calc.	power calc.
Üretici Eklèmeleri	Üretici A	Üretici B	Üretici C
	other eng. units	foreign lang. temp. calc.	(none)

Şekil 14: Spesifikasyon Orneđi

The CIP object model



Here is CIP within the context of DeviceNet and the OSI model.

Şekil 15: CIP Nesne Modeli

CIP aynı zamanda farklı tip aygıtlar için nesne takımlarını minimum ayar (yapılandırma) seçeneklerine ve I/O veri formatlarını belirleyen aygıt profillerini de tanımlar. Standart profillerden birini takip eden aygıtlar aynı I/O verisine ve

yapılandırma seçeneklerine sahip olacaklar, aynı emirlerin hepsine cevap verecekler ve aynı profili izleyen diğer aygıtlar gibi aynı davranışlara sahip olacaklardır. Tablo 2’de gösterilen alttakım aygıt profilleri CIP ve DeviceNet tarafından desteklenmektedir.

Device Profile	Device Type No.
AC Drives	02 _{hex}
Communications Adapter	0C _{hex}
Contactor	15 _{hex}
DC Drives	13 _{hex}
DC Power Generator	1F _{hex}
General Purpose Discrete I/O	07 _{hex}
Generic Device	00 _{hex}
Human-Machine Interface	18 _{hex}
Inductive Proximity Switch	05 _{hex}
Limit Switch	04 _{hex}
Mass Flow Controller	1A _{hex}
Motor Overload	03 _{hex}
Motor Starter	16 _{hex}
Photoelectric Sensor	06 _{hex}
Pneumatic Valve(s)	1B _{hex}
Position Controller	10 _{hex}
Process Control Valve	1D _{hex}
Residual Gas Analyzer	1E _{hex}
Resolver	09 _{hex}

Tablo 3: DeviceNet Tarafından Desteklenen Aygıtlar

DeviceNet aynı zamanda minimum nesne gerekliliği için üreticilerin ürünlerine ek olarak kendi özelliklerini eklemeleri için standart mekanizmalara sahiptir. Bununla birlikte bu ek özellikler DeviceNet tanımlamalarında tavsiye edilen biçim ile kesin uyumlu bir şekilde uygulanmış olmalıdır.

Diğer önemli bir özellikte CIP tabanlı ağ (network)ların diğer ağ (network) mimarilerine ayarlanabilmesidir. Bu DeviceNet gibi CIP tabanlı ağ (network)lar üzerinde mesaj oluşturulması, daha sonra da bu mesajın, uygulama katmanında sunuş olmayan EtherNet/IP gibi CIP tabanlı diğer bir ağ (network)a geçme kabiliyetidir. Bu dikişsiz köprüleme ve yönlendirme kabiliyeti DeviceNet’i diğer fieldbus ağ (network)lar ile çalışmasını sağlar. Bunun anlamı spesifikasyonun içerdiği nesne takımları, köprüleme aygıtlarının bir ağ (network)tan diğerine mesaj içeriğinde oynama yapmadan mesajı iletmek için gerekli olan mekanizmayı tanımlar. Bu nesneleri destekleyen köprüleme aygıtı kullanıldığında zaman, kullanıcı sadece mesajın takip etmesi gereken yolu tanımlamakla sorumludur. CIP, yönetilen ağ (network)lardan bağımsız bir şekilde mesajın doğru olarak ele alınacağını garanti eder.

4.1. Üretici/Tüketici İletişim Modeli

CIP geleneksel kaynak/hedef modelinden ziyade üretici/tüketici tabanlı bir modeldir. Üretici-tüketici tabanlı ağlar bant genişliği kullanımında daha fazla etkinlik sağlarlar. Mesaj ağ (network) üzerinde üretilir, ulaşım adresi ile değil fakat bağlantı kimliği (ID) ile tanımlanır. Çoklu düğümler daha sonra başvurulmuş bağlantı kimliği ile veriyi alabilirler. Bu dinamik bağlantı yaklaşımının sonucu olarak ağ (network) etkinliğinde iki önemli fayda sağlanır.

Eğer düğüm veri almak isterse, herhangi bir zamanda üretilen veriyi almak için sadece

bir kez sorması gerekir.

Eğer ikinci (üçüncü, dördüncü vb.) düğüm aynı veriyi isterse, diğer düğümler ile aynı veriyi eşzamanlı almak için tüm ihtiyacı bağlantı kimliğini bilmektir.

4.1.1. Üretici/Tüketici Ve Kaynak/Hedef İletişim Modellerinin Karşılaştırılması

☐ Kaynak/Hedef iletişim modeli (Noktadan noktaya)

◇ Veriler her düğüme farklı zamanlarda ulaştığından senkronizasyonu sağlamak çok zordur.

◇ Sadece hedef farklı olduğu zaman bile mesaj birçok kez gönderileceğinden bant genişliği boşa harcanır.

src	dst	data	crc
-----	-----	------	-----

Şekil 16: Kaynak/Hedef İletişim Modeli

☐ Üretici/Tüketici iletişim modeli (veri tanımlıdır)

◇ Birçok düğüm tek bir üreticiden gelen aynı veriyi alabilir.

◇ Düğümler rahatlıkla senkronize olabilir.

◇ Bant genişliği daha verimli kullanılır.

identifler	data	crc
------------	------	-----

Şekil 17: Üretici/Tüketici İletişim Modeli

Kaynak/Hedef Modelinin Sınırlamaları

☐ Aynı mesajı birçok ağıta teslim etmek için birçok paket gerektirir.

☐ Veriler farklı hedeflere farklı zamanlarda varacaklardır.

☐ Ağ (network) bant genişliği üzerine baskı yapacaktır.

☐ Ağlarda Mesaj ve kritik zaman I/O' ları için farklı sonuçlar ortaya çıkacaktır.

Üretici/Tüketici Modelinin Yararları

☐ Programlamayı basitleştirir. (Kontrolsüz fonksiyonlarda veri yönetimi için kontrolöre gerek yoktur)

☐ Throughput'u artırır. (Ağ eski veriler ile doldurulmaz (örnek: sensör durumları))

☐ Zamana daha uygun cevaplar sağlar. (Veri sadece ihtiyaç olduğunda bir kez gönderilir (hata azaltımı))

□ Ek kablo gerektirmez. (Aynı ağ üzerinde kontrol, programlama ve tanı yapılır)

□ Üretkenliği artırır. (Tasarım, kurulum, yönetim, v.b. kolaylaştırır)

Üretici/Tüketici modeli polling (oylama), cyclic (döngüsel), change-of-state (cos) (durum değişimi), master/slave (efendi/köle), multimaster (çoklu efendi), peer-to-peer (eşler arası), one-to-one (bire bir), multicast (çoklu), broadcast (yayın) gibi birçok iletişim protokolü tarafından desteklenir.

4.2. Ön Tanımlı Master/Slave Bağlantı Seti

Temelde DeviceNet Peer to Peer mesajlaşma modelini kullanır. Ancak aynı zamanda DeviceNet master/slave ilişkisi üzerine tanımlı basitleştirilmiş iletişim planını destekler. Bu ön tanımlı bağlantı planı ön tanımlı master/slave iletişim seti olarak bilinir. Bu bağlantı metodu, kontrol uygulamalarında çok sık kullanılan I/O mesajlarının hareketlerini basitleştirir.

Birçok sensör ve aktuatör bazı ön tanımlı fonksiyonları gerçekleştirmek için tasarlanmışlardır. Bu fonksiyonların gerçekleşmesi için aygıtta güç verildiğinde aygıtın vereceği ve alacağı veri miktarı ve veri tipi bilinmelidir.

Ön tanımlı master/slave bağlantı seti aygıtta güç verildiği anda bağlantı nesnelerinin neredeyse tamamen ayarlanmış olmasını sağlar. Ağa güç verildikten sonra kalan tek gerekli adım master aygıtının (aygıtının) ön tanımlı bağlantı seti içinde slavelerden mülkiyet talep etmesi için veri akışına başlamasıdır. Slave aygıtlar bir ya da daha fazla takip eden mesaj tipini kullanarak veri üretebilirler.

Tip	İşlem Tanımı
Polled (oylama)	Master'ın tarama istesinde tanımlanan sıraya göre polled I/O mesajı için yapılandırılmış slaveler, master aygıttan "output (çıkış)" verisi alacaklardır. Masterın oylama (polling) hızı tarama listesindeki düğüm sayısı ile belirlenir. DeviceNet'in baud hızı, master tarafından üretilen mesajların boyutu, tarama listesindeki düğümlerin sayısı ve master aygıtının içsel zamanlayıcısı ile belirlenir. Verilen sistem konfigürasyonu için, bu mesajlaşma metodubelirleyicilik sağlayacaktır. Polled I/O "çıkış" verisi unicast vada multicast olabilir.
Cyclic (döngüsel)	Döngüsel I/O mesajı üretmek için yapılandırılmış aygıt verilerini kesin tanımlanmış aralıklarla üretecektir. Bu tip I/O mesajları uygulama için uygun olan oranda veri üretimi için kullanıcıya sistemi yapılandırma imkânı sağlar. Uygulamaya bağlı olarak bu sistem kablo üzerindeki trafik miktarını azaltabilir ve daha etkin bant genişliği kullanımını mümkün kılar.
Change of state (durum değişimi)	Durum değişimi mesajı üretmek için yapılandırılmış aygıt, her değişimde veri üretecektir ya da kalp atışı oranını temel alacaktır. Bu ayarlanabilir kalp atışı oranı tüketici aygıtlarının üreticinin hala hayatta ve aktif olduğunu bilmesini sağlayan bir yoldur. DeviceNet düğümlerin genişliğinde taşmaya yol açmalarını engellemek için durum değişimi mesajı üretim sıklığını sınırlandıran kullanıcı ayarlı üretim sınırlama zamanını da tanımlar. Kullanıcılar verilen uygulama senaryosu içinde bu parametreleri, optimum bant genişliği kullanımı

Tablo 4: Master/Slave Bağlantı Setinde Slave I/O Mesaj Tipleri

4. DeviceNet'in Yararları

- Kablolamayı azaltır ve kolaylaştırır.
- Kablolama ve kurulum maliyetini azaltır.
- Başlama zamanını (sistemi çalışır duruma getirmek için gereken zamanı) azaltır.
- Aksaklık süresini (sorunların çözülmesi için gereken süreyi) azaltır.
- Gücü kesmeden ağ (network) üzerinden aygıtların çıkarılması, eklenmesi yerine bağlanmasını mümkün kılar.
- Farklı üreticilerin ürettiği aygıtların birbiriyle değiştirilmesini olanaklı kılar.
- Süratli hata bulmayı sağlar.
- Hızlı ve verimli iletişim sağlar.
- Ağ (network) üzerinde aygıtların yapılandırılmasını kolaylaştırır. (hem kurulumda hem de çalışma zamanında konfigüre edilebilir.)

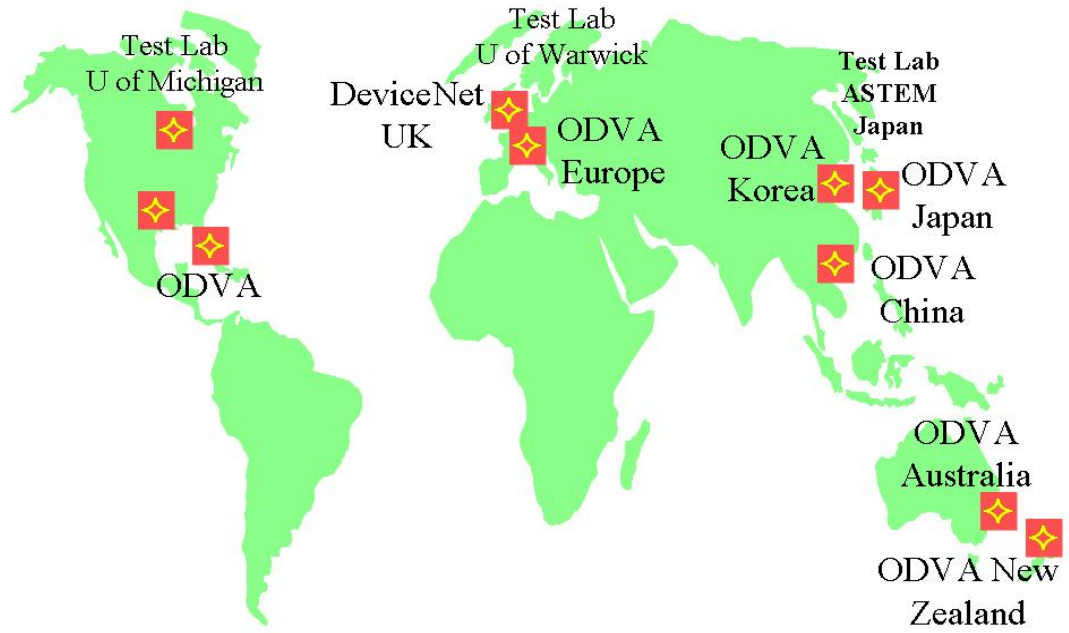
5. DeviceNet'in Uygulama Alanları

- Otomotiv sektörü (otomotiv transfer hatları, boyama ve işleme hatları)
- Yiyecek ve içecek sektörü (işleme ve paketleme)
- Malzeme işleme
- Paketleme
- Yüksek hızlı montaj
- Odun hamuru & kağıt
- Yarıiletken üretimi
- Hızlı cevap gerektiren uygulamalarda
- Güvenlik gerektiren yerlerde (otomatikler vs.)
- Sistemi uzaktan monitörlemek gerektiğinde

6. ODVA (Open DeviceNet Vendor Association)

ODVA DeviceNet spesifikasyonlarını yöneten ve DeviceNet'e dünya çapında destek veren bağımsız bir organizasyondur. ODVA üreticiler ile çalışır, geliştirici araçları, geliştirici eğitimleri uyum testleri ve satış aktiviteleri hakkında destek sağlar. ODVA DeviceNet ürün spesifikasyonlarını yayınlar ve spesifik (özel) ürün sınıfları için aygıt profillerini geliştiren Üretici Özel İlgi Grupları'nı (Special Interest Groups (SIG)) destekler. Herhangi bir üretici ODVA'ya üye olabilir ve Özel İlgi Grup'larına (Special Interest Group (SIG)) katılabilir. SIG'ler DeviceNet spesifikasyonlarının gelişmesine katkıda bulunurlar. Büyük ve küçük şirketler eşit söz hakkına sahiptirler. ODVA kanunlarla DeviceNet spesifikasyonları üzerinde bozulmayan bir koruma sağlar

ODVA Amerika tabanlı bir kuruluştur. Fakat Warwick Manufacturing Group, CAN Lab Avrupa ve ASTEM RI Kyoto (Advanced Software and Mechatronics Reserch Institute of Kyoto) Japonya ile üslerini genişletmiştir.



Şekil 18: ODVA'nın Dünya Üzerindeki Dağılımı

6.1. ODVA'nın Görevleri

- DeviceNet teknolojisini yönetir.
- Spesifikasyonları yayınlar.
- Test yazılımlarını yayınlar, test merkezlerini çalıştırır.
- DeviceNet teknoloji ve ürünlerinin geleceğini yönetmek için forum sağlar.
- Kullanıcılar ve sistem kurucuları için servisler ve DeviceNet araçları sağlar.
- Uluslar arası standardizasyonu takip eder.
- Üretici pazara sunum hızını (Time-to-Market) azaltır.
- DeviceNet'in tanıtımını yapar.
- Grup üyelerinin genel satış eforlarını yönetir.

6.2. ODVA'nın Hizmetleri

- Üretici ID (kimlik) kayıtlarını korur.
- Bağımsız test laboratuvarlarını işletir.
- Kurucular ve geliştiriciler için eğitim sağlar.
- Bilgi kaynağıdır. (Literatür oluşturur, telefonla uzmanlara yönlendirme yapar)
- Çoklu üretici uyum problemlerini çözer ve benzeri olmayan konularda karar verir.
- Web sitesi desteği: www.odva.org
- Dr. DeviceNet (ücretsiz teknik destek)

6.3. ODVA'nın Sağladığı Araçlar

ODVA sistem geliştiricilere veya kurulum teknisyenlerine DeviceNet için çeşitli araçlar sağlar. Bunlar DeviceNet ağ (network) monitörü ve DeviceNet asistanıdır.

6.3.1. DeviceNet Ağ (network) Monitörü

DeviceNet ağ (network) monitörü ağ üzerindeki kısa devreleri tespit eder, gerilim

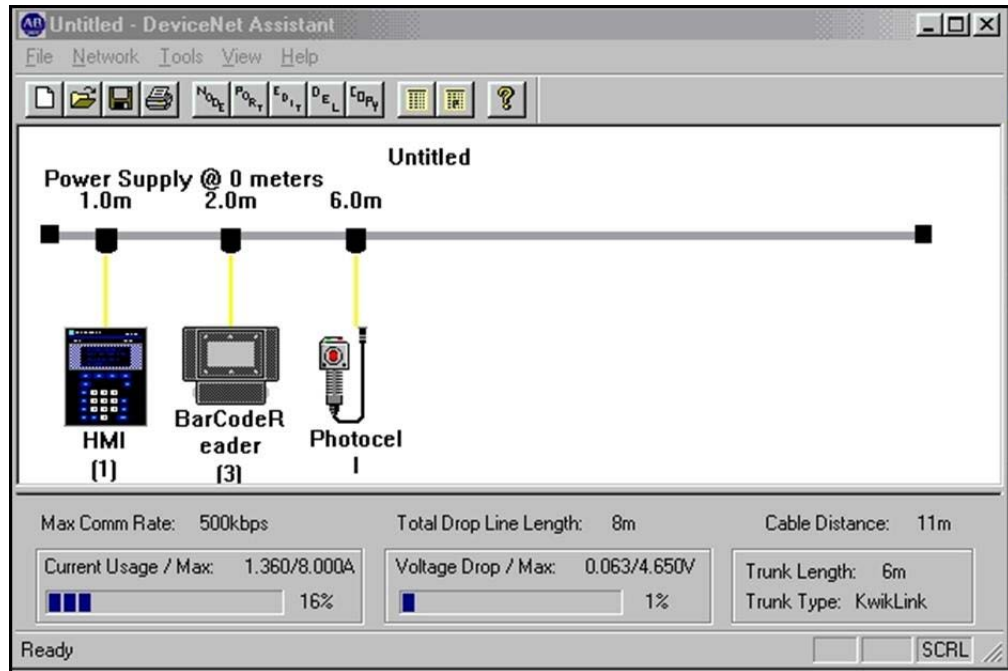
seviyelerini monitörleyerek yüksek ve alçak gerilim seviyelerini gösterir. Ayrıca otomatik olarak veri yolu hızını tespit ederek 125, 250 ve 500 KBPS hızlarını gösterir. Veri yolu trafiğini izleyerek trafik hakkında bilgiler verir. DeviceNet ağ (network) monitörü aynı zamanda hatalı çerçevelerin tespitinde de kullanılır.



Şekil 19: DeviceNet Ağ (network) Monitörü

6.3.2. DeviceNet Asistanı

DeviceNet asistanı sistem dizaynına yardımcı olmak için tasarlanmıştır. Bu araç güç ve uzaklık limit kontrollerini yaparak kablo bileşenlerini seçer. Gerekli olan tüm bileşenleri listesini otomatik olarak oluşturur.



Şekil 20: DeviceNet Asistanı

Bu yardımcı araçlar dışında Dr. DeviceNet, www.odva.org ve DeviceNet Üniversitesi ODVA'nın diğer hizmetleridir. ODVA ürün katalogları (Yazılı, CD-ROM, Web), spesifikasyonlar (Yazılı ve CD-ROM), haber klipleri, DeviceNet haber mektupları (gazeteleri) gibi yayınları da kullanıcılara sunmaktadır.

7. DeviceNet Kullanan Mevcut Ürünler

- Sensörler
- Vanalar
- Motor koruyucu/kontrolörleri
- AC/DC sürücüler
- I/O'lar
- PLC'ler
- PC arabirimleri
- S/W ürünleri
- HMI
- Geliştirici araçları
- Ve daha birçokları

8. DeviceNet Uygunluk (Conformance) Testi

ODVA uygunluk testi ürünlerin fiziksel katmandan uygulama katmanına kadar spesifikasyondaki ilgili konular bakımından uygunluğunu onaylamak için tasarlanmıştır. ODVA'nın yetkili test servis sağlayıcılarının birinden aşağıdaki tüm testlerden geçen aygıtlar ODVA'nın DeviceNet uygunluk (conformance) testinden geçmiş olur.

8.1. Control ve Bilgi Protokolü (Control and Information Protocol (CIP)) Testi

Kontrol ve bilgi protokolü testi ürünün CIP, mesaj ve DeviceNet için gereken servislerin gerekliliklerini karşıladıklarını doğrular. Uygulama katmanındaki aygıt profilinin uygunluğunu test eder.

8.2. DeviceNet Elektronik Veri Tablosu (Electronic Data Sheet (EDS) Testi

DeviceNet elektronik veri tablosu testi ürünün Electronic Data Sheet'inin (ASCII metin dosyası üretici tarafından sağlanır) doğru grameri ve aygıt kimliği, desteklenen mesaj tipleri, yapılandırma parametreleri, parametre listesi dâhil gereken tüm ürün bilgilerini içerdiğini doğrular. Bu dosya ağ (network) yapılandırma yazılım araçları tarafından ağ (network) üzerine aygıtların kurulumuna yardım etmek için kullanılır.

8.3. DeviceNet Fiziksel Katman Testi

DeviceNet fiziksel katman testi ürünlerin fiziksel katmanının tasarımının uygunluğunu ve düzgün bir şekilde işlediğini doğrular. Örnek olarak, bu test operasyon boyunca gerilim ve akım seviyelerinin izin verilen aralık içinde kaldığını, konektörlerin ve belirleyicilerin (indicators) ODVA spesifikasyonlarını karşıladığını doğrular.

8.4. DeviceNet Sistem Testi

DeviceNet sistem testi ürünün çoklu üreticili DeviceNet sistemi içinde çalıştığını onaylar. Ürün 64 düğümlü çoklu üreticili ortama urulur (takılır) ve özellikle potansiyel çoklu işlerliği ve sistem problemlerini tanımlamak için tasarlanmış çeşitli durumlarda çalıştırılır. Ürünler sisteme güç vermek için başlatılan farklı işlem zincirleri (power-up sequences)

süresince açık ve I/O bağlantılarını başarılı bir şekilde kurmalıdır ve ürünler diğer düğümler eklendiğinde ve çıkarıldığında çevrimiçi kalmalıdır. Test anı zamanda aygıtın farklı üreticilerden gelen tarayıcılar ile ağır ağ (network) trafiği altında çalışacağını ve Electronic Data Sheet dosyasının aygıtı yapılandırmak için kullanılabileceğini doğrular.

8.5. DeviceNet Güç Kaynağı Testi (Ek test)

DeviceNet güç kaynağı testi güç kaynaklarının gerekli olan gerilim ve akım değerlerini sağladıklarını ve çeşitli yük koşullarında spesifikasyon gerekliliklerini yerine getirdiklerini doğrular.

8.6. Yarıiletken Testi (Ek Test)

Yarıiletken testi yarıiletken üretim araçları üzerinde kullanılan DeviceNet ürünleri için farklı seviye bir testtir. Yarıiletken özel testlerine kabul edilecek ürünler ilk önce standart testlerden geçmelidirler. Bu ek seviye test, CE ve SEMI S2 gerekliliklerini sağlayan yarıiletken üretim araçları üzerinde kullanılan DeviceNet aygıtları için ODVA spesifikasyonuna ilave edilmiş arabirim rehberi ile konektörlerin, belirleyicilerin (indicators), anahtarların, izolasyonun, gücün ve nesne davranışlarının uygunluğunu doğrular.

9. DeviceNet Spesifikasyon Markası

Ürün bir kere uygun (conformance) uygunluk testlerinden geçtiğinde, ODVA üreticiye uygunluk markasını kullanmasına izin verir. Uygunluk testinden geçen ürünlerin üreticilerinin listesi aynı zamanda ODVA web sitesinde mevcuttur. Kullanıcılar sadece spesifik bir ürünün uygunluk markası alıp almadığını internet sitesinden bulabilirler veya aygıt üzerindeki ürün markasından anlayabilirler.



