

**T.C.
MUĞLA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

**İNTERNET TABANLI UZAKTAN DENETİM İÇİN BİR ARAYÜZ
TASARIMI**

YÜKSEK LİSANS TEZİ

MUAMMER KADİR KARAGÖZ

**NİSAN 2008
MUĞLA**

**T.C.
MUĞLA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

**İNTERNET TABANLI UZAKTAN DENETİM İÇİN BİR ARAYÜZ
TASARIMI**

YÜKSEK LİSANS TEZİ

Muammer Kadir KARAGÖZ

MUĞLA 2008

T.C.
MUĞLA ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yrd.Doç.Dr. Aydın ÇETİN danışmanlığında Muammer Kadir KARAGÖZ tarafından hazırlanan İNTERNET TABANLI UZAKTAN DENETİM İÇİN BİR ARAYÜZ TASARIMI başlıklı tez, 25/04/2008 tarihinde aşağıdaki jüri tarafından Elektronik Bilgisayar Eğitimi Anabilim Dalı'nda yüksek lisans tezi olarak oybirliği ile kabul edilmiştir.

Başkan : Yrd. Doç. Dr. Mahmut TENRUH

İmza :



Üye : Yrd. Doç. Dr. Aydın ÇETİN

İmza :



Üye : Yrd. Doç. Dr. Ayhan İSTANBULLU

İmza :



ÖNSÖZ

Uzaktan denetim sistemleri, özellikle endüstriyel cihaz üretimi yapan firmalar için uzaktan hızlı bakım hizmeti ile satış sonrası müşteri memnuniyetini artırmak, maliyetleri azaltmak ve daha üstün özelliklere sahip yeni makineler üretebilmek gibi avantajlar sunmaktadır.

Bu tezde, endüstriyel cihazların İnternet'e bağlı herhangi bir bilgisayar ya da mobil cihaz yardımıyla doğrudan denetlenebilmesi ve izlenebilmesi amacıyla, mevcut uzaktan denetim sistemlerine oranla daha üstün ve esnek bir arayüz sistemi tasarımı hedeflenmiştir.

Uzaktan erişim arayüzü, mesleki ve teknik eğitim alanında cihazların uzaktan denetimine ilişkin eğitim uygulamaları yapılabilmesine imkân sağlamaktadır. Dolayısıyla, uzaktan erişim ile uzaktan uygulamalı eğitimin gerçekleştirilmesine ve eğitim kalitesinin artırılmasına ve uygulamalı eğitim maliyetlerinin düşürülmesine katkı sağlamak tezin hedefleri arasında yer almaktadır.

Bu tez çalışması süresince zorluklarla karşılaştığımda beni cesaretlendiren, destekleyen, yönlendiren ve yardımlarını esirgemeyen danışmanım Yrd. Doç. Dr. Aydın ÇETİN'e, değerli öneri ve eleştirilerinden dolayı Yrd. Doç. Dr. Mahmut TENRUH ve Yrd. Doç. Dr. Ayhan İSTANBULLU'ya, zor zamanlarımda anlayış gösteren ve koşulsuz destek veren aileme teşekkür ederim.

Muammer Kadir KARAGÖZ

MUĞLA 2008

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	I
İÇİNDEKİLER.....	II
ÖZET.....	III
ABSTRACT	IV
ŞEKİLLER DİZİNİ	V
ÇİZELGELER DİZİNİ	VI
SEMBOLLER ve KISALTMALAR DİZİNİ.....	VII
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ	3
3. JAVA AĞ PROGRAMCILIĞI	6
3.1 Java Programlama Dili	6
3.2 İnternet İletişim Kuralları	7
3.2.1 Temel TCP/IP protokolleri.....	8
3.2.1.1 İnternet protokolü	8
3.2.1.2 İletim denetimi protokolü.....	9
3.2.1.3 Kullanıcı datagram protokolü.....	10
3.2.1.4 Haberleşme bağlantı noktaları	10
3.3 Soket Programcılığı.....	11
3.3.1 TCP soket programcılığı	11
3.3.2 UDP soket programcılığı.....	13
3.4 Java Applet.....	14
4. UZAKTAN DENETİM ARAYÜZÜ TASARIM VE UYGULAMASI.....	16
4.1 Donanım	16
4.1.1 Ethernet Modülü	17
4.1.2 Mikrodenetleyici.....	22
4.1.3 RS485 Alıcı-Verici	29
4.1.4 Asenkron Motor.....	31
4.1.5 Motor Sürücü.....	32
4.1.5.1 Motor sürücü kurulumu	33
4.1.5.2 Seri arayüz	34
4.2 Yazılım	36
4.2.1 Mikrodenetleyici Yazılımı	36
4.2.1.1 Evrensel seri arayüz protokolü	40
4.2.2 Kullanıcı arayüz yazılımı	47
5. BULGULAR.....	52
6. SONUÇ VE DEĞERLENDİRME	54
KAYNAKLAR	56
ÖZGEÇMİŞ	58