Şekil 21: DeviceNet Uygunluk Markası



Şekil 22: DeviceNet Uygunluk Markası Almış Ürün Etiketi

DeviceNet. ve CIP ODVA'nın tescilli markasıdır. DeviceNet CONFORMANCE TESTED ODVA'nın kayıtlı sertifikasyon markasıdır. EtherNET/IP ODVA tarafından lisans altında kullanılan tescilli markadır. ControlNet ControlNet International'ın tescilli markasıdır.

10. ODVA Üyeleri

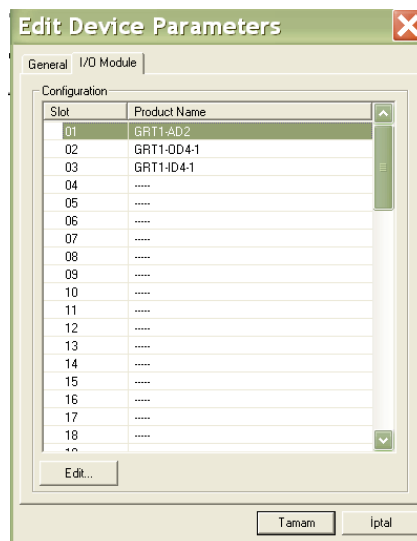
Dünya Üzerinde 700'ün üzerinde ODVA üyesi üretici vardır. Bunlardan bazıları: ABB, Allen-Bradley, ASCO/Joucomatic, Banner, Beldon, Cutler-Hammer, Festo, Fischer-Rosemount, Hitachi, Mitsubishi, Nematron, Omron, Parker Hannifin, Pepperl+Fuchs, Reliance, Schneider Electric, SMC, SST, Toshiba, Turck, Wonderware, Alvey, Jervis B.

Webb, KR Automation, Rapistan Demag, ...

11. DeviceNet Kullanan Önemli Firmalardan Bazıları

Anheuser-Busch, Applied Materials, Baxter Healthcare, Caterpillar, DuPont, General Motors, Giddings & Lewis, Heineken, Hershey Foods, John Deere, LaFarge Cement, Motorola, Nestle, Pepsi, Proctor & Gamble, Philip Morris, Quaker Oats, Ralston-Purina.

Master plc üzerinden devicenet ağı transfer edilemiyor. Slave plc üzerinden ağ transfer ediliyor. Çok önemli bir problem çözemedik
Smart slice ın içinde aşağıdaki alt üniteler var.Edit düğmesine basıldığında ilgili alt birimle ilgili parametre au,yarları yapılır.



Edit Device Parameters

General | I/O Module

Comment :

Network Power Voltage : V (11.0 - 25.0V)

Unit Conduction Time : Hours (0 - 429496729 Hours)

Last Maintenance Date :

Edit Unit Parameters - #01 GRT1-AD2

General | Analog Input 0 | Analog Input 1

I/O Comment :

Last Maintenance Date :

Function Choice

☐ Moving Average ☐ Peak/Bottom ☐ Comparator ☐ Rate of Change
☐ Scaling ☐ Top/Valley ☐ Cumulated Count

Range/Data Allocation

Parameter Name	Value
0000 Input Range	0 - 10V
0001 Analog Data1 Allocation	<input type="text" value="Raw Value"/>
0002 Analog Data2 Allocation	<input type="text" value="Raw Value"/>

Help
Select Analog Data that you would like to allocate to Value Attribute.

Network menüsünden transfer pc to network seçilir. Ağ transfer edilir.

Edit Device Parameters

Communication Cycle Time | Message Timer | Slave Function
General | I/O Allocation(OUT) | I/O Allocation(IN)

Unregister Device List

#	Product Name	Out Size	In Size

Register Device List

#	Produ...	Out Size	Out Ch	In Size	In Ch	C
#01	CJ1W...	2 Byte	3200:Bit00	2 Byte	3300:Bit00	
#02	GRT1...	1 Byte	3201:Bit00	7 Byte	3301:Bit00	

Advanced Setup... Register/Unregisterd

Upload Download Compare

Tamam İptal

Yukarıda nod0 master plc nin adres tablosu görülüyor. 3200 nod1 slave plc nin çıkış adresi, 3300 de giriş adresidir. 2 byte büyüklüğe sahiptir. Nod2 smartsliceninin out kanalı 3201, in kanalı da 3301 dir. BUNLAR MASTER PLC TARAFI ADRESLERİ. ŞİMDİ DE SLAVE LERDE BUNLARA KARŞILIK ADRESLER VAR. Aşağıda I/O allocation out sekmesi görünüyor.

Edit Device Parameters

Communication Cycle Time | Message Timer | Slave Function
General | I/O Allocation(OUT) | I/O Allocation(IN)

#	Product ...	Size	Ch	C
#01	CJ1W-D...	2 Byte	3200:Bit00	
#02	GRT1-D...	1 Byte	3201:Bit00	

Auto Delete Edit... Information

Memory Block 1

Ch	Product Na...
3200:Bit00	#01 CJ1W-...
3200:Bit08	#01 CJ1W-...
3201:Bit00	#02 GRT1-...
3201:Bit08	
3202:Bit00	
3202:Bit08	
3203:Bit00	
3203:Bit08	
3204:Bit00	
3204:Bit08	
3205:Bit00	

Setup...

Memory Block 2

Ch	Product Na...
3400:Bit00	
3400:Bit08	
3401:Bit00	
3401:Bit08	
3402:Bit00	
3402:Bit08	
3403:Bit00	
3403:Bit08	
3404:Bit00	
3404:Bit08	
3405:Bit00	

Setup...

Tamam İptal

I/O allocation in sekmesi aşağıda görülüyor. Burada da master plc adresleri görülebilir. Edit düğmesine tıklanarak bu adresler değiştirilebilir.

Edit Device Parameters

Communication Cycle Time

Message Timer

Slave Function

General

I/O Allocation(OUT)

I/O Allocation(IN)

#	Product ...	Size	Ch	C
#01	CJ1W-D...	2 Byte	3300:Bit00	
#02	GRT1-D...	7 Byte	3301:Bit00	

Auto

Delete

Edit...

Information

Memory Block 1

Ch	Product Na...
3300:Bit00	#01 CJ1W-...
3300:Bit08	#01 CJ1W-...
3301:Bit00	#02 GRT1-...
3301:Bit08	#02 GRT1-...
3302:Bit00	#02 GRT1-...
3302:Bit08	#02 GRT1-...
3303:Bit00	#02 GRT1-...
3303:Bit08	#02 GRT1-...
3304:Bit00	#02 GRT1-...
3304:Bit08	
3305:Bit00	

Setup...

Memory Block 2

Ch	Product Na...
3500:Bit00	
3500:Bit08	
3501:Bit00	
3501:Bit08	
3502:Bit00	
3502:Bit08	
3503:Bit00	
3503:Bit08	
3504:Bit00	
3504:Bit08	
3505:Bit00	

Setup...

Tamam

İptal

Edit I/O Allocate

Block : 1

Start Word : 3300

Allocated : 3300

☒ Low
 ☐ High

Occupied : 2

Byte

OK

Cancel

Aşağıda slave plc'nin (nod1) slave fonksiyonundaki adres tablosu görünüyor. 3370 out, 3270 de in olarak atanmıştır.

Edit Device Parameters

General | I/O Allocation(OUT) | I/O Allocation(IN)

Communication Cycle Time | Message Timer | Slave Function

☒ Auto Connection

OUT

Area : I/O Relay

Allocated : 3370

Occupied : 2 Byte

IN

Area : I/O Relay

Allocated : 3270

Occupied : 2 Byte

☐ User Setup

Poll | Bit-Strobe | COS | Cyclic

OUT

Area : I/O Relay

Allocated : 0

Occupied : 0 Byte

IN

Area : I/O Relay

Allocated : 0

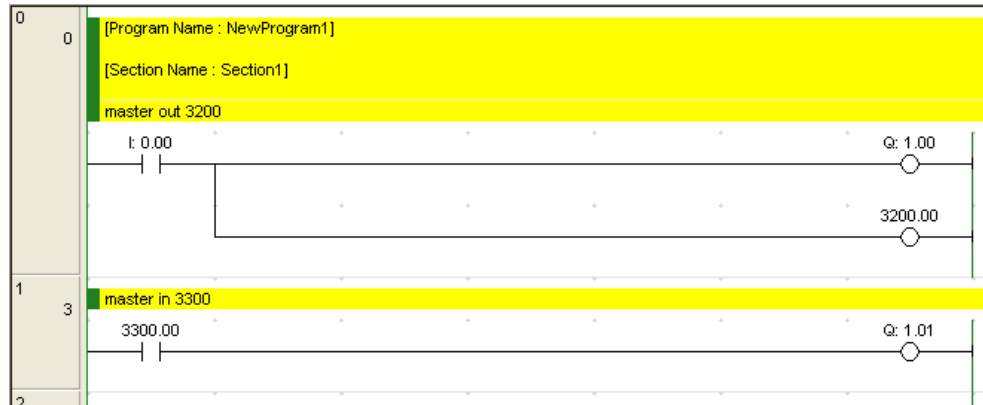
Occupied : 0 Byte

Tamam İptal

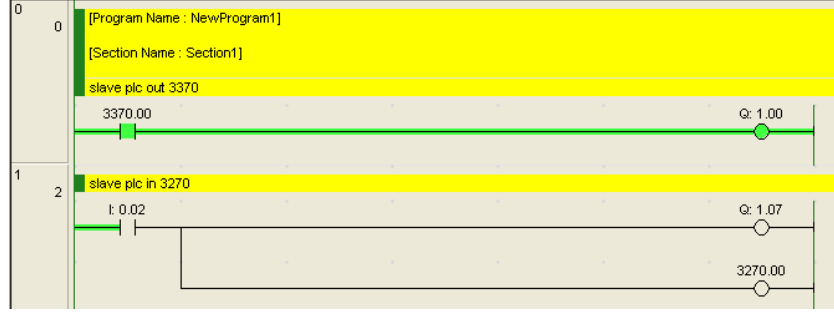
Outlar (Maste 3200-slave 3370)
In ler (Master 3300-slave 3270)

Master ve slave plc lerdeki outlar ve inler birebir haberleşirler. Örneğin 3200.0 adresi lojik 1 yapıldığında 3370.0 adresi de lojik-1 olur. 3300

Master plc'ye aşağıdaki programı yazdım.

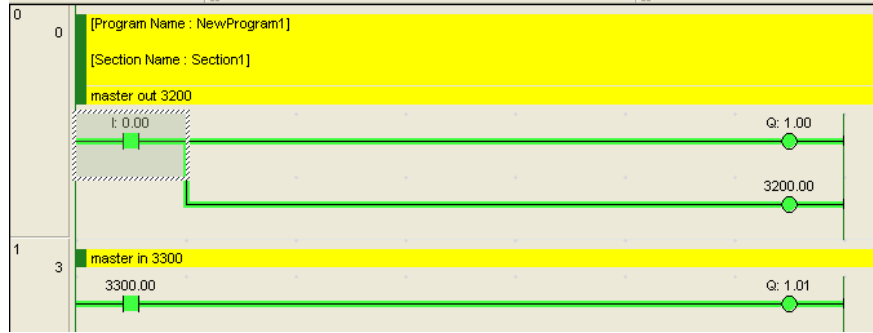


Slave plc'ye de aşağıdaki programı yazdım.



Master plc'de 0.0 girişı aktif edildiğinde yukarıda görüldüğü gibi slave plc de 3370.0 üzerinden 1.0 çıkışı aktif ediliyor.

Slave plc'de 0.2 girişı aktif edildiğinde master plc de 3300.0 üzerinden 1.1 çıkışı aktif olur.



SMART SLICE UYGULAMALARI

1) GRT1-AD2 analog giriş modülünün kullanılması

Analog giriş modülüne kanal0'a 0-10V giriş verildi. Smart slice nod numarası 2.

Edit Unit Parameters - #01 GRT1-AD2

General | Analog Input 0 | Analog Input 1

Comment :

Unit Conduction Time : Hours (0 - 429496729 Hours)

Default Connection Path (In) :

Default Connection Path (Out) :

Last Maintenance Date :

Available Channel :

Edit Unit Parameters - #01 GRT1-AD2

General | Analog Input 0 | Analog Input 1

I/O Comment :

Last Maintenance Date :

Function Choice

☐ Moving Average ☐ Peak/Bottom ☐ Comparator ☐ Rate of Change
☐ Scaling ☐ Top/Valley ☐ Cumulated Count

Range/Data Allocation

Parameter Name	Value
0000 Input Range	0 - 10V
0001 Analog Data1 Allocation	Raw Value
0002 Analog Data2 Allocation	Raw Value

Help
 NOTE! Input Range isn't enabled only by changing this parameter. RESET or re-start is required.

Yukarıda görüldüğü gibi giriş 2 kanal yapıldı. Analog kanal 0 0-10V a ayarlandı. Gerilim giriş yapıldı. Bağlantı terminaleri AD2 modülü üzerinde sol sütunda yukarıdan 2. ve 3. terminaler. IN0 + 2. terminal, IN0 – de 3. terminal.

Edit Device Parameters

General | I/O Module

Comment :

Network Power Voltage : V (11.0 - 25.0V)

Unit Conduction Time : Hours (0 - 429496729 Hours)

Last Maintenance Date :

Upload Download Compare Reset

Tamam İptal

Network gerilimi 24V yapılmalı.

Edit Device Parameters

Communication Cycle Time | Message Timer | Slave Function

General | I/O Allocation(OUT) | I/O Allocation(IN)

Unregister Device List

#	Product Name	Out Size	In Size

Register Device List

☒ Auto allocation as is registered.

#	Produ...	Out Size	Out Ch	In Size	In Ch	C
#01	CJTW...	2 Byte	3200:Bit...	2 Byte	3300:Bit...	
#02	GRT1...	1 Byte	3201:Bit...	7 Byte	3301:Bit...	


Advanced Setup... Register/Unregisterd

Upload Download Compare

Tamam İptal

3201 out kanalı, 3301 ise in kanalıdır. PLC memory CIO alanının aşağıdaki gibi değiştiriyoruz. Potu çeviriyoruz.

3302 adresinde 9,8V için 1706 bilgisi üretildi.



Start Address: 3300
On
Off
SetValue
ChangeOrder
ForceOn
ForceOff
ForceCancel

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0000	0004	1706	FFFF	0000	0000	0000	0000	0000	0000
CIO3310	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3320	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3330	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3340	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3350	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3360	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3370	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3380	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3390	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3400	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

J: On/Off, T: ChangeOrder
Ctrl+J: ForceOn, Ctrl+K: ForceOff, Ctrl+L: ForceCancel



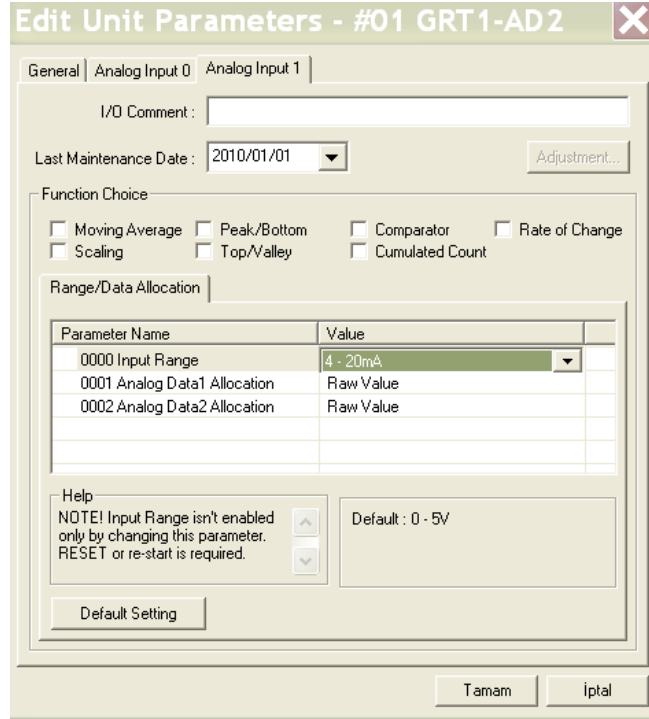
Start Address: 3300
On
Off
SetValue
ChangeOrder
ForceOn
ForceOff
ForceCancel

J: On/Off, T: ChangeOrder
Ctrl+J: ForceOn, Ctrl+K: ForceOff, Ctrl+L: ForceCancel

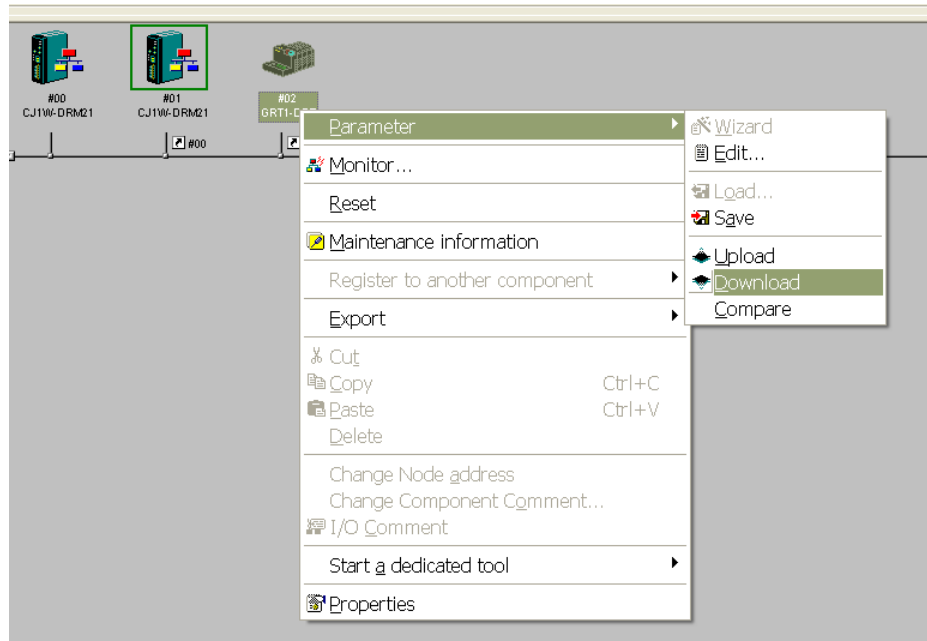
0V için yukarıdaki gibi 0000 değeri elde edildi.

2) GRT1-AD2 analog giriş modülünün kullanılması –Kanal-1'e analog akım çıkışlı indüktif sensör bağlanacak.

Kanal1'in parametreleri 4-20mA yapıldı.



CX İntegratörde nod2 nolu smarslice sağ tıklandı. Download ile yeni parametre ayarları bu cihaza yüklendi.



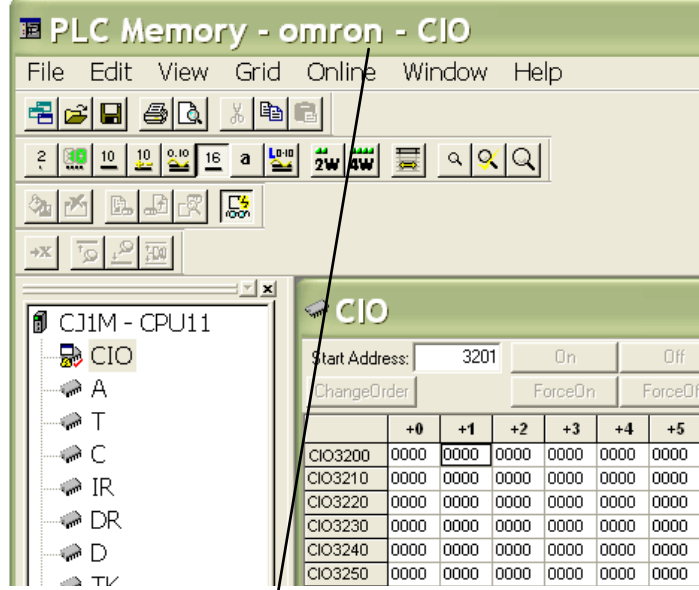
Sonra sensör analog1'e bağlandı. Kanal-1 3301 e bağlı. Zaten yukarıda 3301 adresine 0004 sayısı görülüyor.

3) OD4-1 Çıkış modülünün kullanılması

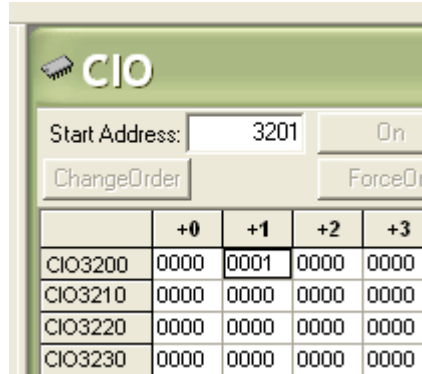
PNP transistör çıkışlarına sahip. 0, 1, 2, 3 diye çıkışları var. Bağlantılar aşağıda. Outputlar 3201'e bağlı.

0	1
NC	NC
G	G
2	3
NC	NC
G	G

HİÇBİR ÇIKIŞ YOK.



1 DEĞERİ GİRİLİNCE ÜNİTE ÜZERİNDE 0.ÇIKIŞ AKTİF OLUYOR. 24V değeri geliyor.



2 değeri girilince sadece 1. çıkış aktif, 3 değeri girilince 0. çıkış ve 1. çıkış aktif oluyor.

4 girilince 2. çıkış, 8 girilince 3. çıkış aktif oluyor. 24 v'a ulaşılıyor.

4) ID4-1 giriş modülünün kullanılması

PNP transistör girişlerine sahip. Giriş verilmesi için +24V uygulanması gerekiyor. 0, 1, 2, 3 diye girişleri var.

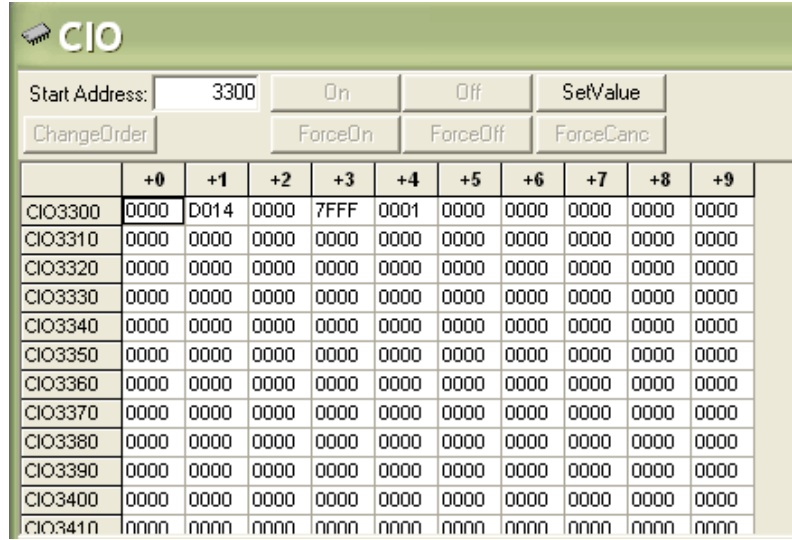
Bağlantılar aşağıda. .

0	1 Sinyal
V	V +24V
G	G 0V
2	3
V	V
G	G

3 kablolu sensöre 24V ve 0V yukarıdaki sırada bağlanır. Sensör sinyali(24V) ilgili girişe uygulanır. Doğrudan sinyal uygulanacaksa limit anahtar gibi Anahtarın bir ucu 24V'a diğer ucu da ilgili girişe bağlanır.

Örnek olarak girişlere doğrudan 24 V uygulayalım Memory alanını izleyelim.

0. girişe 24V verildi. 3304 nolu adreste 0001 görüldü.



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0000	0014	0000	7FFF	0001	0000	0000	0000	0000	0000
CIO3310	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3320	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3330	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3340	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3350	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3360	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3370	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3380	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3390	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3400	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
CIO3410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

1. girişe 24V verildi. 3304 nolu adreste 0002 görüldü.

2. girişe 24V verildi. 3304 nolu adreste 0004 görüldü.

3. girişe 24V verildi. 3304 nolu adreste 0008 görüldü.

Örneğin 0. ve 1. girişlere 24V verilseydi 0003 görülecekti.

PROFIBUS

Profibus(Process Field Bus) geniş kapsamlı üretim ve proses otomasyonu için tasarlanmış üreticiden bağımsız açık saha bus standartıdır. Profibus haberleşme sistemi Siemens'inde içinde bulunduğu bir çok PLC üretici firma tarafından geliştirilen ve standart olarak kabul edilen bir ağ sistemidir.Farklı amaçlar için geliştirilen PROFIBUS sistemleri olmasına rağmen biz sadece PROFIBUS DP (merkezi olmayan çevresel birimlerin) üzerinde duracağız.

PROFIBUS DP (dezentrale peripherie) otomasyon cihazı ile merkezi olmayan cihazlar arasında hızlı bir şekilde ver alış verişimi sağlayan bir haberleşme sistemidir. Özellikle PLC'nin merkezde, çevre birimlerinin (slave) çalışma sahasında (işin yapıldığı yerde) olduğu durularda iletim hatlarının oluşturulması çok kolay bir şekilde gerçekleştirilmektedir.

Merkezdeki CPU (master) giriş bilgilerini slave'lerden okur, bunları işler ve çıkış bilgilerini slave'lerin çıkışlarına yazar.

Profibus Teknik Özellikleri

Her bir bus bölümüne 32, toplam 126 katılımcı bağlanabilir.

- Çevre birimleri (slave'ler ve saha elemanları (sensör, motor) çalışma esnasında takılıp çıkarılabilir.
- Bu dağılımı "token-passing" sisteminin "master-slave" sisteminin yönetimine göre yapılır.

- Veri transferi iki damarlı blendajlı kablo veya optik iletkenler ile yapılır.
- Veri iletim mesafesi elektrik kabloları ile 12 km , optik kablolar ile 23.8 km kadar olabilir

Modüler deęiřtirme ve cihazların deęiřtirilebilmesi mmkndr.

- Her trl PROFIBUS-DP network sisteminde temel adımlar
- Btn slave modller bir master ile birleřtirilir.
- 2 master kullanılarak network geniřletilebilir.
- Adres aralıęı her slave modl iin 244 byte msaade eder.
- Master konfigrasyonu tamamlandıęında network yapısı da belirlenir.
- Profibus ayarlarını yapan kullanıcı, tm slave modller iin tek bir ID numarası kullanmalıdır. Aynı ID numarası verildięinde sinyal akıřması yařanacaęından PLC hata moduna geer.
- Profibus sistemler PLC' ye direk olarak bir kablo vasıtası ile baęlantı saęırlar.



Remote I/O (Profibuslı Valf Adası)

PROFIBUS, kullanım alanına ve tipine gre , haberleřme protokolleri (DP-PA-FMS) sunar .

Profibus DP (DP-Decentralized Periphery/Daęıtılmıř evre anlamına gelmektedir) , en ok kullanılan haberleřme tipidir. Hız, etkinlik ve dřk baęlantı maliyeti zellikleri sebebiyle tercih edilir. Merkezi daęıtılmıř cihazlarla haberlesme saęlandığı gibi (Siemens ET200) akıllı saha cihazları ile de haberlesme saęlanabilir(Motor suruculeri, debimetreler vs...)

Profibus-DP haberleřme fiziksel yapı olarak RS485 ve/veya fiber optik alt yapısını kullanır.

FMS (FMS-Fieldbus MessageSpecification Saha veriyolu mesaj tanımlaması anlamına gelmektedir.) genel bir haberleřme tipidir. Akıllı kontrol aygıtlar arasındaki haberleřme iin geliřmiř uygulama fonksiyonlarını sunar. TCP/IP' nin gelişiminin ve kullanımının hızla artması sonucu olarak, FMS gelecekte daha az rol alacaktır .

FMS' de DP gibi fiziksel haberleřme yapısı olarak RS485 ve/veya fiber optik alt yapısını kullanır.

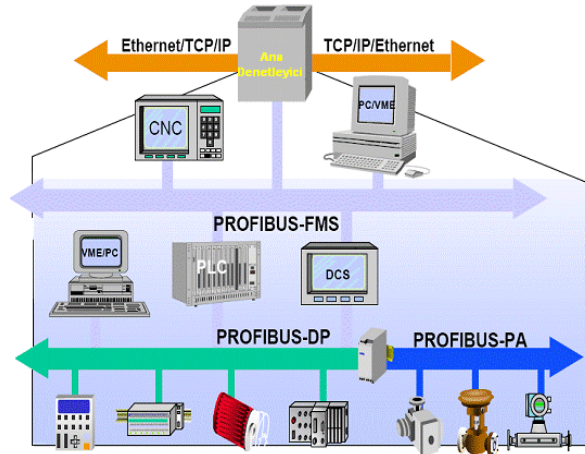
Profibus-PA (PA-Process Automation), daha ok proses otomasyonunda kullanılan ve mevcutta kullanılan haberlesme kablosu ile ekipmanın enerji beslemesi de saęlanan haberleřme tipidir. Profibus DP ile aynı master zerinde alıřabilir.

Bilindięi gibi ISO/OSI Modelinde 7 katman vardır:

- Katman 7 Uygulama Katmanı
- Katman 6 Sunum Katmanı
- Katman 5 Oturum Katmanı
- Katman 4 Tařıma Katmanı
- Katman 3 Aę Katmanı
- Katman 2 Veri Baęı Katmanı

- Katman 1 Fiziksel Katman

Profibus, OSI referans modeline yönelik bir protokol mimarisidir. Etkili bir haberleşme protokolü olan DP ,kullanıcı arabirimi olarak 1. ve 2. katmanları kullanır 3 .ve 7 .Katman kullanılmaz. Bu mimari, etkili ve hızlı iletim sağlar. Genel haberleşme protokolü olan FMS' te 1. 2. ve 7. katmanlara farklı önemler atanmıştır.Uygulama katmanı olan 7 .katmanda, Saha Veriyolu Mesaj Tanımlaması (FieldbusMessage Specification-FMS) ve Düşük Katman Arayüzü (Lower Layer Interface-LLI) vardır. FMS, ana birim-ana birim ve ana birim-uydu haberleşmesi için gerekenetkili haberleşme hizmetlerini tanımlar. LLI 2. Katmanın veri iletimprotokolünde FMS hizmetlerini tanımlar.



Profibuslı Fabrika Yapısı

ISO/OSI modelinde, katmanlardan 1. ve 2. Katman, DP ve FMS versiyonları için aynıdır. Diğer bir deyişle, hem fiziksel iletim ekipmanı (Örneğin RS-485 tanımlamalarına uygun hatlar, fiber optik iletkenler, fiş bağlantıları, ve parça bağlantıları için tekrarlayıcılar), hem de paket (telegram) veri iletim formatı benzerdir Katman 3 ve 6 Profibus ' ta kullanılmaz. FMS versiyonu, uygulama katmanında tanımlıdır. Uygulamanın verimliliği açısından, Profibus-DP versiyonunun 7. katmanı boş bırakılmıştır. Bu nedenle, DP versiyonu, katman 2'ye ait standart bir uygulama olarak kabul edilebilir.

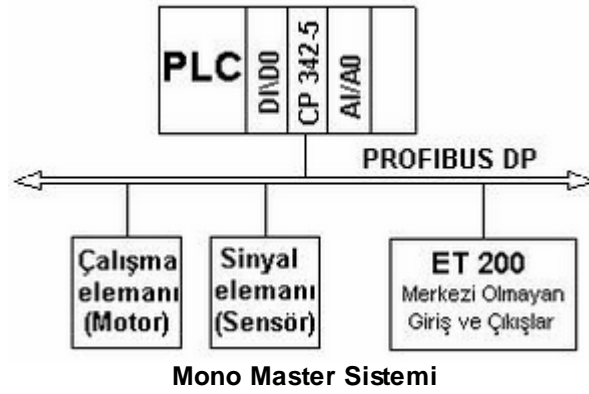
Profibus bus yapısında, bir Master için maksimum 32 adet slave cihaz takılabilir. Maksimum 4 adet repeater takılmak suretiyle bu adet 126'e kadar çıkartılabilir.

Profibus-DP ile haberleşme hızı 9.6 kD ile 12 MD arasındaadır. Profibus-PA da is haberleşme hızı 31.25 kD'dir.

PROFIBUS DP iki şekilde oluşturulabilir:

- Mono master
- Multi master

Mono Master Sistemi

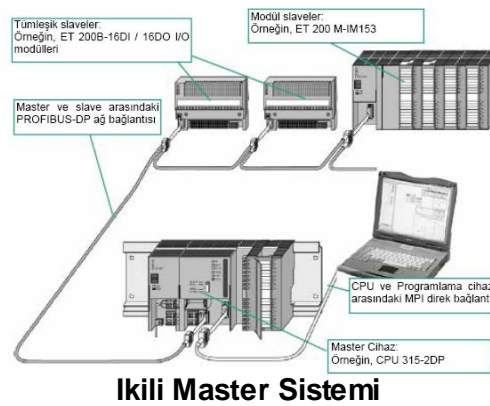


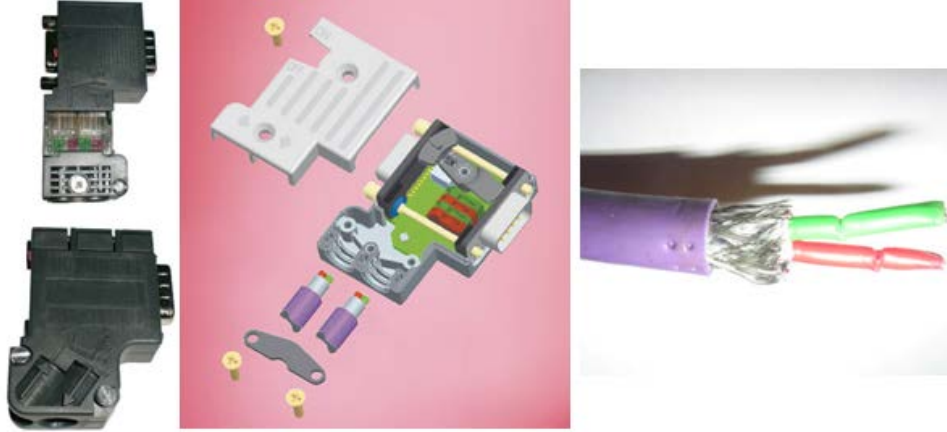
Tek merkezli kumanda şeklidir. Merkezi kumanda birimi olarak PLC kullanılır ve çevresel birimler (slave'ler PLC'e bağlanırlar. Program belirlenen çevrim dahilinde slave'lerden bilgileri alır ve onları değerlendirir.

CAN-OPEN İLETİŞİMİ

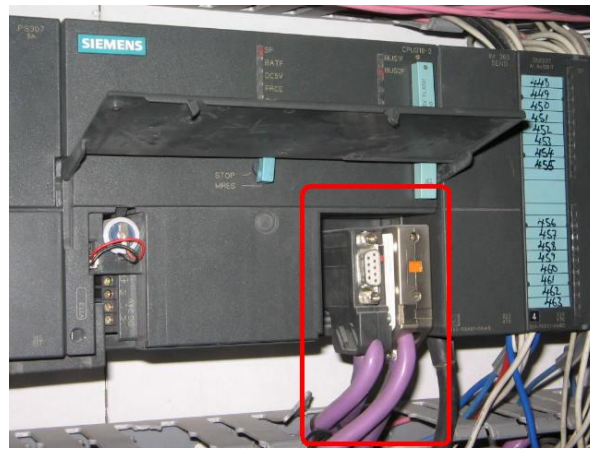
Bu sistemde birden fazla master bulunur. Bu masterlar birbirinden bağımsız olarak, her biri bir master ve ona ait slavelerden meydana gelen alt sistemleri oluştururlar. Ana sisteme ait farklı görevleri yerine getirirler. İlave görselleştirme, arıza takip düzeneği gibi.

Slavelere ait giriş çıkış görüntüleri bütün masterlerden okunabilir. Çıkışlara bir şey yazılması ise sadece ilişkilendirilmiş master tarafından gerçekleştirilebilir. Masterler birbirileri ile veri alışverişi yapabilirler. Multi master sisteminde çevrim süresi oldukça uzundur. Bu sistemler “Token Passing” (bayrak yarışı) sistemine göre çalışırlar, yani bayrağa sahip olan gönderme hakkına sahip olur. Bu hak master den mastere belli zaman aralıklarında devredilir





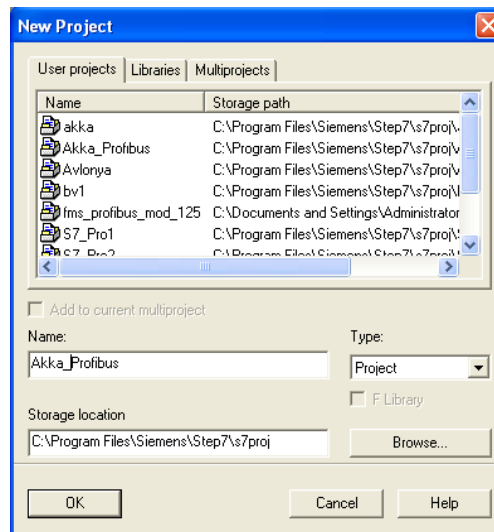
Profibus Kablosu ve Soketi



Profibus Bağlantısı Yapılmış PLC

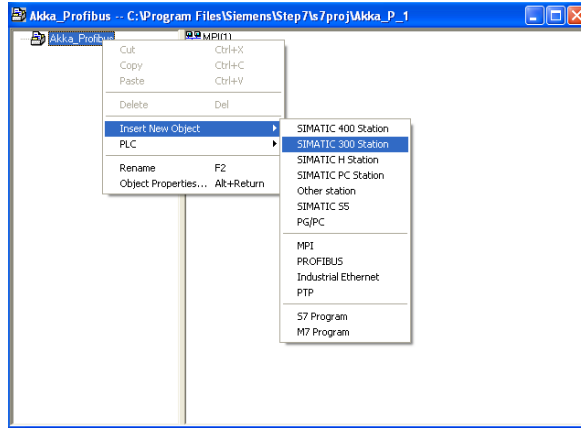
SIEMENS S7313C-2DP'nin PROFIBUS BAĞLANTISI

1. Yeni bir proje oluşturunuz.

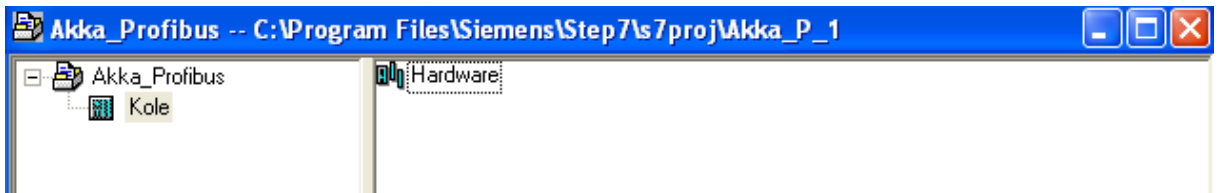


Donanımı yapılandırılın.

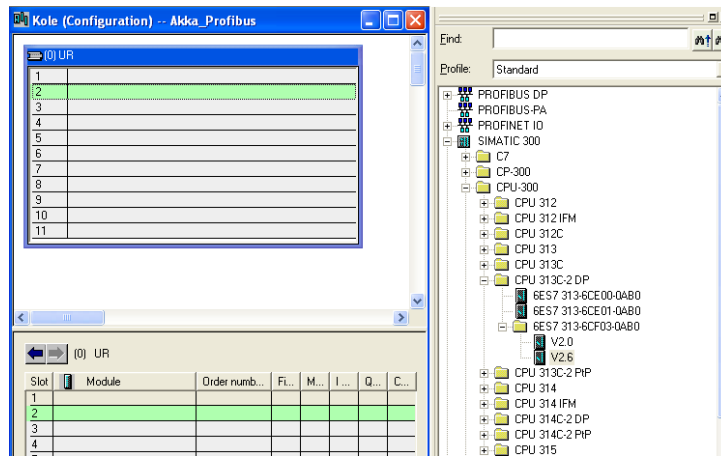
2. **SIMATIC 300 Station** klasörünü seçiniz.



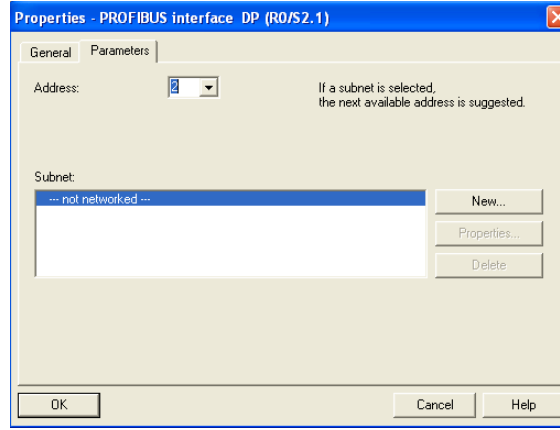
3. Kole ismini verelim.



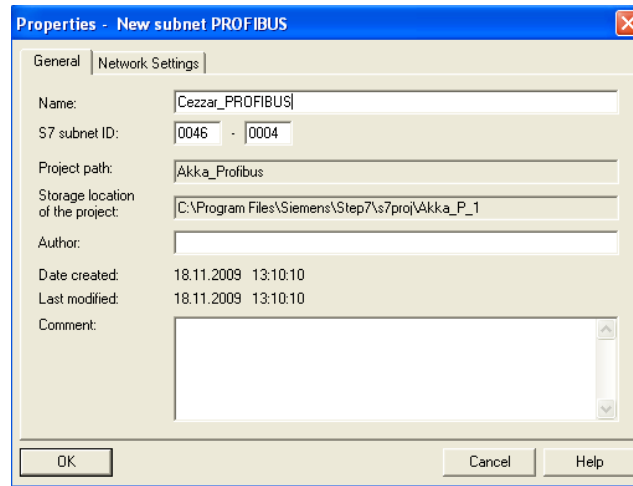
Hardware üzerinde çift tıklayalım. "HW Config" penceresi açılmıştır. PLC donanımı tanımlanır. Bu örnekte CPU 313C-2DP PLCnin 6ES7 313-6CF03-0AB0 modelinin 2.6 modeli kullanılacaktır.



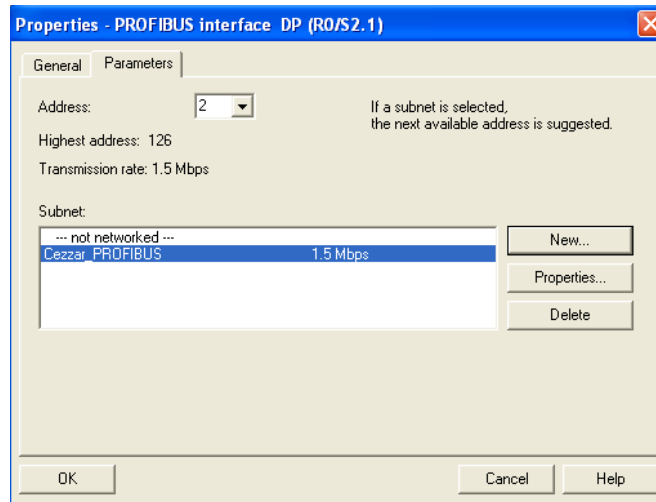
4. Donanım eklendikten sonra DP üzerinde çift tıklanır.



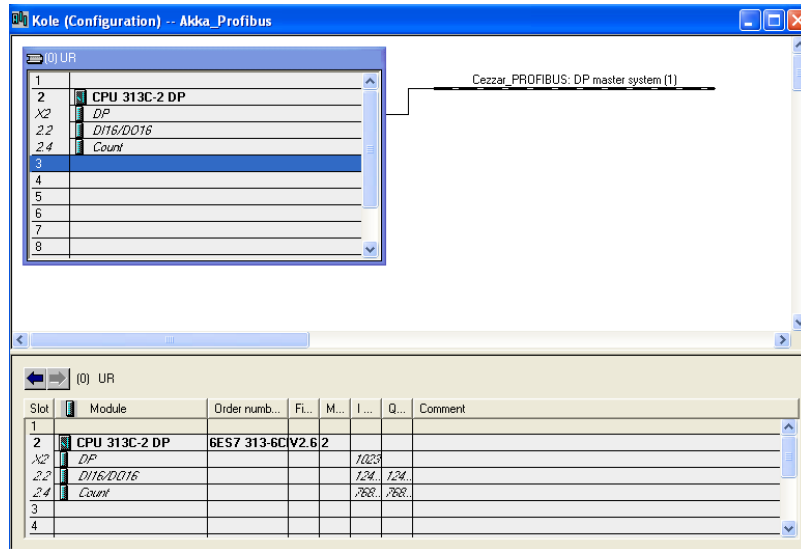
5. New düğmesine bir fiske aşkedin.



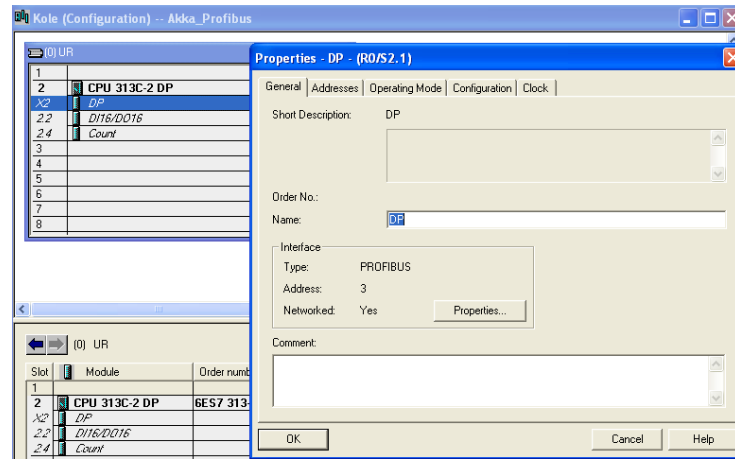
6. Bir isim yazarak OK deyin.



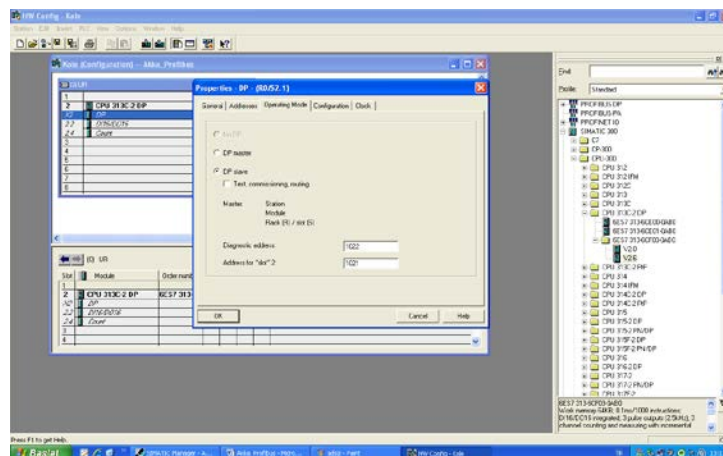
7. Adres alanını 3 yapın.



8. DP alanını çift tıklayın.



9. Operating Mode sekmesinde Mod olarak DP Slave'i işaretleyin.



10. Configuration sekmesine girerek New işaretlenir.

Properties - DP - (R0/S2.1) - Configuration - Row 1

Mode: MS (Master-slave configuration)

DP Partner: Master

DP address:

Name:

Address type:

Address:

"Slot":

Process image:

Interrupt OB:

Local: Slave

DP address: 3

Name: DP

Address type: Input

Address: 0

"Slot":

Process image: OB1 PI

Diagnostic address:

Length: 1

Unit: Byte

Consistency: Unit

Comment:

OK Apply Cancel Help

11. Kolenin profibus giriş ve çıkış adreslerini tanımlayacağız.

Verilerin 1 bayt uzunluğunda olacağını kabul ederek address Type alanında Input için 0 yazarak OK diyelim.