İNTERNET TABANLI UZAKTAN DENETİM İÇİN BİR ARAYÜZ TASARIMI

(Yüksek Lisans Tezi)

Muammer Kadir KARAGÖZ

**MUĞLA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

2008

ÖZET

Günden güne uzaktan denetim ihtiyacı artmakta ve diğer alanlara yayılmaktadır. Uzaktan denetim ve izleme sağlayan sistemler sayesinde firmalar, bazı ürün sorunları uzaktan anında giderebilecek ve yeni ürün geliştirme çalışmalarını daha doğru yönlendirebilecektir. Bu sistemler aynı zamanda eğitim alanında uzaktan kuramsal eğitime destek olarak uzaktan laboratuvar eğitiminin yapılabilmesini sağlayacaktır.

Bu tezde, uzaktan cihazların denetlenebilmesi ve izlenebilmesi için İnternet tabanlı bir arayüz sistemi tasarımı gerçekleştirilmiştir. Sistemin kullanıcı arayüz yazılımı Java Applet teknolojisi kullanılarak tasarlanmıştır. Tasarlanan sistemde, arayüz yazılımı vasıtasıyla kullanıcı tarafından gönderilen istekler mikrodenetleyicili bir arayüz donanımı tarafından alınıp değerlendirildikten sonra cihaz yönlendirilmektedir. Örnek uygulama için bir motor sürücüsü İnternet üzerinden denetlenmiş ve izlenmiştir.

Anahtar Kelimeler: Uzaktan denetim, İnternet, Java, Soket programlama, Çevrimiçi mühendislik.

Sayfa adedi: 58

Tez yöneticisi: Yrd. Doç.Dr. Aydın ÇETİN

DESIGN OF AN INTERNET BASED REMOTE CONTROL INTERFACE
(M. Sc Thesis)

Muammer Kadir KARAGÖZ

MUĞLA UNIVERSITY
INSTITUTE of SCIENCE and TECHNOLOGY

2008

ABSTRACT

Day by day, the remote control requirements are rising and expanding to many areas. The firms immediately will be able to resolve some of the product problems and correctly orientate development studies of new products by force of the systems which supply remote control and monitoring. These systems, in the education area, will also make it possible to realize the remote laboratory education supporting distance theoretical education.

In this thesis, a prototype remote control interface system has been designed to control and monitor machines across the Internet. User interface software of the system has been designed using the Java Applet Technology. In the designed system, the tasks which sent by user via interface software, are accepted and evaluated by interface device with a microcontroller. Thereafter, the machine is controlled and monitored by the microcontroller. For the sample application, an inverter has been controlled and monitored over the Internet.

Key Words: Remote control, The Internet, Java, Socket programming, Online engineering

Page number: 58

Adviser: Asst.Prof.Dr. Aydın ÇETİN

ŞEKİLLER DİZİNİ

<u>Sekil No</u>	<u>Sayfa No</u>
Şekil 3.1 TCP/IP Katmanları.....	8
Şekil 3.2 IP datagramı.....	8
Şekil 3.3 TCP parça yapısı	9
Şekil 3.4 Applet yaşam döngüsü	15
Şekil 4.1 Uzaktan denetim arayüz devresi blok diyagramı.....	16
Şekil 4.2 Ethernet modülü devre şeması.....	18
Şekil 4.3 VE kapıları ile seviye kaydırma.....	20
Şekil 4.4 Ethernet modülünün mikrodenetleyici ile bağlantısı	21
Şekil 4.5 Ethernet modülü besleme devresi	21
Şekil 4.6 DIP pakete sahip PIC18F452 bağlantı uçları.....	22
Şekil 4.7 PIC18F452 osilatör bağlantısı	24
Şekil 4.8 Arayüz devre şeması	25
Şekil 4.9 SPI veri iletişimde ana/uydu bağlantısı.....	27
Şekil 4.10 RS485 genel ağ topolojisi.....	31
Şekil 4.11 SN75176A'nın mikrodenetleyici ile bağlantısı	31
Şekil 4.12 MM420 güç terminalleri.....	34
Şekil 4.13 MM420 denetim terminalleri.....	34
Şekil 4.14 RS485 sonlandırıcı	35
Şekil 4.15 RS485 hattı sonlandırma	36
Şekil 4.16 Telegram yapısı.....	40
Şekil 4.17 Açık veri bloğu.....	41
Şekil 4.18 Kullanıcı arayüz appleti.....	47
Şekil 4.19 Akış şeması ana bloğu	48
Şekil 4.20 Azalt ve Artır tuşları için akış şeması	50
Şekil 4.21 Dur ve Dön tuşları akış şeması	50
Şekil 4.22 TersYön ve P tuşları akış şeması	51