Properties - DP - (R0/S2.1)

General | Addresses | Operating Mode | Configuration | Clock

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste...
1	MS	--	--	I 0	1 Byte	Unit

New... Edit... Delete

MS Master-slave configuration

Master:

Station:

Comment:

OK Cancel Help

12. Tekrar New diyelim. Bu sefer Output için 0 verelim

Properties - DP - (R0/S2.1)

General | Addresses | Operating Mode | Configuration | Clock

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste...
1	MS	--	--	I 0	1 Byte	Unit
2	MS	--	--	Q 0	1 Byte	Unit

New... Edit... Delete

MS Master-slave configuration

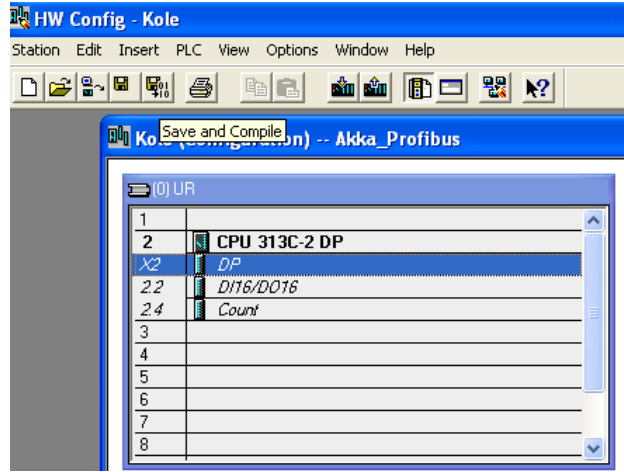
Master:

Station:

Comment:

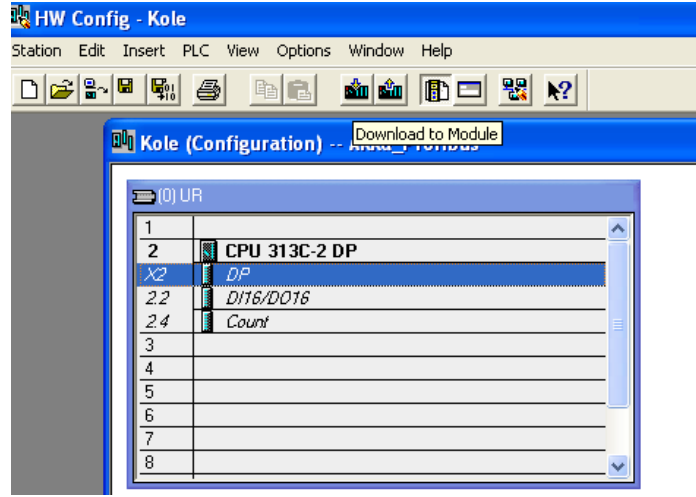
OK Cancel Help

13. Yapılan ayarları kaydedelim.



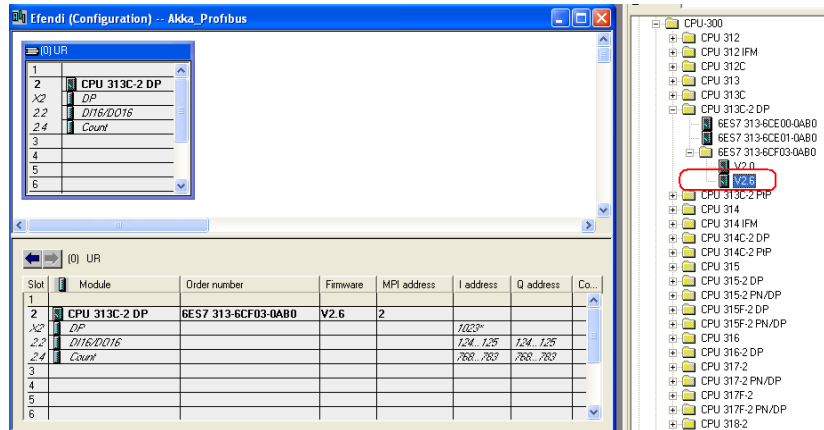
14. Donanım ayarlarını gönderelim.

15.

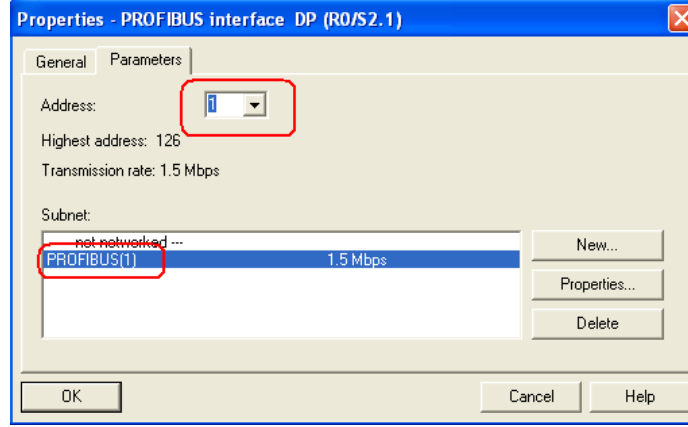


Programı Master PLCyi ekledikten sonra yazalım.

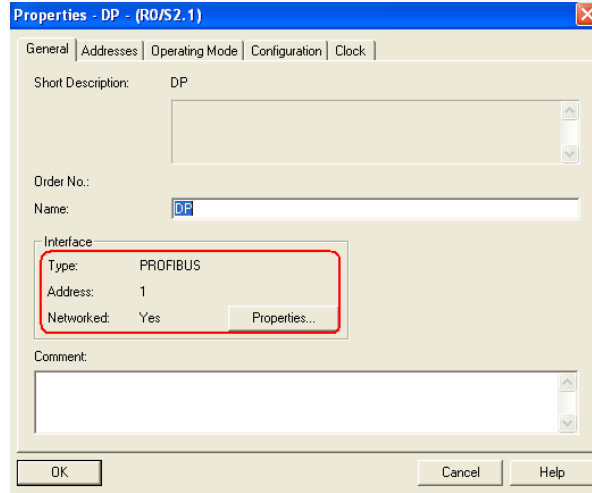
16. Projeye yeni bir PLC ekleyerek Efendi ismini verelim. Hardware ayarlarını yapalım.



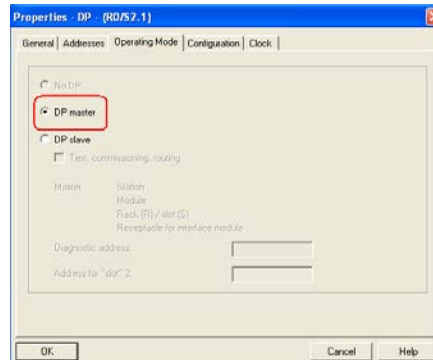
Ray üzerine PLCyi koyar koymaz daha önce ayarlanan PROFIBUS ağına dahil olup olmayacağımız sorulur. Seçerek yola revan olalım. Gösterilen diyalog kutusunda önerilen adresleri uygulayınız.



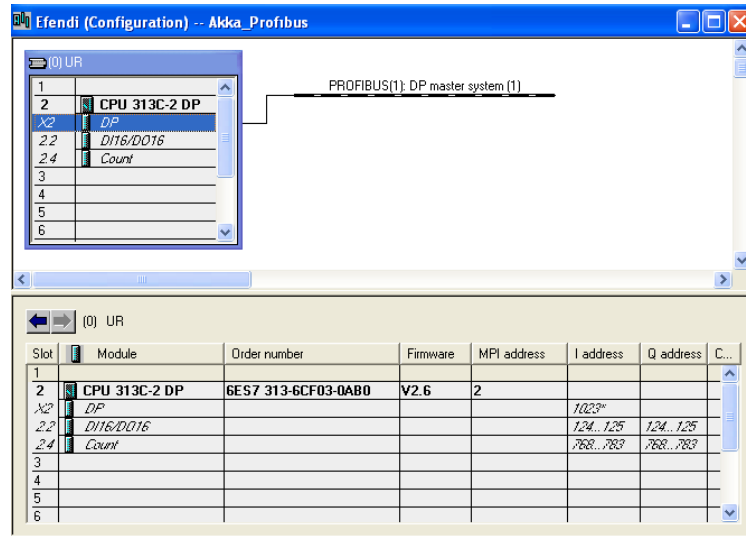
17. PLCnin DP alanı üzerinde çift tıklayalım.



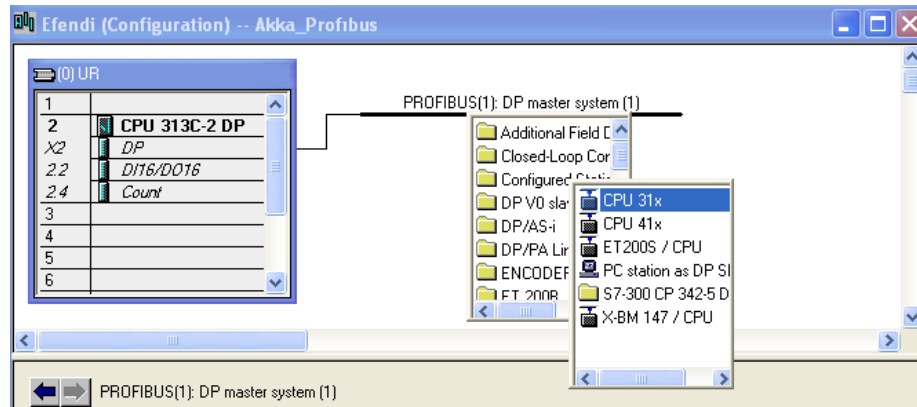
18. Görüldüğü gibi PROFIBUS ağına dahil oldu. Operating Mode sekmesi tıklanır. DP Master işaretli olmasına özen gösterilir.



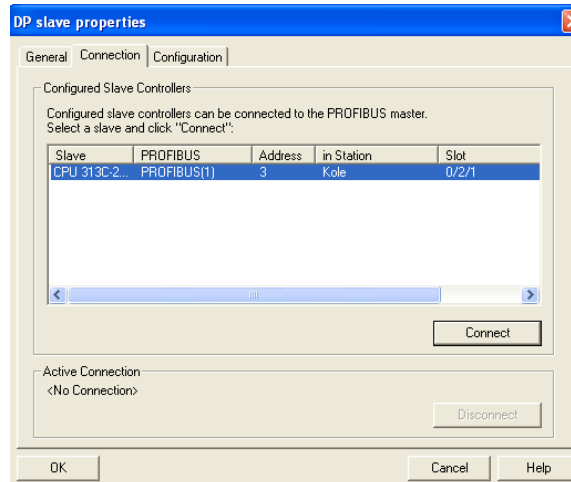
Onay verildikten sonra PLCye bağlı bir kablo çizilir. Bu PROFIBUS hattıdır. Şimdi buna Kole isimli PLCyi bağlayalım.



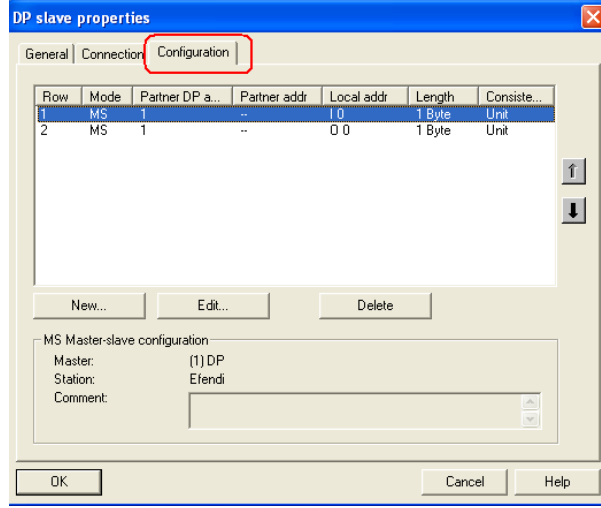
19. Kablo üzerinde farenin sağ tuşu ile tıklayalım. Insert Object seçilir. Configured Stations'dan CPU31x işaretlenir.



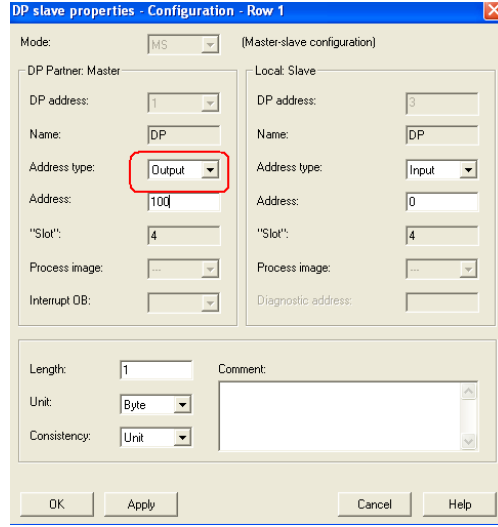
20. Ekran gelen pencerede Connect tıklanır.



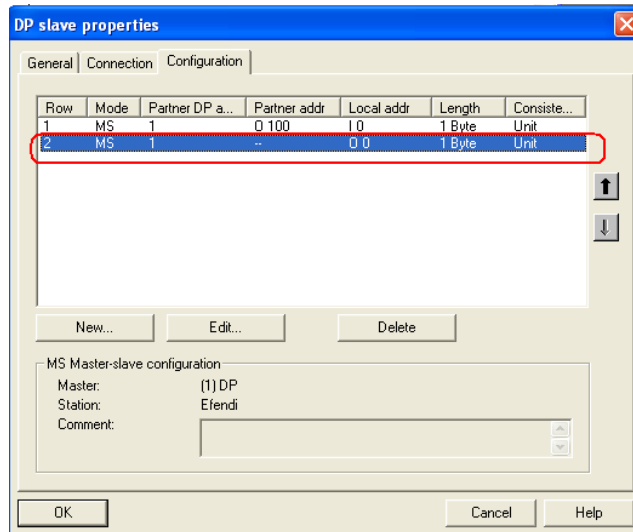
Configuration sekmesi tıklandığında daha önce Slave için yaptığımız hafıza ayarları ekrana zuhur eder.



21. Edit seçeneği tıklanarak Master için Output=100 verilir. Onay verilir.



22. Bir alttaki satıra geçilerek Edit işaretlenir ve Master için Input=100 verilir.



DP slave properties - Configuration - Row 2

Mode: MS (Master-slave configuration)

DP Partner: Master

DP address: 1

Name: DP

Address type: Input

Address: 100

"Slot": 5

Process image: ---

Interrupt OB: ---

Local Slave

DP address: 3

Name: DP

Address type: Output

Address: 0

"Slot": 5

Process image: ---

Diagnostic address: ---

Length: 1

Unit: Byte

Consistency: Unit

Comment:

OK Apply Cancel Help

23. Onaylanır.

DP slave properties

General Connection Configuration

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste...
1	MS	1	0 100	0 0	1 Byte	Unit
2	MS	1	1 100	0 0	1 Byte	Unit

New... Edit... Delete

MS Master-slave configuration:

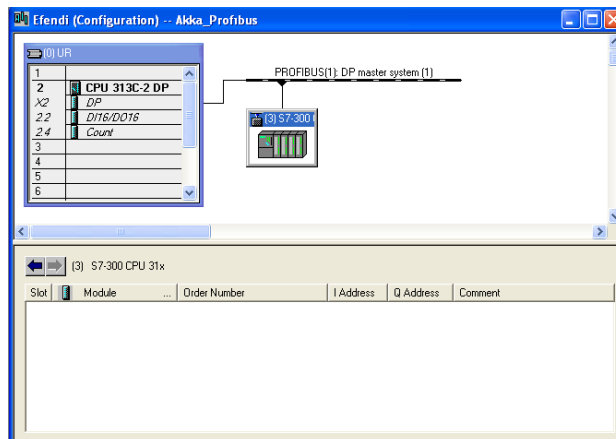
Master: (1) DP

Station: Efendi

Comment:

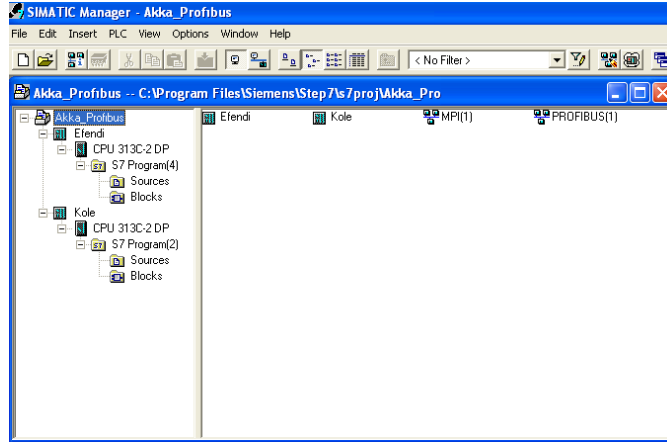
OK Cancel Help

Hafıza tablosu ayarlanmıştır.
Görüldüğü gibi efendi-köle birbirine raptedilmiştir.



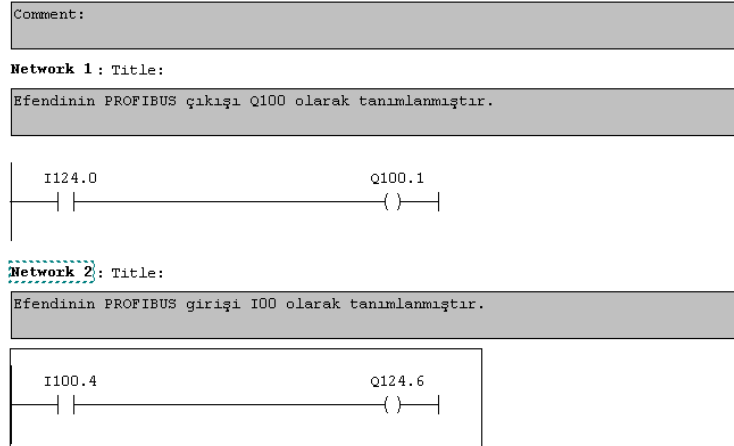
24. Bu yarıları da master PLC ye gönderelim.

Proje alanımıza dönelim.

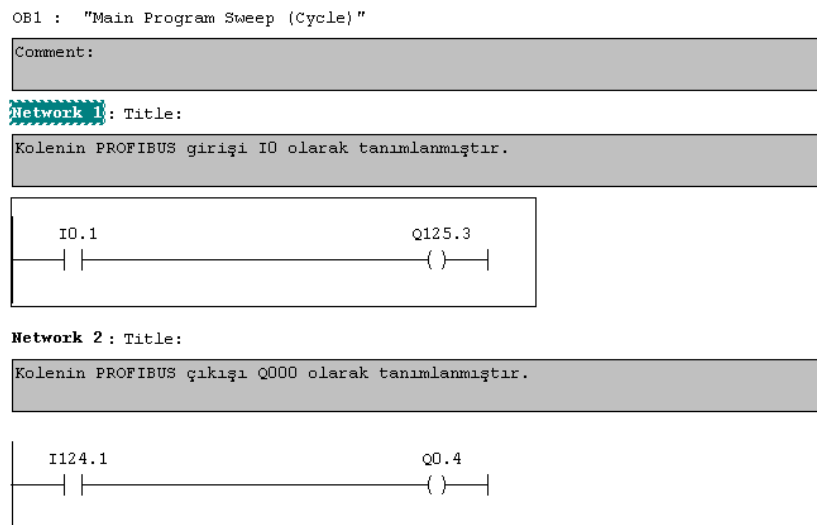


Master için programızı yazalım. Program mantığımız oldukça basittir. Efendideki bir giriş düğmesine tıkladığımızda Kole'nin herhangi bir çıkışı yanacak. Kole'deki bir girişe basıldığında Efendi'de bir çıkış yanar.

25. Efendi programı:



26. Kole programı:



GSD-Dataları

GSD, ağdaki her Profibus cihazının kimliğidir ve cihazın teknik özellikleri, haberleşme yetenekleri, diyagnoz değerleri gibi verileri içerir.

Profibus-Yönetmeliği:

Profibus-DP/FMS montaj yönetmeliğinde belirtilen kurallara uyulmalıdır.

1. Fonksiyon Topraklama (FE): Bir cihazın sahada topraklanmasına denir ve bilinen topraklamadan (PE) farklıdır. PE insanların ve cihazların korunması için olmakla beraber FE ekranlamada vs. oluşan elektrostatik yüklerin deşarjını sağlar.
2. EMV: Sistemdeki kablolar ve SAI-Arayüze çalışma esnasında elektromanyetik darbe etki eder. Bu durum verileri etkileyebilir ve sistem hatalarına sebep verebilir. SAI-Aktif Üniversal serisinin arayüzlerinde BUS-POWER kısmının tespit deliğinde FE bağlantısı öngörülmüştür. SAI-Arayüz iletken bir yüzey üzerine veya düşük empedanslı bir kablo ve pabuç ile tespit civatasına monte edilerek FE koruma fonksiyonu kullanılabilir. PE kesinlikle FE yerine kullanılmamalıdır.
3. Profibus-Adresleme: SAI-Arayüzün Profibus-DP ağında hangi adres ile bulunacağını gösterir. SAI-Arayüzün adresleme kısmında bulunan lojik anahtar üzerinden Profibus-DP adresi ayarlanmalıdır.
4. Adres/Hexadesimal-Kod: Profibus-DB maksimum 126 farklı adresi tanıyabilmekle beraber 1-125 arasındaki adresler tanımlanmıştır. 126 numaralı adres konfigürasyon ayarları için kullanılmaktadır. 01 ve 02 numaralı adresler Profibus-Master için rezerve edilmiştir. Arayüz adresi 2 adet lojik anahtar üzerinden Hexadesimal olarak ayarlanır. Öncelikle desimal adres hexadesimal adrese çevrilmelidir. (Tablo 1) Örneğin 93 numaralı Profibus-DP adresi Hexadesimal 5D'ye tekabül eder. (1. şalter: 5 ve 2. şalter: D)
5. Profibus-DP'de her adres sadece bir defa kullanılmalıdır ve Proje yazılımında kayıtlı adresler ile aynı olmalıdır.

İnternet Kaynakları:

<http://www.istanbulbilgi.com.tr/BB/Profibus-Teknolojisi.htm>
www.bilesim.com.tr/yazdir.php?t=3&id=1442&sn=0

CAN-BUS

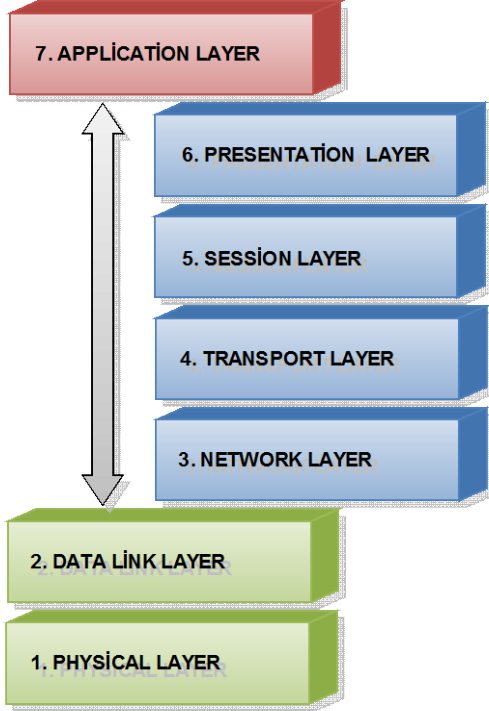
Açılımı "Controller Area Network" olan CAN BUS, 1980'lerde Robert Bosch tarafından otomotivde kablo yumağı yerine bir kablodan yazılım kontrollü veri transferini sağlamak amacıyla geliştirilmiştir. CAN otomotiv endüstrisindeki en bilinen haberleşme sistemidir. Her ne kadar başlangıçta yalnızca otomotiv endüstrisi uygulamaları için tasarlanmış olsa da yüksek performansı güvenilirliğinden dolayı bir çok dağıtık endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır. Güvenliğin çok önemli olduğu gerçek zamanlı uygulamalarda da kullanılır. Öyle ki istatistiksel olasılık hesapları sonucunda bir asırda bir tane tespit edilemeyen mesaj hatası yapabileceği tespit edilmiştir.

Uygulama alanı yüksek hızlı ağlardan düşük maliyetli çoklu kablolamalı sistemlere kadar geniştir. CAN-BUS, otomotiv elektroniği, akıllı motor kontrolü, akıllı sensörler ,

asansörler, makine kontrol birimleri, kaymayı engelleyici sistemler, trafik sinyalizasyon sistemleri, akıllı binalar ve laboratuvar otomasyonu gibi uygulama alanlarında maksimum 1Mbit/sn lik bir hızda veri iletişimi sağlar.

CAN-BUS iletişimini OSI referans modeli üzerinden açıklamak istersek,

- Nesne katmanı (Object layer)
- İletim katmanı (Transfer layer)
- Fiziksel katman (Physical layer) şeklinde üç alt bölüme ayırabiliriz.



Nesne ve iletim katmanları, data link layerin tüm servis ve fonksiyonlarını oluşturur.

Nesne katmanının görevleri,

- Hangi mesajın transfer edileceğini tespit etmek
- İletim katmanında hangi mesajın alınacağına karar vermek
- Donanımla ilgili uygulamaya arayüz sağlamaktır.

İletim katmanının başlıca görevi transfer protokolüdür. Örneğin çerçeve (frame) kontrolü, mesaj önceliği belirleme, hata kontrolü, hata sinyalleşmesi ve hata kapatma gibi... İletim katmanı yeni bir mesaj yollamadan önce iletim hattının boş olmasına dikkat eder. İletim hattının veri alınmasından da sorumludur. Ayrıca senkron iletişim için veri transferi sırasında bit zamanlamasının bazı parametrelerini göz önünde bulundurur.

CAN-BUS üzerinden haberleşen tüm sistem bileşenlerine ünite (nod) denir. Örneğin 10 adet PLC cihazınızı bir CAN-BUS iletişiminde haberleştirdiğinizi varsayalım. Bu sistemde her PLC bir üniteyi oluşturur.

Fiziksel katman ise üniteler arasında veri haberleşmesi sırasındaki tüm elektriksel kısımdır.

Can-Bus Sisteminin Özellikleri

CAN-BUS sisteminde tüm üniteler iletim hattına eşit öncelikli veri yollama hakkına sahiptir. Buna multimaster çalışma denir. Bu sistemde her ünite bir anda veri yollamaya çalışırsa çatışmalar

meydana gelir. Bu durumu önlemek için tüm ünitelerin her an iletim hattını dinler ve hattın boş olduğu anı yakalamaya çalışır. Hattı boş gören ünite kendisine ait veriyi yollar. Tüm ünitelerin eşit mesaj yollama özelliği bu iletişim sisteminin mesaj öncelikli sistem olmasından kaynaklanır. İnternette tüm bilgisayarların bir numarası (IP) vardır ve veriler o numaraya ait kişiye yollar. CAN-BUS sisteminde ise ünitelere değil mesajlara numara verilir.

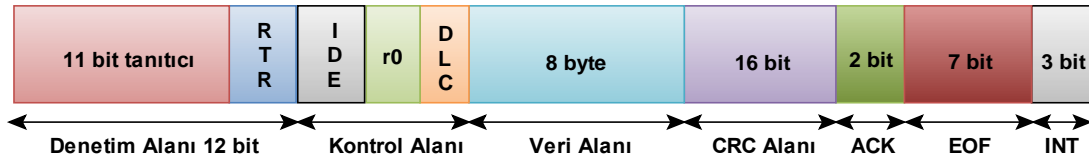
Örneğin yolda kaza yapma ihtimali olan bir otomobili düşünelim. Sürücü çarpacağını fark eder ve frene basar. Burada CAN-BUS sisteminin özelliğini aracın çarpışması durumuna göre inceleyelim. Önden bir çarpışma durumunda sensörler devreye girer ve hava yastıklarının açılması, yakıtın kesilmesi gibi bir dizi önlemler alınmalıdır. Bu gibi işlerden sorumlu tüm ünitelere 1 no'lu mesaj (kaza oldu güvenlik sistemleri devreye girsin) iletilmelidir. Bunu diğer ünitelere sensörlü sistem söyleyecektir. Ancak o arada başka bir ünite motor ısısının kaç derece olduğunu sürücü göstergesindeki ekrana bağlı üniteye göndermeye çalışsın. Bu durumda CAN-BUS sistemi tek kablo üzerinden eşit erişimli mesaj yollanmasına izin verir fakat öncelikli mesajların daha önce iletilmesi iletim çakışmalarını önler.

CAN, iletişim ortamına erişim yöntemi olarak bit öncelikli yapı ile CSMA/CD kullanır. Bu yöntem , mesajların çarpışmamasını garanti etmekler beraber, iletişim hattının uzunluğunu sınırlandırır. Dolayısıyla, CAN düğümleri 1Mbit veri iletim hızı ile 40 m ve 40 Kbit/s veri iletim hızı ile 1000 m'lik bir veri yolu üzerinden bağlanabilir.

CSMA/CD (Carrier Sense Multiple Access with Collision Dedection) yapısı CAN sisteminin daha güvenli olmasını sağlar. Bu sistemi kısaca açıklayalım. CAN sistemindeki her ünite iletim ortamını dinlemek zorundadır. Eğer bir ünite iletim ortamına veri aktarmak isterse önce yolu dinler, yol boş ise veriyi iletir. Bazen uzak mesafelerden kaynaklanan sebeplerden dolayı bir ünite yola veri aktarırken, uzaktaki ünite diğer mesajın yolda olduğunu anlayamadığından kendi mesajını da yolu boş zannederek bırakır. Böylece çatışma (collision) oluşur. İki ünite de çatışmayı sezer ve veri aktarımını bir süre bekletir. Daha sonra iletim ortamı boş olduğunda verileri tekrar yollamaya çalışırlar.

CAN-BUS sistemine gönderilen tüm veriler, ağdaki her ünite tarafında alınır. Üniteler içinde bulunan filtre yardımıyla kendisini ilgilendiren mesajları alır ilgilendirmeyen mesajları siler.

Can-Bus Veri Paket Yapısı



Mesaj ID Alanı (Arbitration field)

CAN sistemlerinde veriler paketler halinde iletilir. Ancak iki tip paketleme yapılır ve özel adları vardır. 11 bit tanımlayıcıya sahip olanlar CAN 2.0A diğer adıyla standart CAN, 29 bit tanımlayıcıya sahip olanlar ise CAN 2.0B extended (geliştirilmiş) CAN denir. Aralarındaki temel fark, tanımlanabilecek mesaj sayısıdır. Standart CAN sisteminde $2^{11}=2048$ mesaj tanımlanabilirken geliştirilmiş Can sisteminde $2^{29}=536870912$ mesaj tanımlanabilir. Bu bilginin tutulduğu alana mesaj ID alanı adı verilir. Mesaj önceliğini belirlemek için buradaki sayı dikkate alınır. Ayrıca mesaj ID alanı RTR biti de içerir. Eğer 1 ise gönderilecek pakete istek çerçevesi (remote frame) denir. Eğer 0 ise veri çerçevesi (data frame) denir. Bir ünite neden data frame yollar? Bu sorunun yanıtı kendisinde bulunan veriyi iletmektir. Bir ünitenin istek çerçevesi yollamasının nedeni, ünitelerin başka ünitelerden gelecek bilgiye ihtiyaç duymalarıdır. Karşı tarafa "bana şu bilgiyi yolla" demenin yolu istek çerçevesi kullanmaktır. İstek çerçevesi yollayan bir ünitenin veri alanı (data field) yoktur. Çünkü veri istemektedir. İstenen veriye ilişkin mesaj ID alanında vardır.

Kontrol Alanı (Control field)

6 bitten oluşur. İlk biti standart veri paketlemesi mi ya da geliştirilmiş veri paketlemesi mi yapıldığını belirtir. Ayrıca veri alanının kaç byte değerinden oluştuğunu belirten bitlere sahiptir.

Veri Alanı (Data field)

CAN –BUS iletişimde en fazla 8 byte bilgi iletilebilir. Bu değer 8 byte dan daha az olabilir. Gönderilecek veri uzunluğu kontrol alanında belirtilir.

Dönüşsel Artıklık Kontrol Alanı (CRC field)

15 bitlik CRC sequence ve CRC delimiterden oluşur. Görevi pakete ait CRC kodunu tutmaktır. Üniteye gelen pakete cevap verilebilmesi için ilk önce paketin doğruluğu kontrol edilmelidir. Bunun için de ilk önce alınan paketin CRC paketi hesaplanır. Daha sonra alıcı ünite paket ile birlikte gelen CRC değeri ile hesaplanan değeri karşılaştırılır. İki değer birbirine eşit ise alınan paket geçerlidir.

Eğer iki değer, birbirine eşit değilse mesajı alamadığını belirtmek için CRC hatasından kaynaklı hata oluştuğunu belirten hata çerçevesi (error frame) yollar. Bu bilgiyi alan gönderici veriyi tekrar yollamaya çalışır. Bu bilgiyi ünitelerden sadece biri yollasa bile veri tekrar yollanmalıdır.

Alındı Bilgisi Alanı (ACK field)

Bunu bir mesaj iletimiyle açıklayalım. Gönderici başla biti ile iletim hattında şu an gönderici benim der. Ardından mesaj ID alanı, kontrol alanı, CRC alanı gönderilir. Alındı bilgisi alanında ise iletim ortamı çekinik tutulur. Eğer diğer tüm ünitelerden biri, mesaj onu ilgilendirse yada ilgilendirmese dahi mesajı alabiliyorsa iletim ortamını baskın yapar ve böylece gönderici en az bir ünite veriyi alabildiği için bitir bitini yollayıp iletim ortamını diğerlerinin kontrolüne bırakır. Yani alındı bilgisi alanında “aldınız mı?” sorusuna yanıt beklenir. Eğer alındı bilgisi sürecinde herhangi bir üniteden alındığına dair bilgi alamazsa ACK hatasından kaynaklı hata oluştuğunu belirten hata çerçevesi üretilir ve gönderici tekrar yollamaya çalışır. Eğer gönderen istek çerçevesi yollamışsa, alıcı da iletim hattının boş bir anında cevabını göndericiye yollar.

Hata Çerçevesi (Error frame)

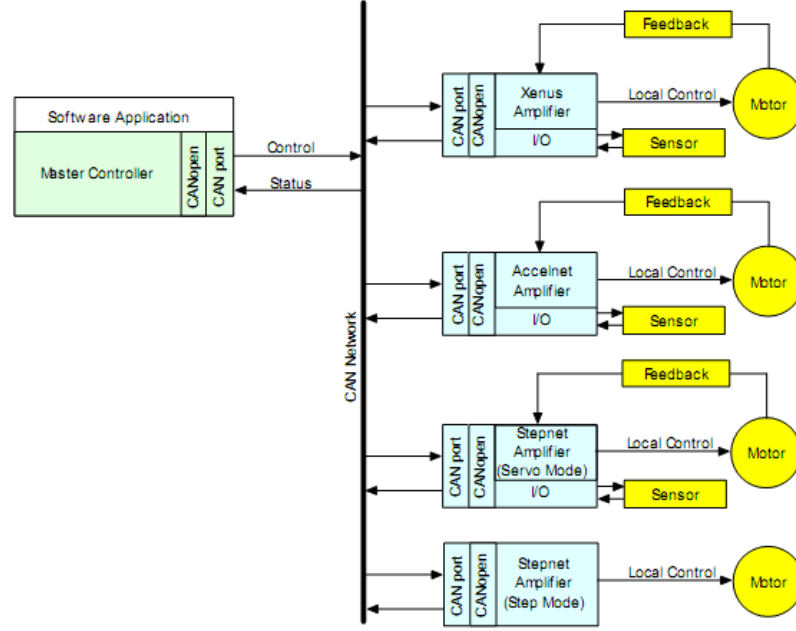
Veri çerçevesi veya istek çerçevesinin gönderiminde ya da alımında hata oluştuğunda gönderen veya alıcılar tarafından ne tip hatanın olduğunu gösteren mesaj çerçevesidir.

Can-Open

Birçok endüstriyel iletişim protokolü, CAN-BUS iletişim yapısına dayanır. Bunlara örnek olarak Can-Open ve devicenet iletişim protokollerini örnek gösterebiliriz. Can-Open iletişimi, motor sürücüleri (motion control) gibi değişik tipteki endüstriyel cihazların iletişimde kullanılır. Aşağıdaki şekilde de görüldüğü gibi master denetleyici iletişim ağı boyunca birçok cihaza komutları gönderdiği gibi durum bilgilerini de alır. Ağdaki her cihaz, master denetleyiciye ve diğer cihazlara veri gönderebilir. Bunu sağlamak için Can-Open, cihazların çalışma profillerini ve komutlarını ağ içinde paylaşan bir protokole sahiptir.

Can-Open, Can-Bus iletişiminin uygulama katmanında (application layer) tanımlıdır. Uygulama katmanı, ağ içindeki cihazlar arası veri alışverişini sağlamak için iletişim

nesnelerini (communications object) kullanır. Can-Open uygulamaları iletişim nesneleri üzerine oluşturulur.



Can-Open Özellikleri

- Can-Open , Can-Bus üzerinde seri iletişim yapısına sahiptir.
- 1Mbps hızda 25 m'ye kadar veri iletimi sağlanabilir.
- Çoklu master özelliği kullanılabilir.
- 127 bağlantı noktasını destekler.
- Değişik türdeki endüstriyel cihazlarda kullanılabilir.

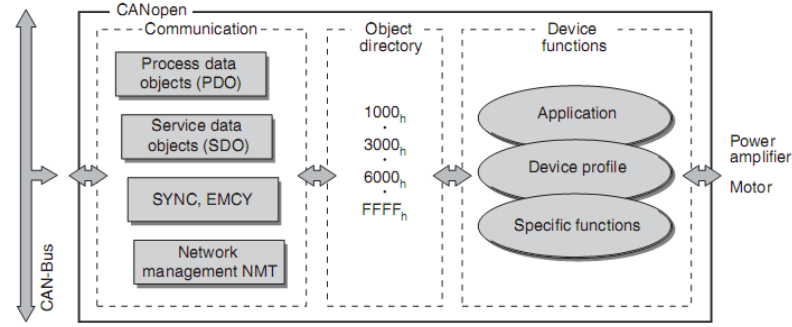
Can-Open İletişimi

Aşağıda, master PLC'nin bir servo motor sürücüsü ile olan Can-Open iletişim basamakları görülmektedir.

- Can-Open master, ağdaki tüm cihazları başlatmak için bir kontrol word bilgisi yollar.
- Cihazlar, çalışma durumları gösteren bir mesaj yollar.
- Can-Open master, cihazlara homing (servo sistemin başlangıç konumuna getirme) çalışma durumunu yürütmek için gerekli komutları yollar.
- Cihaz (servo sürücüsü), homing işleminin bittiğini gösterir.
- Can-Open master, cihazın point to point (noktadan noktaya) çalışma modunu ve bu mod için gerekli koordinatları sağlayan komutları gönderir.
- Cihaz, servo motorun anlık durum bilgisine ve hızına göre hareketini sağlar. Bunun yanında pozisyon bilgisini de Can-Open ağına gönderir.
- Master cihaz, bundan sonraki hareketin koordinatlarını yürütür.

Can-Open iletişimde tüm işlemler iletişim nesneleri aracılığı ile yürütülür. Nesneler, veri iletişimi, bağlantı yapılandırma veya ağdaki cihazların durumlarının izlenmesi gibi görevleri yürütürler.

Can – Open iletişimde tüm işlemler, nesne kütüphanesi tarafından yürütülür. Ağ içindeki cihazlar çalışma özelliklerin göre gerekli olan nesneleri kullanır. Nesneler, veri tiplerine, cihaz fonksiyonlarına ve iletişimdeki rollerine göre sınıflandırılır.



Nesne kütüphanesindeki tüm nesneler, 16 bit olarak tanımlanmıştır ve hegzadesimal olarak 4 karakterli grup şeklinde gösterilir. Nesneler kullanıldıkları işlemlere göre gruplandırılması aşağıdaki tabloda açıklanmıştır.

İndeks aralığı (hex)	Nesne grubu
1000h – 2FFFh	İletişim profili
3000h – 5FFFh	Üretici firmaya özel nesneler
6000h -9FFFh	Standart nesneler
A000h -FFFFh	Ayrılmış

Can-Open iletişimindeki standart çalışma profilleri, sisteme uygun şekilde üretilen tüm cihazlarda (PLC, İnverter, sürücü, ağ elemanları vb.) kullanılabilir. Bu profillerin standart hale getirilmesinden CAN in Automation (CIA) birliği sorumludur.

CIA birliğinin standart olarak sunduğu bir çok endüstriyel profil vardır. Bunlara örnek olarak IO modülleri (CIA 401), ölçüm cihazları ve kapalı çevrim denetleyiciler (CIA 404), lineer ve dairesel enkoderler(CIA 406), iş makinelerini (CIA 422) verebiliriz.

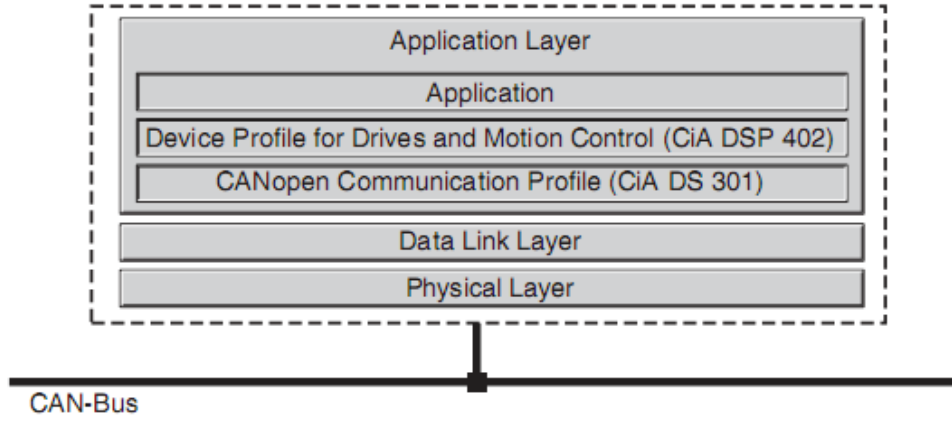
Aşağıda can-open iletişim profili ve motor sürücüleri için hazırlanmış cihaz profili açıklanmıştır.

İletişim profili (DS 301) : Bu profil, cihazlar ile Can iletişimi arasındaki çalışma sistemini oluşturur. 1995 yılında, DS301 olarak tanımlanmış olan bu standart, değişik tür ve marka cihazların ortak veri yapısını Can-Open altında düzenler. İletişim nesneleri, cihazların, ağdaki diğer cihazlarla olan veri ve parametre iletişimi görevlerini yürütür. Ayrıca cihazın ilk çalışma durumunu düzenler ve bu durumları ağ içinde izler.

Cihaz profili (motion controller) (DSP 402) : Cihaz profili, ağdaki cihazların konum kontrolünü, hareketin izlenmesini ve çalışma modlarının ayarlamalarını yapar. Nesnelerin görevleri aşağıdaki şekilde sınıflandırılabilir.

- Cihaz kontrolü ve izlenmesi
- Standart parametre ayarları
- Çalışmayı başlatma, veri doğrulama ve çalışma modlarının yürütülmesi

Üretici firmaya özel profiller : Temel çalışma durumları, standart profillerle sağlanabilir fakat özel ve karmaşık işlemler için firmalar farklı fonksiyonlar sunar.

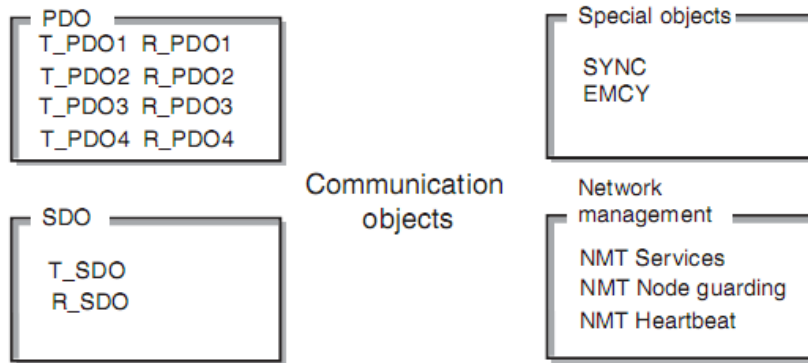


Her nesne, 16 bit indeks ve 8 bitlik alt indekse sahiptir. 8 bitlik alt indeks, nesnenin özel çalışma durumu belirler. Örneğin 607Dh nesnesi, servo motorda “software position limit” olarak tanımlanmıştır. Bu nesnenin alt indeksleri ise aşağıdaki gibidir.

İndeks	Alt indeks	İsim	Anlamı
607Dh	00h	-	Veri alanı
607Dh	01h	min. Position limit	Alt sınır anahtarı
607Dh	02h	max. Position limit	Üst sınır anahtarı

İletişim Nesneleri (Communication objects)

İletişim nesneleri, DS301 standardı altında sınıflandırılmıştır. Bu nesneler dört gruba ayrılır.



PDO (Process Data Object) : Yapılan işlemler için gerekli verilerin iletimini gerçek zamanlı olarak sağlar. T_PDOx, gönderilen nesneleri, R-PDOx ise alınan nesneleri sembolize eder.

SDO (Service Data object) : Nesne kütüphanesine erişim için kullanılan nesnelerdir. Bu şekilde okuma ve yazma işlemi yapılabilir. T-SDO nesne kütüphanesine yazma işlemini, R-SDO ise okuma işlemini açıklar.

SYNC object (synchronisation object) : Ağ içindeki cihazlar arasındaki senkronizasyonunu yapar.

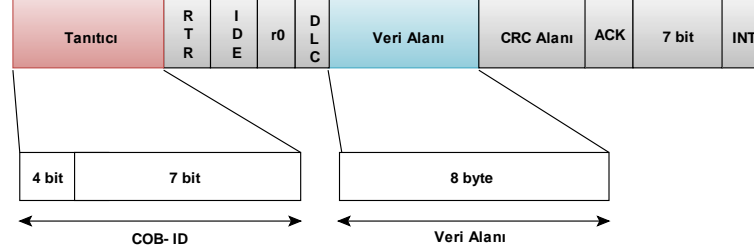
EMCY object (emergency object) : Ağ içindeki cihazları ve cihazlara entegre elemanların hata mesajlarını gösteren nesnedir.

NMT services : Ağ yönetimi için kullanılan nesnedir.

NMT Node Guarding -NMT heartbeat : Ağı izlemek için kullanılan nesnelerdir.

Can-Open Veri iletişimi

Can mesaj yapısını daha önce açıklamıştık. Şimdi mesaj üzerinde veri iletimini inceleyelim.



Can mesajı içindeki tanıtıcı bölümü 11 bit değerindedir ve COB-ID (communication object identifier) olarak adlandırılır. COB-ID, Can-Open nesnelerinin iletim önceliğini ve nesnelerin tanımlanması görevini yürütür. İletişim nesnelerinin COB-ID değerleri ağıdaki cihazların bağlantı adreslerine (node id) göre adlandırılır. Örneğin, ağıdaki 5 no'lu cihazın T_PDO1 nesnesinin COB-ID numarası

$$384 + \text{nod-id} = 384(180h) + 5 = 389 (185h) \text{ olarak tanımlanır.}$$

Can mesajında verileri ise en fazla 8 byte değerindeki paketler halinde iletilir.

Can-Open Uygulaması

Uygulamamızda Telemechanique marka Twido modüler plc ile yine aynı firmanın Lexium 5 servo motor sürücüsünün kontrolü anlatılacaktır.

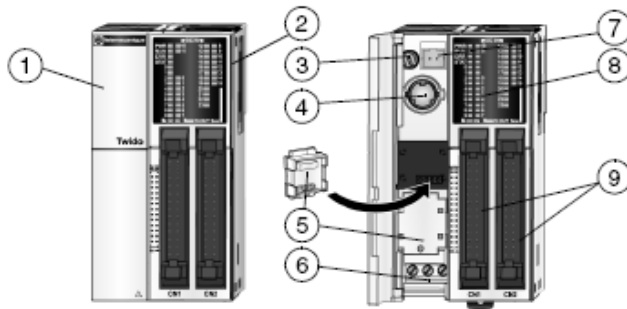


CANopen



Twido modüler

Uygulamada kullanılan Twido PLC'nin (TWD LMDA 20DTK) özellikleri şunlardır.



- 1) Kapak
- 2) Genişleme konektörü
- 3) Analog potansiyometre
- 4) Seri port
- 5) Kartuş kapağı
- 6) 24 V terminal
- 7) Analog giriş konektörü
- 8) LED
- 9) Giriş-çıkış terminalleri
- 10) İletişim konektörü

Girişler	12 adet
Çıkışlar	8 adet (transistör)
Genişleme	4 modül
Entegre fonksiyonları	1 potansiyometre, 2 pozisyon kontrol 1 analog giriş
Arayüz	RS-485
Komut	3000

Twido PLC' ye ise TWNCO1M Can modülü eklenmiştir. Bu modül ile birlikte, 125-250-500 kbit/s hızlarında iletişim yapılabilir. Bu modüle 16 adet slave cihaz bağlanabilir.