ÇİZELGELER DİZİNİ

<u>Çizelge No</u>	<u>Sayfa No</u>
Çizelge 4.1 Tanımlı görev kimlikleri.....	42
Çizelge 4.2 Tanımlı yanıt kimlikleri.....	43
Çizelge 4.3 Denetim sözcüğü görevleri.....	43
Çizelge 4.4 Durum sözcüğü geri bildirimleri.....	44

SEMBOLLER ve KISALTMALAR DİZİNİ

f_{osc} : Osilatör frekansı

Kısaltmalar

BOP	Temel Operatör Paneli (Basic Operator Panel)
BRG	Baud Oran Üretici (Baud Rate Generator)
DAQ	Veri Edinme (Data Acquisition)
DTMF	Çift Tonlu Çoklu Frekans (Dual-tone multi-frequency)
EJB	(Enterprise Java Beans)
FTP	Yapraklı Burgulu Çift (Foiled Twisted Pair)
G/Ç	Giriş/Çıkış
IP	İnternet Protokolü
IR	Kızılötesi (InfraRed)
JAR	Java Arşiv
JDBC	Java Veritabanı Bağlanırlığı (Java Database Connectivity)
JSP	Java Sunucu Sayfaları (Java Server Pages)
JVM	Java Sanal Makinesi (Java Virtual Machine)
MSSP	Ana Eşzamanlı Seri Bağlantı noktası (The Master Synchronous Serial Bağlantı noktası)
PLL	Faz Kilitlemeli Döngü (Phase Locked Loop)
PSP	Paralel Uydu Bağlantı noktası (Parallel Slave Bağlantı noktası)
RF	Radyo Frekans (Radio Frequency)
SCADA	Veri Tabanlı Kontrol ve Gözetleme Sistemi (Supervisory Control and Data Acquisition)
SCK	Seri Saat (Serial Clock)
SI, SDI	Seri Giriş, Seri Veri Girişi (Serial Input, Serial Data Input)
SO, SDO	Seri Çıkış, Seri Veri Çıkışı (Serial Output, Seri Data Output)
SPI	Seri Çevresel Arayüz (Serial Peripheral Interface)
STP	Korumalı Burgulu Çift (Shielded Twisted Pair)
TCP	İletim Denetimi Protokolü (Transmission Control Protocol)
UDA	Uzaktan Denetim Arayüzü
UDP	Kullanıcı Datagram Protokolü (User Datagram Protocol)

USART	Evrensel Eşzamanlı Eşzamansız Alıcı Verici (Universal Synchronous Asynchronous Receiver Transmitter)
USS	Evrensel Seri Arayüz (Universelles-Serielles-Schnittstellenprotokoll)

1. GİRİŞ

Günümüze kadar elektrikli cihazların uzaktan denetimi için farklı yöntemler gerçekleştirilmiştir. Makine denetiminde kablosuz denetim (IR, RF), yapay zekâ uyarlamalı denetim gibi çok sayıda yöntem geliştirilmiştir. Özellikle İnternet'in yaygınlaşmasından sonra oldukça uzak noktalardan hatta dünyanın diğer ucundan makineleri denetleme ve izleme olanağı doğmuştur. Teknolojinin gelişimi ile birlikte uzaktan denetim uygulamaları giderek artmakta ve İnternet uygulamalar için en uygun ortam olarak görünmektedir.

Uzaktan denetim sistemleri, özellikle endüstriyel cihaz üretimi yapan firmalar için uzaktan hızlı bakım hizmeti ile satış sonrası müşteri memnuniyetini artırmak, maliyetleri azaltmak ve daha üstün özelliklere sahip yeni makineler üretebilmek gibi avantajlar sunmaktadır. Mesleki ve teknik eğitim alanında uzaktan erişim arayüzü, mevcut cihazların uzaktan denetimine ilişkin eğitim uygulamaları yapılabilmesine imkan sağlamaktadır. Dolayısıyla, uzaktan erişim ile uzaktan uygulamalı eğitimin gerçekleştirilmesine ve eğitim kalitesinin artırılmasına ve uygulamalı eğitim maliyetlerinin düşürülmesine katkı sağlamaktadır.