Lexium 5 Servo Motor Sürücüsü

Lexium 5 servo motor sürücüsü, bir çok endüstriyel alanda uygulanabilecek farklı çalışma fonksiyonlarına sahiptir. Bu fonksiyonları 2 grupta inceleyebiliriz.

- 1) Kendi üzerinden yapılan ayarlar
 - Homing (servo motorun başlangıç pozisyonu)
 - Jog (Manuel mod)
 - Auto tuning (servo motorun ve sürücüsünün otomatik ayarları)
- 2) Çalışma modları
 - Pozisyon kontrolü
 - Point to point (servo motorun belirli bir noktadan diğer noktaya hareketi)
 - Motion sequence mod (programlanabilir parametrelili hareket)
 - Electronic gearing (pulse pozisyon kontrolü)
 - Hız kontrolü
 - Hız kontrolü için hızlanma ve yavaşlama
 - Anlık hız kontrolü
 - Akım kontrolü
 - Akım regülasyonu

Lexium 5 servo motor sürücüsünün kontrolü 2 şekilde yapılabilir.

- 1) Lokal çalışma
- 2) Endüstriyel ağlar ile

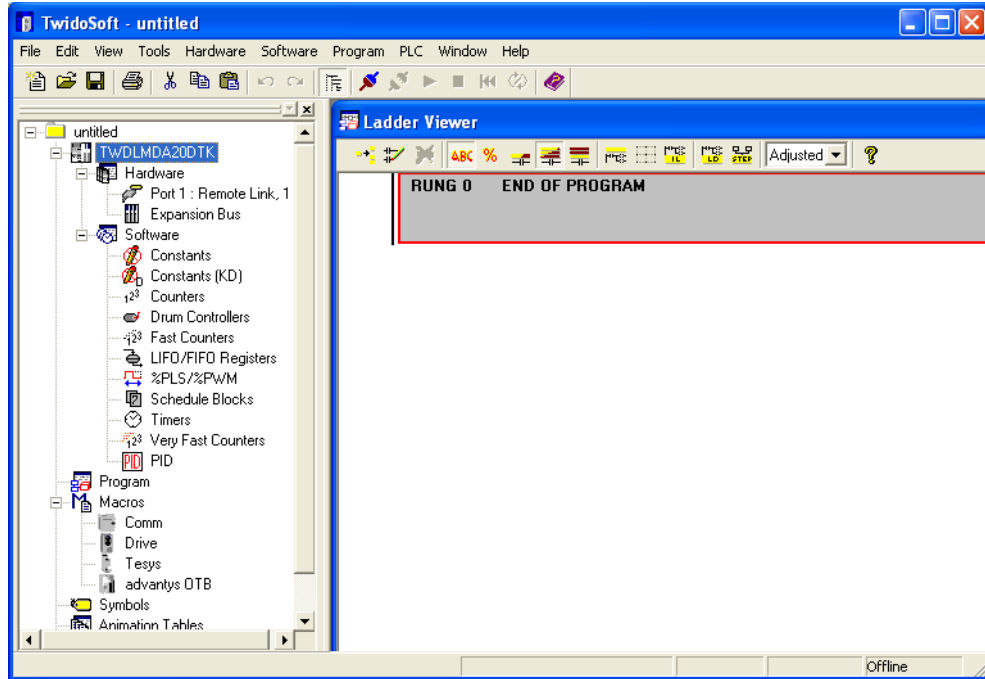
Aşağıdaki tabloda servo sürücü fonksiyonlarının kullanımı görülmektedir.

Çalışma Fonksiyonu	Kontrol Şekli		Veri İletimi
	Endüstriyel Ağ	Lokal Çalışma	
Servo Sürücü Ayar Fonksiyonları			
Homing	x		İletişim ağı veya power suite programı
Jog	x	x	İletişim ağı, power suite programı veya kullanıcı arayüzü
Auto tuning	x	x	İletişim ağı veya power suite programı
Çalışma Fonksiyonları			
Point to point	x		İletişim ağı veya power suite programı
Motion sequence	x	x	İletişim ağı veya power suite programı
Electronic gearing	x	x	Pulse/direction (P/D), A/B veya CW/CCW sinyali
Speed control with ramp	x		İletişim ağı veya power suite programı
Current control	x	x	Analog giriş,iletişim ağı veya power suite programı

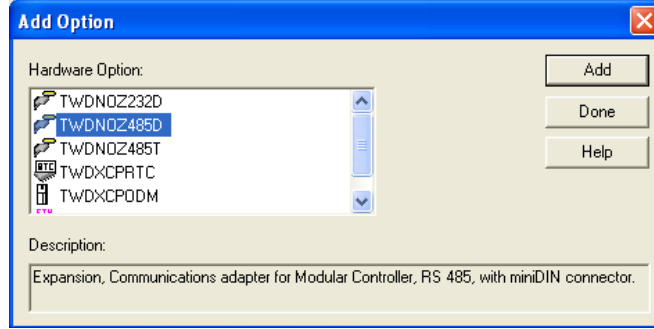
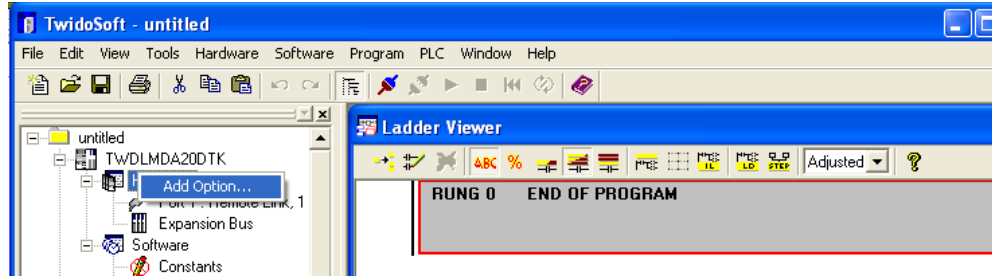
PDO Kullanarak Pozisyon Kontrolü

Bu uygulamada PDO (Process data object) kullanarak servo motor sürücüsüne uygun verileri yollayıp pozisyon kontrolü yapacağız.

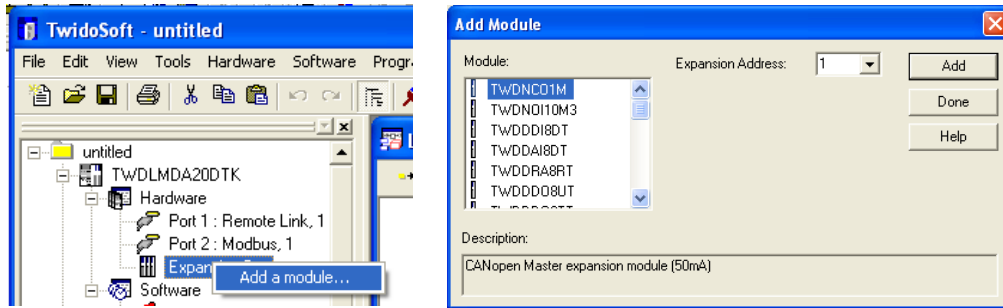
1. İlk olarak twidosoft programını başlatıyoruz.



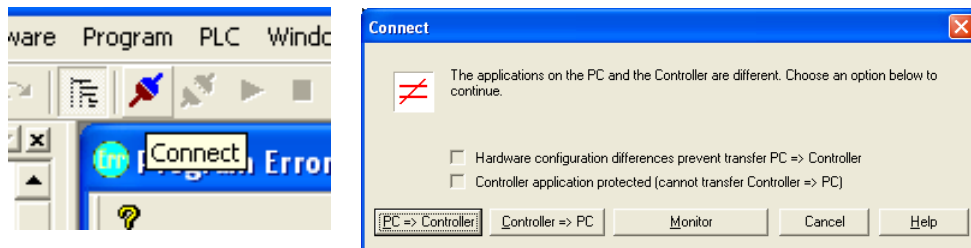
2. PLC' ye bağlanacak donanımın seçimi yapılır. İlk olarak modbus modülü seçilir.



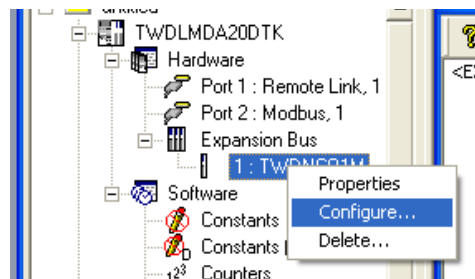
3. Can-open master modülü genişleme adresi 1 olarak seçilir.



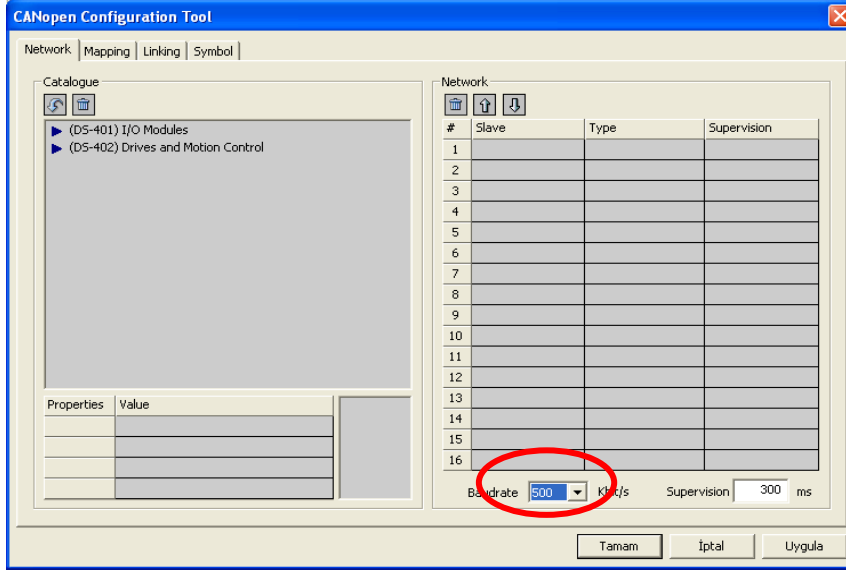
4. Modül ayarlarının doğru olup olmadığının kontrolü için plc'ye bağlanalım.
PC=> Controller aktif ise ayarlar doğrudur. Bağlantıyı tekrar durduralım.



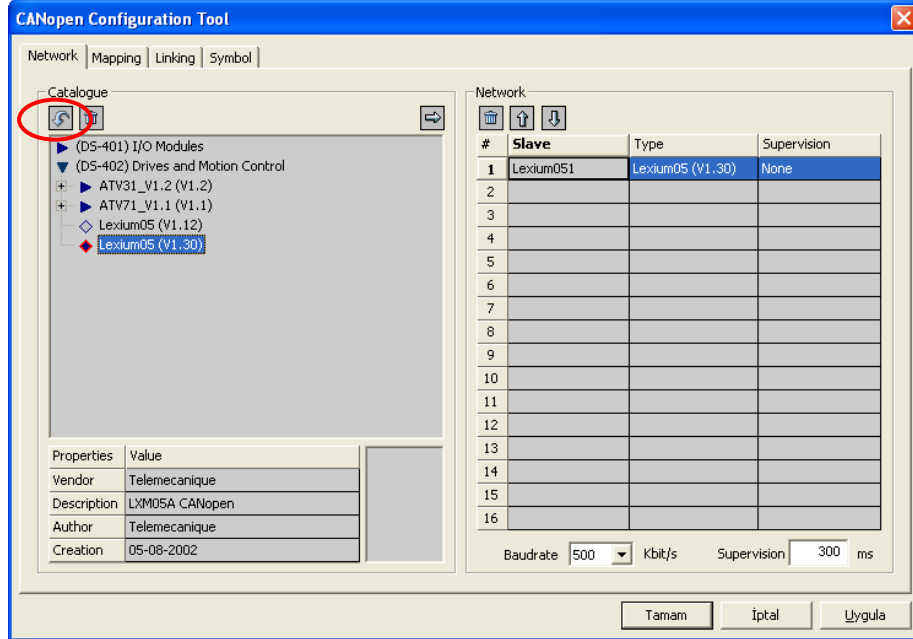
5. Şimdi canopen modülü ayarlarını yapalım.



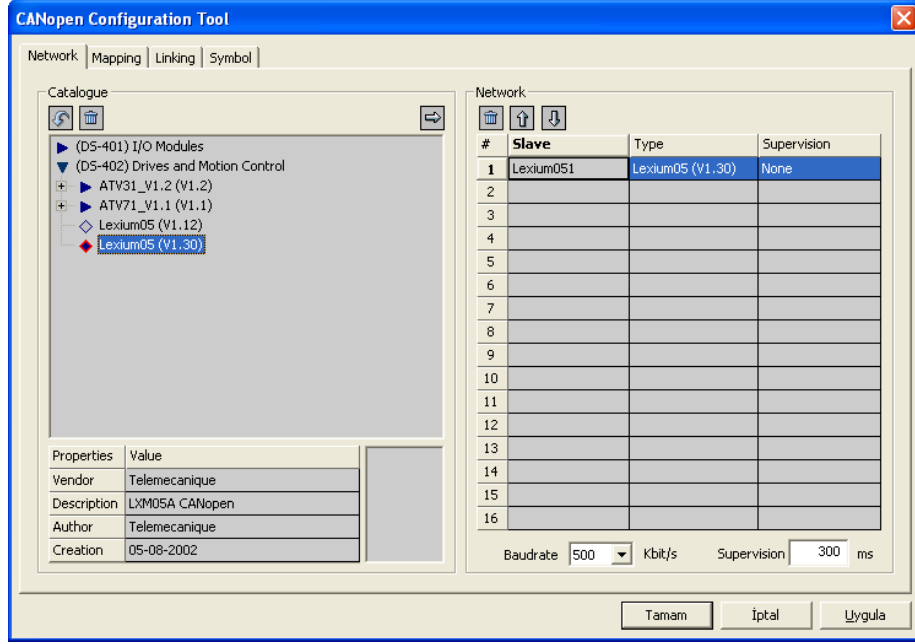
İlk olarak canopen boudrate hız ını 500 kb/sn olarak ayarlayalım.



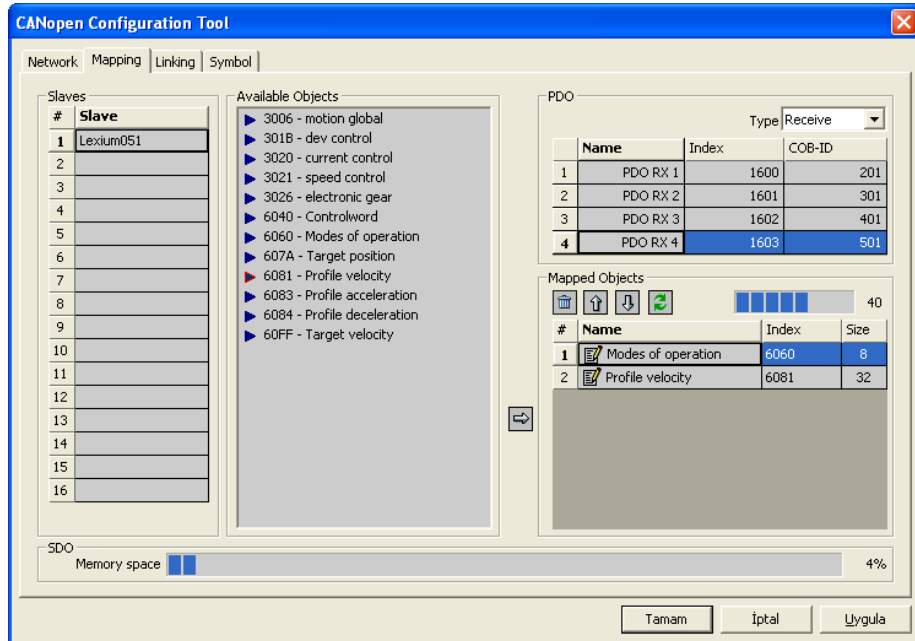
6. Canopen bağlantısında kullanacağımız slave cihaz olan Lexium 5'in EDS dosyası programa yüklenir. Bu programda servo sürücünün sürümü 1.30 olarak seçilir. (Lexium05 versiyonu sizin servo sürücüdeki yüklü versiyon ile aynı olmalıdır. Bu uygulamada v1.30 kullanıldı)



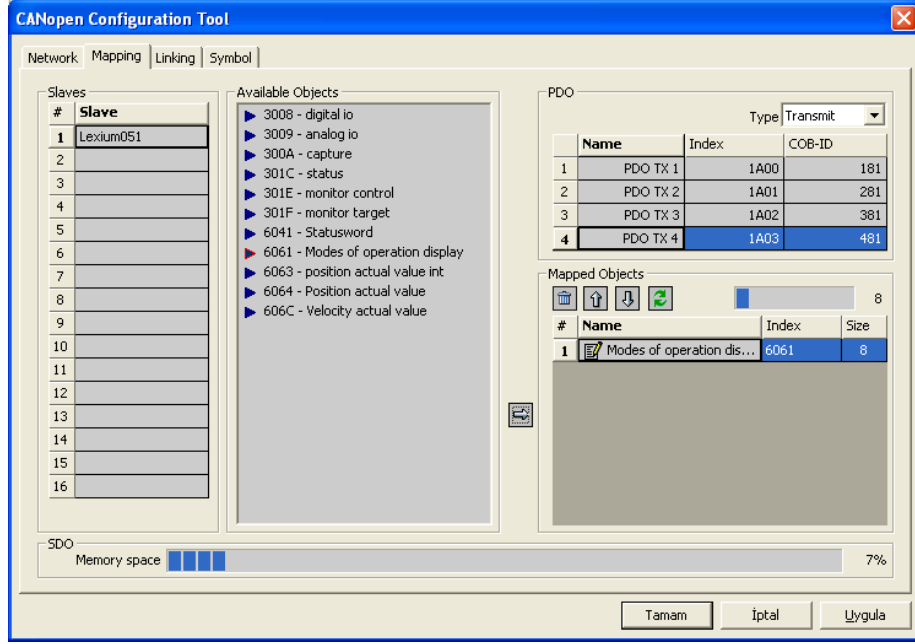
7. Şimdide PDO'ları yani komut ile değil de sadece PLC ile lexium05 arasında kullanılacak ortak data alanlarını belirleyeceğiz. Network kısmından lexium05'i seçiyoruz üzerine çift tıklayarak sağ tarafa aktarıyoruz. Bu bizim Canopen master modülümüz tarafından kontrol edilecek 1'nolu slave modülümüz olacak. (Slave adresini yine servo sürücü üzerindeki panelden ayarlamalısınız.)



8. Şimdiye Profile Position için gerekli PDO'ları seçelim. Pozisyon Kontrolü yapabilmek için Recieve bölümüne yani Lexium5'e "modes of operation" ve "profile velocity" bilgilerini de gönderelim.

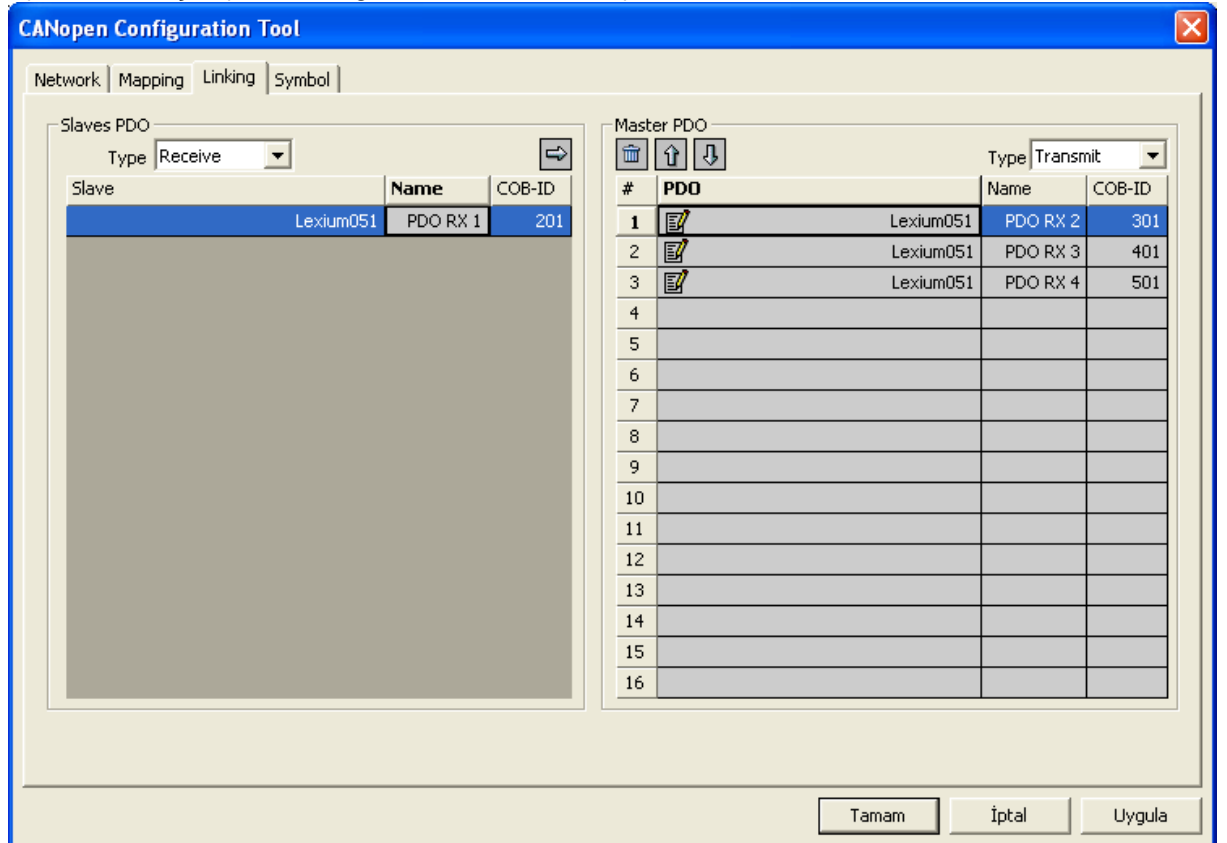


9. Transmit bölümünde ise "Modes of operation display" PDO'su ile hangi çalışma modunda olduğumuzu okutabiliriz. PDO lara ek olarak yalnızca bu iki PDOyu eklemek yeterli olacaktır. Diğerleri zaten ekli durumda gelecektir.

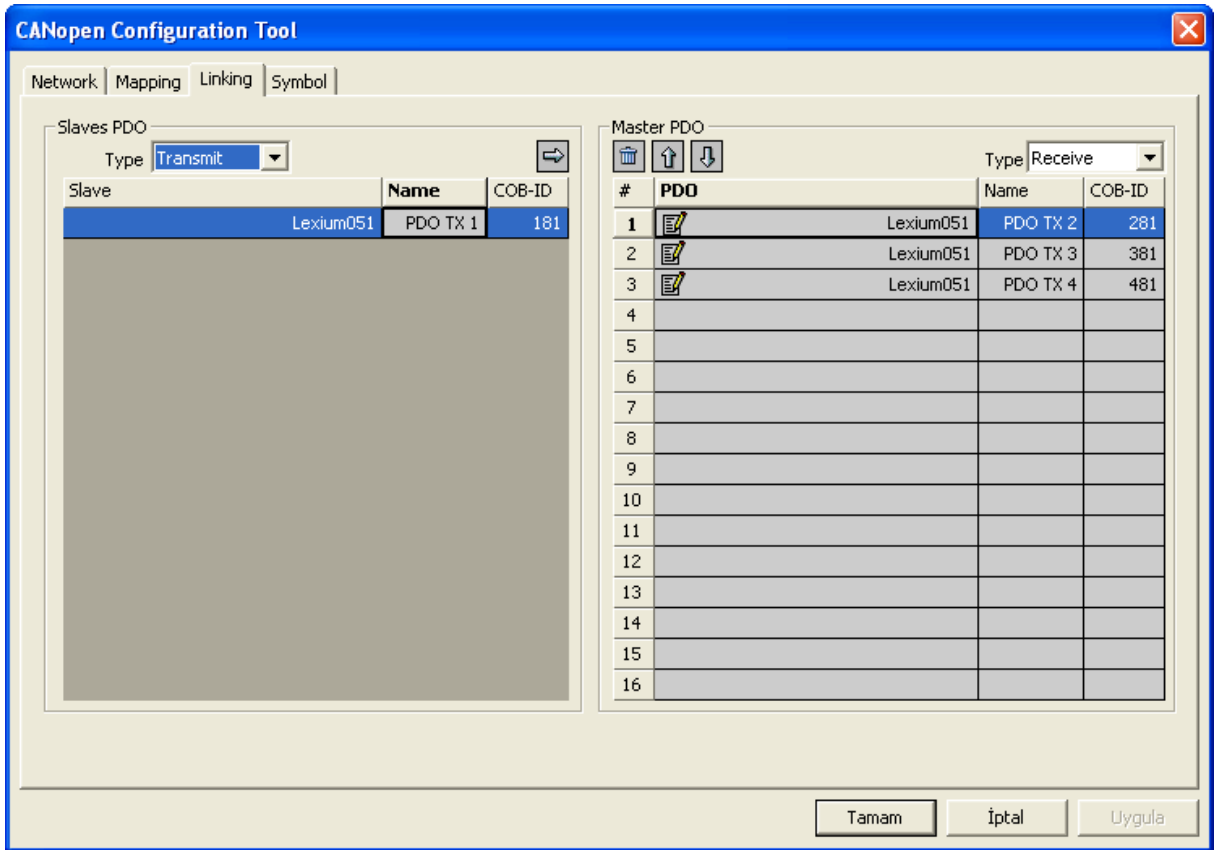


10. Şimdide linking bölümünden hangi alanların alış veriş içerisinde olacağını belirleyelim.

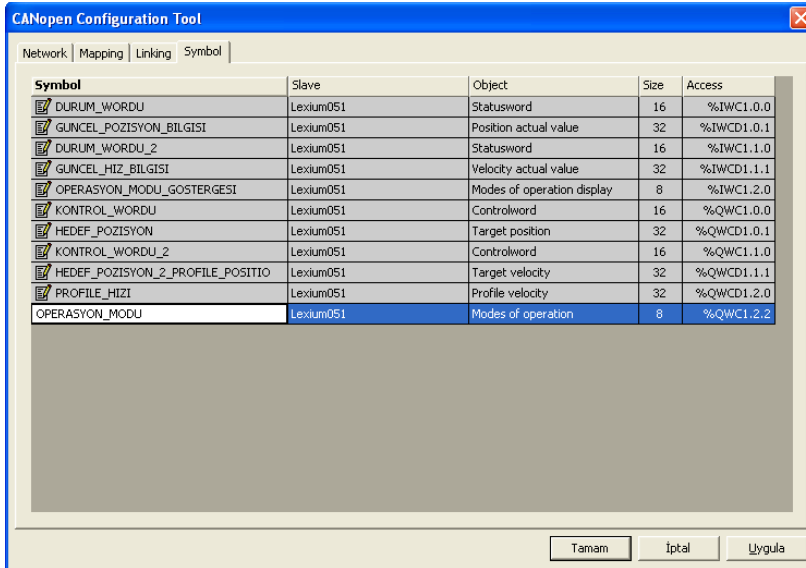
a) Transmit için.(PLC den gönderilecek PDO'lar)



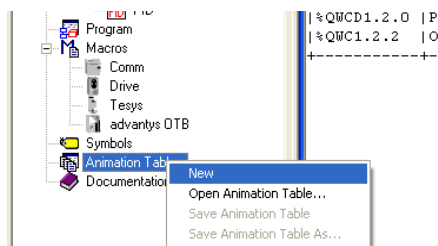
b) Recieve için (PLC ye gelecek PDO'lar)

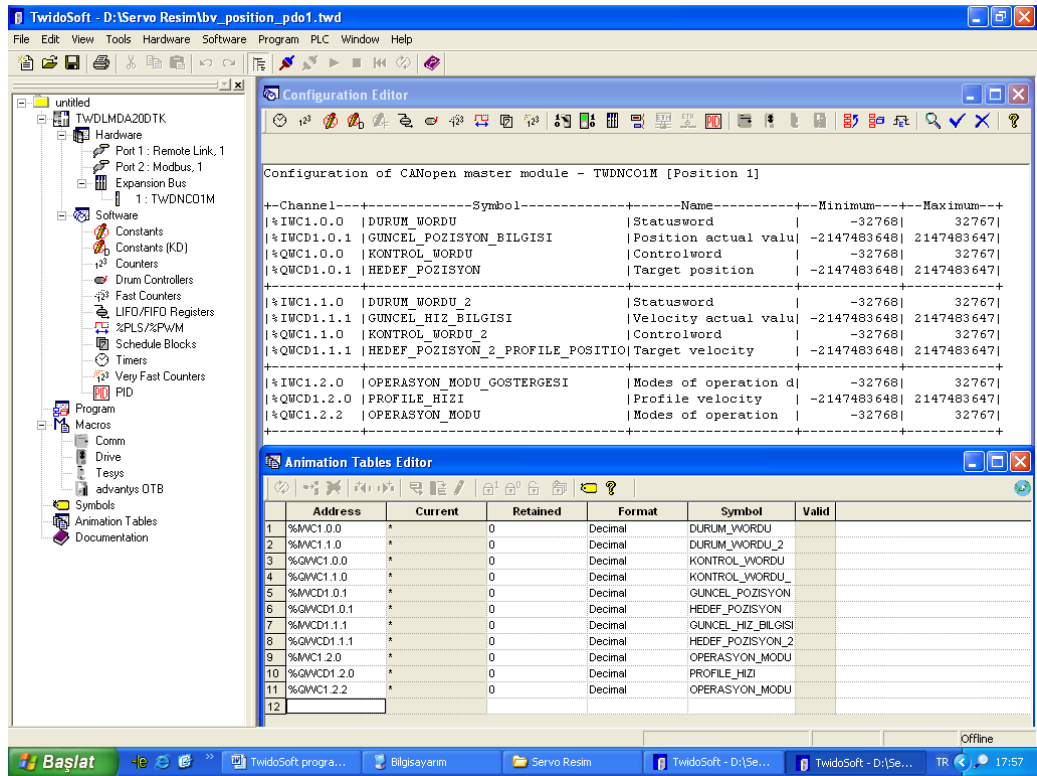


11. Şimdi PDO'ları adlandıralım.

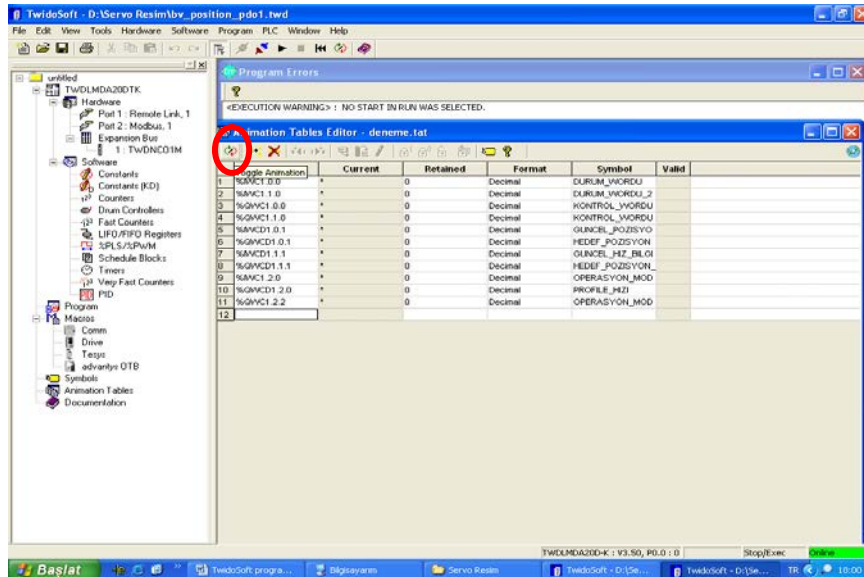


12. Uygula ve tamam dedikten sonra animasyon tablosu oluşturalım.





13. Animasyon tablosunu oluşturduktan sonra bu animasyon tablosunu kapatırken kaydedelim. Şimdi PLC ye bağlanalım. Ve animasyon tablosunu simülasyon moduna getirelim.



14. Kontrol wordunu hegzadecimale çevirip 0000 olduğunu görelim. Şu anda servo boş konumdadır. Yani mil dönebilir.

Animation Tables Editor - Animating - deneme.tat						
	Address	Current	Retained	Format	Symbol	Valid
1	%MVC1.0.0	26161	0	Decimal	DURUM_VWORDU	✓
2	%MVC1.1.0	26161	0	Decimal	DURUM_VWORDU_2	✓
3	%QWVC1.0.0	0000	0000	Hexadecimal	KONTROL_VWORDU	✓
4	%QWVC1.1.0	0	0	Decimal	KONTROL_VWORDU	✓
5	%MWCD1.0.1	-22682	0	Decimal	GUNCEL_POZISYO	✓
6	%QWVCD1.0.1	0	0	Decimal	HEDEF_POZISYON	✓
7	%MWCD1.1.1	0	0	Decimal	GUNCEL_HIZ_BILGI	✓
8	%QWVCD1.1.1	0	0	Decimal	HEDEF_POZISYON	✓
9	%MVC1.2.0	252	0	Decimal	OPERASYON_MOD	✓
10	%QWVCD1.2.0	0	0	Decimal	PROFILE_HIZI	✓
11	%QWVC1.2.2	0	0	Decimal	OPERASYON_MOD	✓
12						

15. Şimdi kontrol word değerine 000f bilgisini çift tıklayarak girelim. Bu durumda servo motorun milini kontrol edersek milin kilitlemiş sabit olduğunu görürüz.

Animation Tables Editor - Animating - deneme.tat						
	Address	Current	Retained	Format	Symbol	Valid
1	%MVC1.0.0	17975	0	Decimal	DURUM_VWORDU	✓
2	%MVC1.1.0	17975	0	Decimal	DURUM_VWORDU_2	✓
3	%QWVC1.0.0	000F	0000	Hexadecimal	KONTROL_VWORDU	✓
4	%QWVC1.1.0	0	0	Decimal	KONTROL_VWORDU	✓
5	%MWCD1.0.1	-22682	0	Decimal	GUNCEL_POZISYO	✓
6	%QWVCD1.0.1	0	0	Decimal	HEDEF_POZISYON	✓
7	%MWCD1.1.1	1	0	Decimal	GUNCEL_HIZ_BILGI	✓
8	%QWVCD1.1.1	0	0	Decimal	HEDEF_POZISYON	✓
9	%MVC1.2.0	252	0	Decimal	OPERASYON_MOD	✓
10	%QWVCD1.2.0	0	0	Decimal	PROFILE_HIZI	✓
11	%QWVC1.2.2	0	0	Decimal	OPERASYON_MOD	✓
12						

16. Şimdide pozisyon ve hız bilgileri girelim.

Animation Tables Editor - Animating - deneme.tat						
	Address	Current	Retained	Format	Symbol	Valid
1	%MVC1.0.0	22071	0	Decimal	DURUM_VWORDU	✓
2	%MVC1.1.0	22071	0	Decimal	DURUM_VWORDU_2	✓
3	%QWVC1.0.0	0000	0000	Hexadecimal	KONTROL_VWORDU	✓
4	%QWVC1.1.0	001F	0000	Hexadecimal	KONTROL_VWORDU	✓
5	%MWCD1.0.1	19999	0	Decimal	GUNCEL_POZISYO	✓
6	%QWVCD1.0.1	20000	0	Decimal	HEDEF_POZISYON	✓
7	%MWCD1.1.1	-1	0	Decimal	GUNCEL_HIZ_BILGI	✓
8	%QWVCD1.1.1	0	0	Decimal	HEDEF_POZISYON	✓
9	%MVC1.2.0	1	0	Decimal	OPERASYON_MOD	✓
10	%QWVCD1.2.0	100	0	Decimal	PROFILE_HIZI	✓
11	%QWVC1.2.2	0	0	Decimal	OPERASYON_MOD	✓
12						

Parametre Değerleri

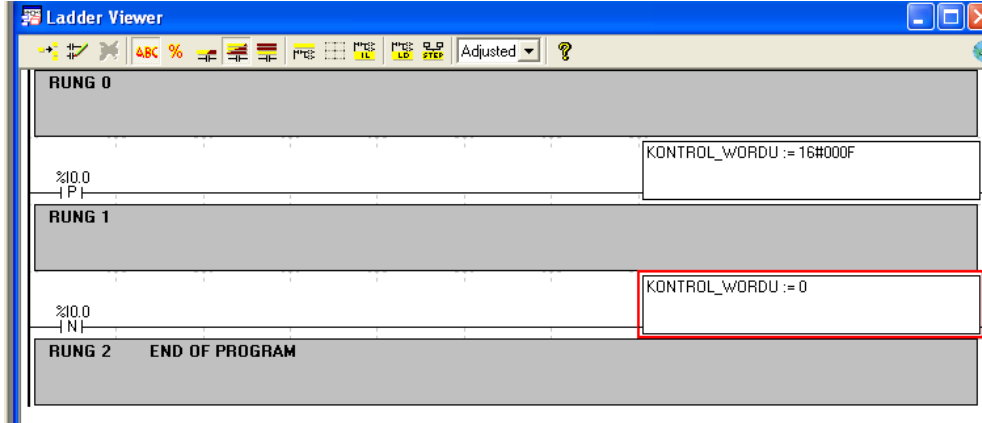
Hedef_Pozisyon=20000

Profile_hizi=100

Operasyon_modu=1

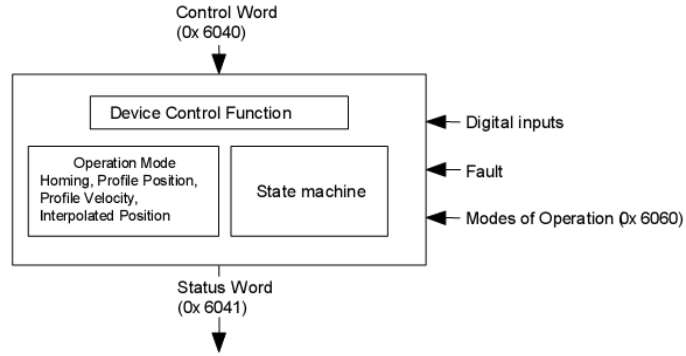
İlk olarak kontrol wordu "000f" yapılacak. Ardından tekrar "001f" yapılacak. Motorun döndüğü görülecektir.

Ladder Diyagramında PDO Kullanma



Control Word ve Status word ları:

Girilen kontrol word komutlarına göre status (durum) wordu değişim yapısı aşağıda verilmiştir.



Control ve status word hakkında daha ayrıntılı bilgiler aşağıda verilmiştir.

Control Word (0x6040)

Kontrol wordu 16 bitlik bir veridir. İçeriği Aşağıdaki gibidir.

0. Bit	Aç. Bu bit Amplifier (sürücünün) aktif olması için 1 olmalıdır.
1. Bit	Gerilim aç. Bu bit Amplifier (sürücünün) aktif olması için 1 olmalıdır.
2. Bit	Hızlı Dur. Bu bit 0 ise sürücü quickstop komutunu icra eder.
3. Bit	Operation enable. Bu bit Amplifier (sürücünün) aktif olması için 1 olmalıdır.
4-6 bitler	Operation mode specific. Farklı operasyon modları için kullanılır.
7. Bit	Reset Fault. 0 dan 1 e geçişte sürücüde kalan tüm hataları resetler.
8. Bit	Halt. Bu bit 1 iken sürücü halt konumuna geçer.
9-15 Bitler	İleride Kullanım amacı ile ayrılmıştır. Varsayılan değer 0 dır.

Örneğimizde Kontrol_wordu girişin %I0.0 yükselen kenarda iken "000F" yaparak sürücünün aktif duruma geçmesini sağladık.

	0				0				0				F			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Motorun serbest duruma geçmesi için ise Kontrol_wordu girişin %I0.0 düşen kenarda iken “0000” yaparak sürücünün serbest konuma geçmesini sağladık.

	0				0				0				0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Status Word (0x6041)

Durum wordu da status word gibi 16 bitlik bir bilgidir. Sürücünün durumu hakkında bilgi verir. İçeriği Aşağıdaki tabloda verilmiştir.

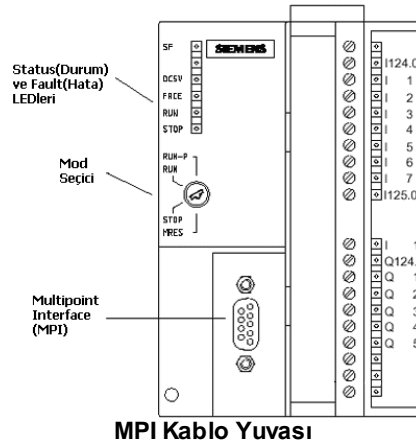
0. Bit	Başlamak için hazır.					
1. Bit	Başlamış.					
2. Bit	Operation Enabled. 1 ise operasyon açık.					
3. Bit	Hata. Eğer 1 ise sürücüde hata var.					
4. Bit	Voltage Enabled. 1 ise operasyon için gerekli minimum voltaja sahip.					
5. Bit	Quickstop (Hızlı Dur). Eğer 0 ise sürücü hızlı dur uyguladı.					
6. Bit	Açma aktif durumda (açık).					
7. Bit	Uyarı. Eğer 1 ise uyarı var. Uyarı ile ilgili bilgiler Manufacturer Status Register den okunabilir.					
8. Bit	Eğer 1 ise bir önceki operasyon zamanından önce durdurulmuş.					
9. Bit	Uzak Kontrol. 1 iken Sürücü Canopen ile kontrol edilmektedir.					
10. Bit	Hedefe Ulaşıldı. Hedef pozisyona ulaşıldığında 1 .					
11. Bit	İç Limitler aktif. 1 ise Sürücülerden herhangi birinin iç limitleri (Akım, Gerilim, Hız veya konum) aktif.					
12-13 Bitler	Operasyon mod belirteci.					
	Bit	Profile Position Mode	Profile Velocity Mode	Profile Torque Mode	Homing Mode	Interpolated Position Mode
	12	Setpoint acknowledge.	Speed = 0.	Reserved	Homing attained.	Interpolated pos. mode active.
	13	Following error.	Maximum slippage error.	Reserved.	Homing error.	Reserved.
14. Bit	1 İken sürücü hareket operasyonu var 0iken ise operasyon tamamlandı. Bu bit hareketin bittiği anda sıfırlanı ancak motorun pozisyonunda olduğu anlamına gelmez.					
15. Bit	Reserved					

MPI HABERLEŞME SİSTEMİ

MPI (Multipoint Interface) haberleşme sistemi özellikle işlemciler arası haberleşme işlemlerinde çok yoğun olarak kullanılır. Konfigürasyon ve kullanımı oldukça basittir. İki damarlı (profibus) kablosu bir kablo ve MPI bağlantı bağlayıcısı(konnektör) dışında bir donanıma ihtiyaç duymazlar.

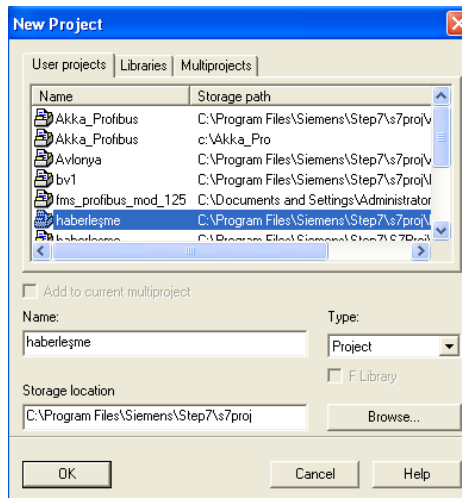
Haberleşme kablosu (profibus kablosu) MPI hattına, programlama cihazı bağlantı kablosu (MPI kablosu) bağlanıyormuş gibi bağlanmalıdır.

Maksimum 32 adet katılımcı bağlanabilir ve iletim hattı uzunluğu en fazla 50 metre olabilir, 50 metrenin üzerindeki mesafeler için RS 485 yükseltici kullanmak gerekir. Her yükseltici hat uzunluğu 1000m kadar çıkarabilir. Toplam 10 yükseltici kullanılabilir. İletim hattının başlangıç ve bitiş noktalarındaki bağlayıcılara sonlama direnci konmalıdır. (konnektör “on” konumuna alınmalıdır)

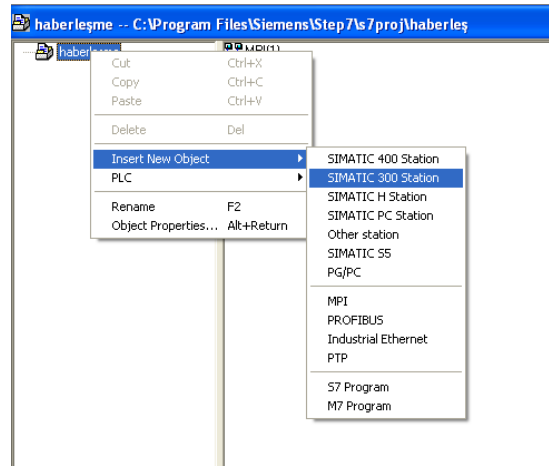


MPI Bağlantısı

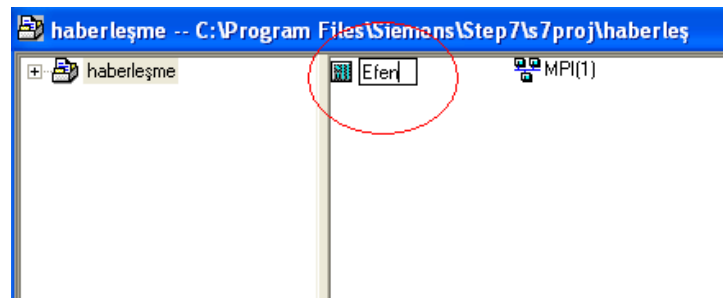
1. ADIM: Yeni bir dosya açalım.



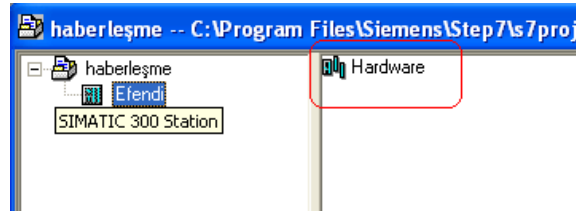
2. ADIM: İstasyon ekleyelim.



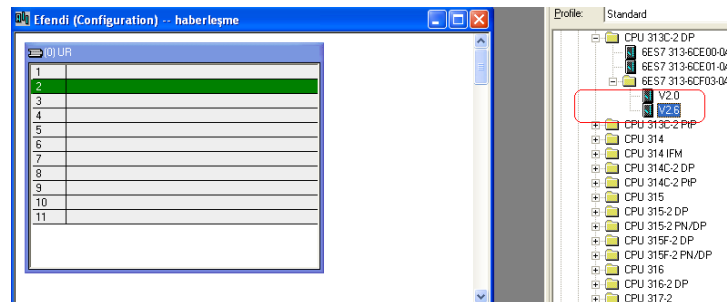
3. ADIM: Efendi ismini verelim.



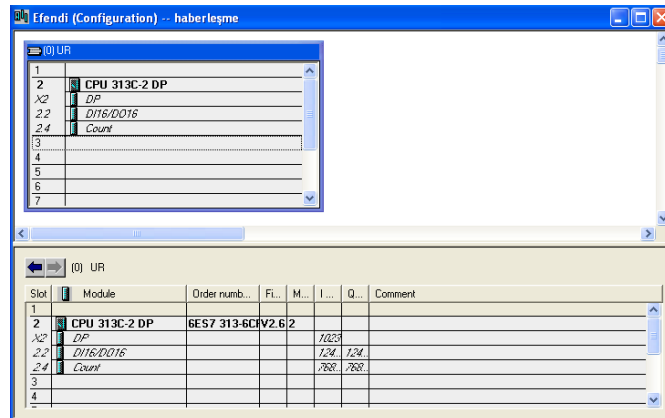
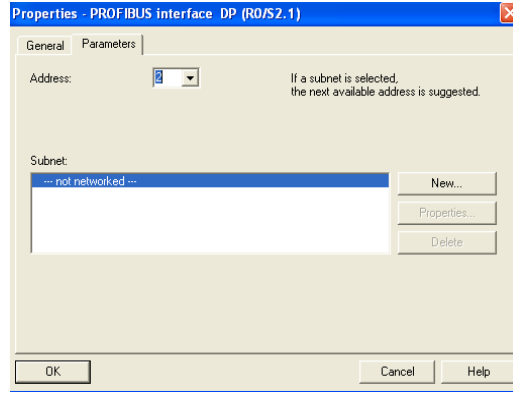
4. ADIM: Hardware üzerinde çift tıklayalım.



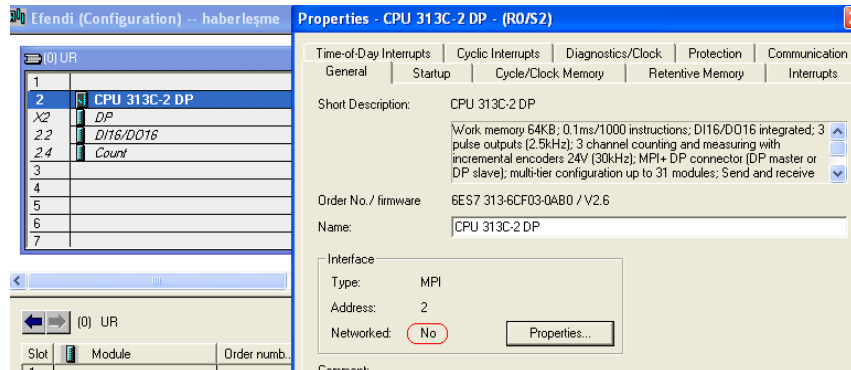
5. ADIM: Donanımı tanıtalım.



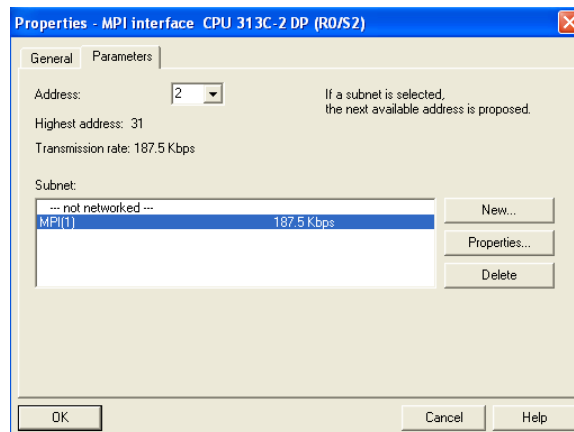
Bu esnada ekrana gelen ekranda OK diyelim.

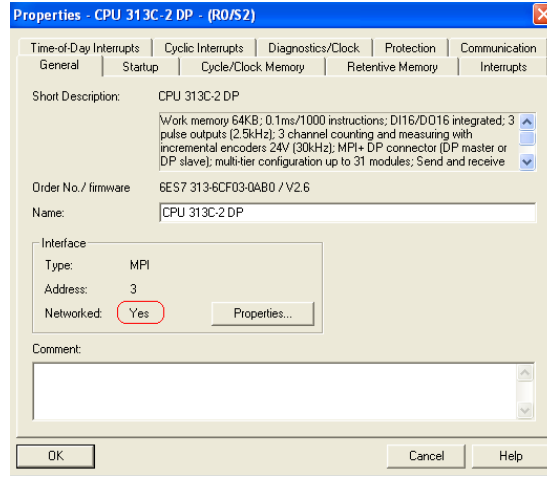


6. ADIM: Donanım ayarları görünmektedir. CPU satırında çift tıklayalım.

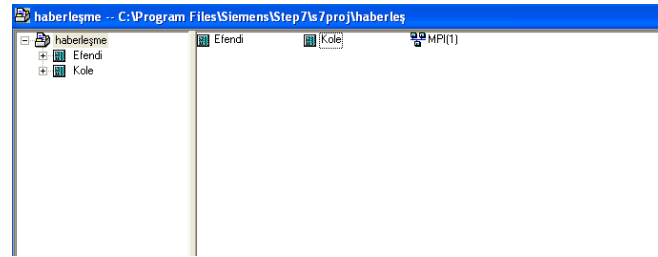


7. ADIM: Properties düğmesine tıklayalım.

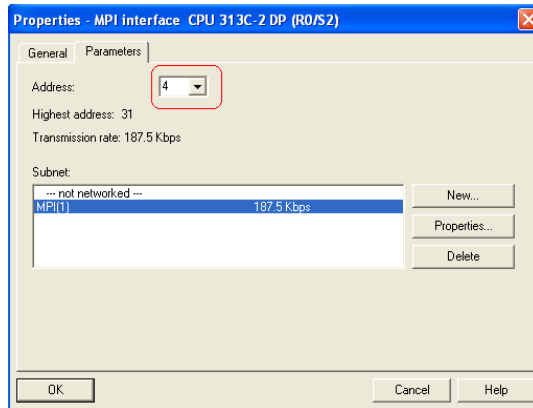




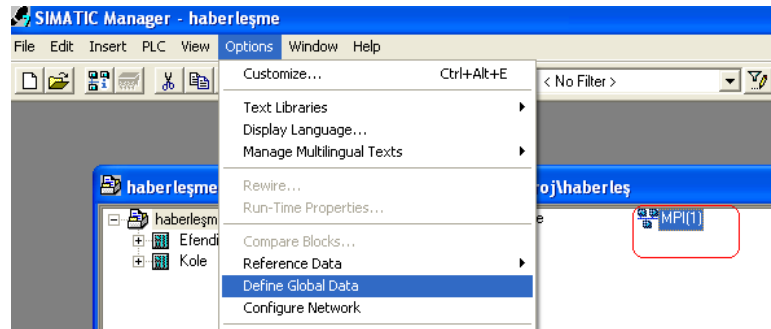
8. ADIM: Aynı şekilde Kole isimli başka bir istasyon kuralım.



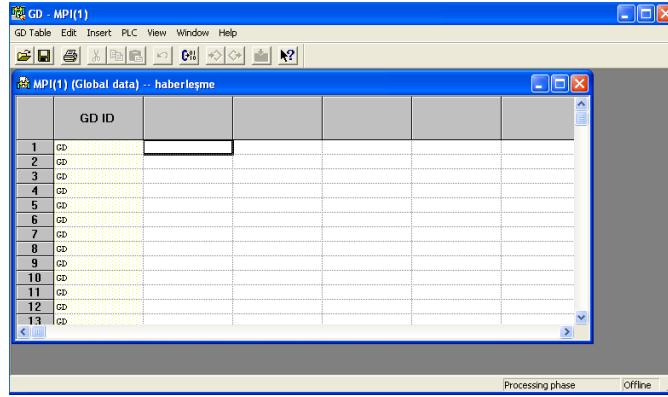
9. ADIM: Donanımı tanıtalım.



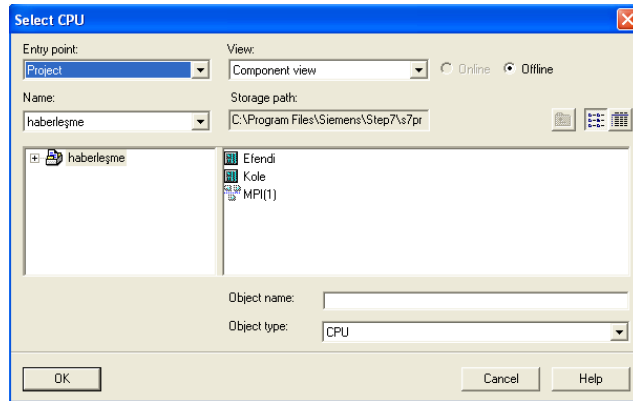
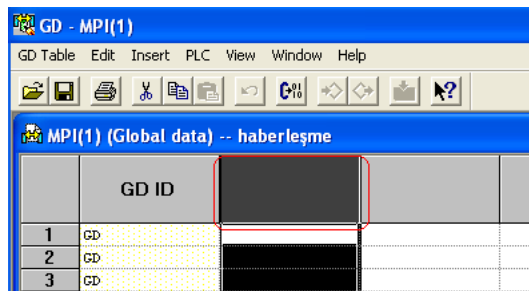
10. ADIM: Proje alanına geçelim. MPI işaretli iken Options/Define Global Data'ya girelim.



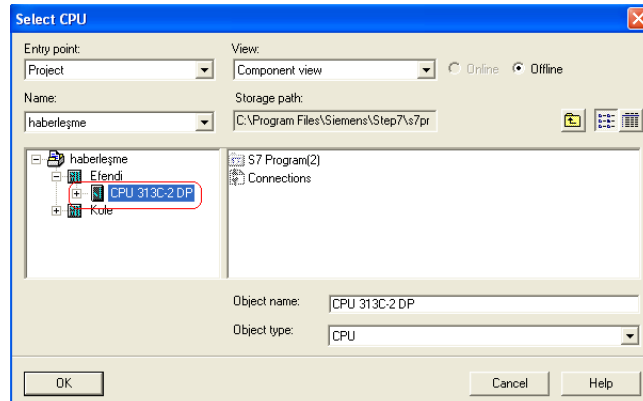
11. ADIM: GD penceresi açılır.

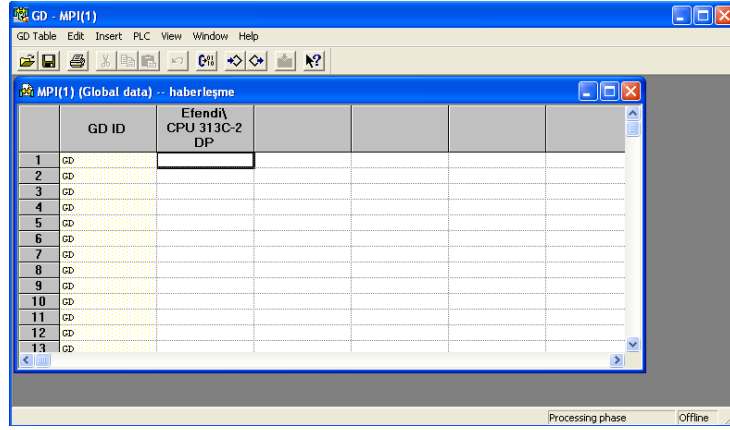


12. ADIM: Şekildeki ekranda çift tıklayalım.

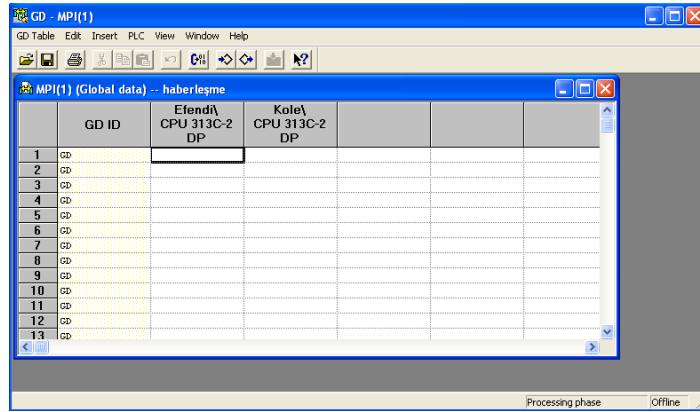


13. ADIM: Efendi işaretlenir ve onay verilir.





14. ADIM: Aynı şekilde Kole de getirilir.

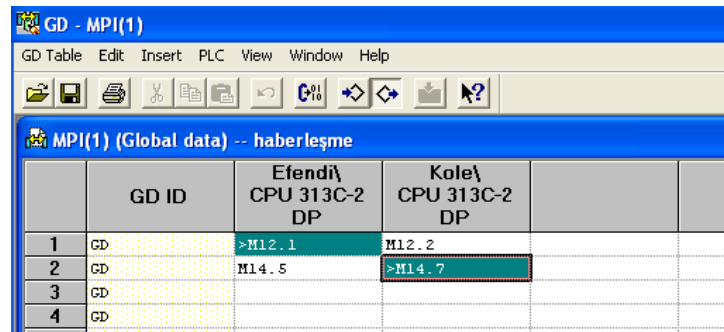


15. ADIM: Tabloyu dolduralım.

MPI(1) (Global data) -- haberleşme					
	GD ID	Efendi\ CPU 313C-2 DP	Kole\ CPU 313C-2 DP		
1	GD	M12 . 1	M12 . 2		
2	GD	M14 . 5	M14 . 7		
3	GD				
4	GD				
5	GD				

16. ADIM: Her satırda sadece bir tane Sender (gönderici) olabilir. Diğerleri alıcı olabilir. İlk adres alanında sağ tuşla tıklanır ve

Aynı şekilde ikinci satır için yapılır.

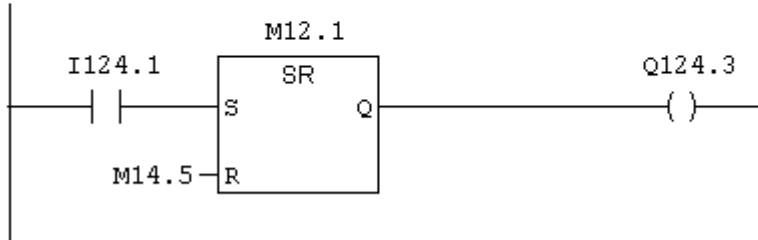


17. ADIM: Kaydedelim.

18. ADIM: Efendi için programı yazalım.

Network 1: Title:

Comment:



19. ADIM: Kole için program:

Network 1: Title:

Comment:



Network 2: Title:

Comment:



20. ADIM: Programları PLClere irsal ederek çalıştırın.

USB

UNIVERSAL -BUS (Evrensel Seri Veriyolu)

Plug and Play aygıtların sistemlerde daha kolay bir şekilde kullanımını sağlamak amacıyla geliştirilmiş yeni bir yapı olan USB (Evrensel Seri Veriyolu), harici bağlantılar için standart ve hızlı bir arabirim sunmaktadır. Aralarında IBM, Intel ve Digital'inde bulunduğu yedi firmanın oluşturduğu bir grup tarafından tasarlanan USB, özellikle bilgisayar ile telekomünikasyon alanındaki gelişmeler karşısında doğan ihtiyaçların karşılanmasını amaçlamaktadır. USB sayesinde sisteme bağlanan aygıtlar herhangi bir ayar gerektirmeden çalıştırılabilmektedir. 127 adede kadar birçok farklı aygıtın aynı sistemde kullanımına izin veren bu yapıda bilgisayar parçaları dışındaki ürünlerde desteklenmektedir (Video, müzik, haberleşme ürünleri gibi).

USB Yapı Avantajları

Öncelikle USB standart bir yapıdır. Aygıt bağlantılarında 4 pinli, özel plastik kaplamalı, yüksek kalitede kablo kullanılmaktadır. USB yapısını kullanan bütün aygıtlar için bu kablo yapısı standarttır. Bu da USB yapısının en önemli özelliklerinden biri olan uyumluluğu sağlamaktadır. Bu yapıdaki her türlü aygıt USB desteği sunan her sistemde herhangi bir uyumsuzluk sorunu yaşanmadan kolayca kullanılabilir.

USB yapısını kullanan birimler daha basit dizaynlara sahip olduğundan kurulum ve kullanım bakımından kolaylık sağlanmıştır. Standart arabirimi sayesinde işletim sistemleri tarafından da desteklenen USB sayesinde, ürünlerin yazılım-donanım test gereksinimi minimum düzeye indirilmiştir.

En önemli özelliği olan tam Pnp desteği ile bağlanan aygıtlar hiçbir ayar gereksinimi olmadan ve sistemin kapatılmasına gerek kalmadan kullanılabilir. 127 aygıt kadar sunulan birim desteği aynı sistemde birçok farklı ünitenin çoklu kullanımını mümkün kılmaktadır.

USB yapısında data 16 bit'lik paket formatında aktarılmaktadır. Veri transferi 1.5 Mbit/sec oranında düşük-hız seviyesinde veya 12Mbit/sec oranında yüksek-hız seviyesinde gerçekleşebilmektedir. Anakart üzerindeki chipset tarafından denetlenen USB portundan aktarılan verinin kontrolü ile birimler arasındaki iletişim daha güvenli bir şekilde yapılabilir.

USB Kullanım Alanları

USB standardı 250'yi aşkın bilgisayar ve iletişim firması tarafından desteklenmektedir. Klavyeden monitöre kadar birçok ürünün USB yapıdaki modeli kullanılmaktadır. İletişim alanında da kullanım için uygun yapıda olan USB sayesinde dijital PBX santrallerin ve network bağlantılarının kullanımı mümkündür. Video ve müzik ürünleri de aynı arabirim ile Pc sistemine bağlanabilmektedir. Standart yapısı sayesinde kolay bir şekilde çok sayıda ürün bağlantısına olanak sağlayan USB, henüz yaygın bir kullanım alanına sahip değildir. Daha fazla hız ve ürün bağlantısı için talep arttıkça bu yapının kullanımı da yaygınlaşıp standart bir konuma gelecektir.

USB Veri Yolu ve Getirdiği Yenilikler

USB (Universal -Bus = Evrensel Seri Veriyolu) 1995'te ortaya çıkmıştır. USB bağlantı standardı sayesinde farklı tipte konnektörlere, DMA kanal değişikliklerine gerek duyulmayacak, IRQ çakışmaları ortaya çıkmayacak, jumperlara gerek olmayacak; bir tek PC'ye 127 adede kadar cihaz bağlanabilecektir. USB, bilinen birçok PC konnektörünün (Centronics paralel, RS-232 seri, Mini-DIN ve Sub-D (Oyun portları, printer portları, klavye ve mouse konnektörleri, modem ve birçok network adaptörü)

vb.) yerini alacaktır. -uç olarak USB, ana güç, yüksek hızlı video ve çok yüksek hızlı networkler dışında birçok harici bağlantının yerini alacaktır.

USB veriyolunu kullanan cihazlar direk olarak PC'lere bağlanabilecekleri gibi, USB hubları aracılığı ile de bağlantı kurabilirler. USB, star (yıldız) topolojiyi kullanır (Şekil 1). Bu topolojide cihazlar bağlandıkları bilgisayardan veya USB hublarından en fazla 5 m. uzaklıkta olabilir. USB, iki adet veri transfer hızını desteklemektedir: 1,5 Mbps ve 12 Mbps; bu band genişlikleri PC kamera, monitör, modem, printer, scanner, mouse, joystick, klavye, ses kartları ve ses sistemleri, telefon, network cihazları ve daha birçok cihazın çalışabilmesi için yeterlidir. USB, senkron (eş zamanlı) ve asenkron veri transferini desteklemektedir. Bütün USB cihazlar tek tip konektör kullanmaktadır.

USB, aralarında Intel, Microsoft, Compaq, IBM gibi firmaların da bulunduğu 400 'ün üzerinde kuruluş tarafından desteklenmektedir. -zamanlarda üretilen hemen bütün sistemlerde ve anakartlarda USB desteği bulunmaktadır.

USB'nin Getirdiği Avantajlar:

Tek bir PC'ye 127 adede kadar cihaz bağlayabilme

Hiçbir sürücüye, IRQ ayarlarına, DMA kanallarına ve I/O adreslerine, genişleme yuvalarına gerek duymadan kolay kurulum

Çevresel cihazlar için PC'yi kapatıp açmadan Tak ve Çalıştır fonksiyonelliği

Bütün cihazlar için tek tip konektör

PC'yi kapatmadan cihaz ekleme ve kaldırma özelliği

Bilgisayar kullanıcıları yıllardır bilgisayarlarına bir yan donanım alırken bağlantı konusunda bocaladılar. Paralel ve seri portlar yavaş ve konfigürasyon problemi içerir, SCSI için fazladan kart almanız gerekir ve sürücü problemleri her zaman olasıdır. Bu durum artık bıkkınlık safhasına geldiği anda Evrensel seri yol, (Universal -Bus, USB) bu ihtiyaca cevap şekilde dünyaya geldi. Küçüklüğü ve kolay kullanımı ile bir anda donanım dünyasında parlayan bu arabirimi günümüzde kullanmayan yan ürün neredeyse yok gibi. USB ile IRQ ya da DIP anahtarlar ile saç baş yolmadan yavaş ya da hızlı her türlü veri yollanabiliyor.

Günümüzde USB kullanımı -derece yaygınlaşmış ve pek çok ürün dalında yüzlerce şirket USB bilgisayar ürünleri üretmektedir. Bu ürünler arasında Fare, tuş takımı, Joystick, telefon, kamera, modem, CD-ROM sürücü, teyp ve floppy sürücüleri radyolar, ses kartları ve ses yongalı hoparlörler, ISDN modemler ve ağ ürünleri sayılabilir. Şekilde bir USB-ISDN modem görülmektedir.

USB, bütün verinin bir çift kablo üzerinde dengeli bir şekilde iletilişi, gerektiğinde başka bir çift kablo ile güç aktarımının yapılabilirdiği seri iletişim protokolü ve fiziksel bağlantıya verilen addır. USB 1.0 standartları iki farklı kablo ve konektör tanımlamıştır. 12Mbps'lik hızlı USB bağlantısında kullanılan kablolar iyi korunmuştur (İletim yapan kabloların etrafı hasır şeklinde ince kablo ağı ile kaplanır, bu işaretlerin elektriksel gürültüden arınmasını sağlar.). 1.5Mbps'lik yavaş USB ise daha ucuz ve nispeten korumasız bir kablo kullanır. Bütün kabloların bir ucu 'A', diğer ucuda 'B' olarak adlandırılır. USB hattı üzerinden bilgi telefon halarında olduğu gibi iki kablo üzerinden gönderilir (dengeli iletişim) ve NRZI adı verilen bir kanal kodlaması uygulanır (Bu kısımlar detay isteyen arkadaşlara bilgi temin etmeye çalışırım).

Kablo uzunlukları da 12Mbps için 5m ve 1.5Mbps için 3m olmak üzere sınırlandırılmıştır. USB topolojisine (cihazların bağlanma yapısı) sıralı yıldız (tiered star, üstteki şekil) yapısı adı verilir. Yapıda görülen HUB'lar bağlantı sayısının artırılmasını sağlarlar ve sistemde sadece bir tane ana sistem denetleyicisi ya da "kök hub" (Yani bilgisayarımız) bulunur. Sistem üzerindeki bütün USB yan birimleri bilgisayar tarafından yönetilir (Elektronik ile ilgilenenlerin tanıdığı terim ile Efendi-Köle, Master-Slave yapısı). Bugün bilgisayar üzerindeki USB denetleyici birim genellikle yongasetleri içerisinde yer almaktadır. Bir cihaz USB ağına bağlandığı anda ana bilgisayar cihazın kimliğini ve hangi donanım sürücüsünün kullanılacağı belirlenir (bu işleme 'enumeration' adı veriliyor). Enumeration işlemi sırasında PC, yan birime bir adet cihaz adres numarası verir.

USB standardı iki tür yan birim cihazı tanımlar. Tekil bileşenler tek başlarına bir fonksiyonu yerine getirirler. Bu tip cihazlara örnek olarak USB fare ya da Ses kartını örnek gösterebiliriz. Bileşik USB cihazları ise bir USB arabirimini paylaşan -en fazla yan donanım içerirler. Örneğin Ses ve görüntü için aynı USB bağlantısını kullanan bir video kamera. Hub cihazları ise köprülerdir, kısıtlı USB bağlantısının artırılması için kullanılırlar. Hub'lar da sadece bir tane yukarı-akış (upstream, yani bilgisayara doğru veri akışı) bağlantısı, -en fazla aşağı-akış (downstream, yan birimlere doğru veri akışı) yer alır. Hublar da aslında USB cihazlarıdır ve yetenekli özellikle taşıyabilirler. Günümüzdeki tipik uygulamalardan birisi Monitörün alt kısmına bir USB Hub yerleştirmektir. Bu yolla bilgisayar arkasındaki kablo karışması da bir nebze azalmış olur.

USB'nin iyi yönlerinden birisi de USB cihazlarının sisteme her an takılıp çıkarılabilmeleridir. Enerjiyi kesip yeniden vermek gerekmemektedir. Hub'lar otomatik olarak USB ağına meydana gelen değişimleri sezerler. Hub enerjisini Bağlı olduğu Üst Hub'dan (Bu bilgisayar ya da başka bir Hub olabilir) ya da varsa kendi içerisindeki güç kaynağından temin eder. Eğer bir sisteme pek çok USB birimi bağlamayı düşünüyorsanız üzerinde enerji kaynağı olan bir HUB seçmenizi öneririm. Hub kendisine yeni bir cihaz bağlandığında ya da çıkarıldığında bunu hemen tespit edip üst cihaza rapor verir. Yeni cihazla gerekli iletişim bağlantısını kurar.