Günümüzde farklı makineler için, İnternet üzerinden denetim ve izleme sistemleri mevcuttur. Bu çalışmada mevcut uzaktan denetim ve izleme sistemlerinden farklı bir yöntem sunulmuştur. Diğer uzaktan denetim sistemleri iki uzak bilgisayar arasında veri alış-verişi mantığı üzerine kuruludur. Sunucu tarafta bu işe adanmış bir bilgisayar kullanılması hem maliyeti artırmakta hem de daha sık sorun yaşanmasına neden olmaktadır.

Bu çalışmada uzaktan denetim ve izleme işlevlerinin herhangi bir uzak bilgisayardan yapılabilmesi ve sunucu tarafta bilgisayar yerine tasarlanacak bir arayüz devresi bulunmasının daha uygun olduğu düşünülmüştür. Tasarlanacak arayüz devresinin duruma göre uyarlanabilmesi için flaş program belleğine sahip bir mikrodenetleyici tercih edilmiştir. İnternete bağlı herhangi bir bilgisayar ya da mobil cihaz tarafından denetim ve izleme yapılabilmesi için özel hazırlanmış bir Java Applet'i bulunduran Web sayfasının kullanılması uygun görülmüştür.

Tezin ikinci bölümünde günümüzdeki mevcut uzaktan denetim sistemleri incelenmiş ve üstünlük ve sakıncaları değerlendirilmiştir. Üçüncü bölümde, insan-makine arayüzü geliştirmek için kullanılan Java teknolojisi ve ağ programlama

mantığı üzerinde durulmuştur. Dördüncü bölümde, tasarlanan sistemin bileşenleri donanımsal ve yazılımsal olarak açıklanmıştır. Bu bölümde aynı zamanda arayüz sisteminde kullanılan teknolojilere değinilmiştir. Beşinci bölümde, tasarlanan arayüz sistemine ilişkin bulgular sunulmuştur. Altıncı bölümde, çalışmaya ilişkin sonuç ve değerlendirme sunulmuştur.

2. LİTERATÜR ÖZETİ

Boğaziçi Üniversitesi'nde Alp vd. (2003) tarafından yapılan bir çalışmada, İnternet üzerinden standart bilgisayar özelliklerine sahip bir robot (APES) için erişim yöntemi tasarlanmıştır. APES seçici olarak görebilen, olabildiğince ufak bir gövde üzerinde iki serbestlik dereceli bir başa sahip, doğrusal ve dönme hareketleri yapan mobil bir robottur. Gerçek zamanlı veri ve görüntü aktarımı robot üzerinde bulunan sunucu yazılım ile kullanıcı arayüzü yüklü herhangi bir uzak bilgisayar arasında gerçekleştirilebilmektedir. Arayüz yazılımları Java teknolojisi kullanılarak gerçekleştirilmiştir.

Gazi Üniversitesi'nde Çolak vd. (2005) tarafından yapılan bir çalışmada seri ve paralel RLC devrelerinin İnternet üzerinden analizine olanak sağlayan, elektrik eğitimine yönelik bir eğitim aracı geliştirilmiştir. Bu uygulama ile İnternet üzerinden elektrik devrelerinin kararlı ve geçici durum tepkileri grafiksel olarak izlenebilmektedir. Sunucu yazılımı olarak Apache Server, etkileşimli Web sayfası hazırlamak için PHP, veri tabanı hazırlamak için MySQL ve hesaplama yaptırmak ve grafik çizdirmek için MATLAB programı kullanılmıştır. Benzetim çalışmalarına sadece Gazi Üniversitesi yerel ağından erişilebilmektedir.

Dumlupınar Üniversitesi'nde Atabaş vd. (2007) tarafından yapılan bir çalışmada ısıtma sistemlerinin otomasyonu ve internet üzerinden denetimi için model geliştirilmiştir. Geliştirilen model, Web tarayıcısı olan herhangi bir bilgisayar üzerinden elektronik denetim kartının bağlı olduğu ve denetim yazılımı yüklü sunucu bilgisayar yardımıyla gerçek zamanlı sıcaklık izleme ve denetim sağlamaktadır. İnternet tabanlı endüstriyel sıcaklık ölçme ve denetim sistemi üzerine benzer bir çalışma Kırıkkale Üniversitesi'nde Arslan (2005) tarafından gerçekleştirilmiştir.

Selçuk Üniversitesi'nde Eşme (2006) tarafından yapılan bir çalışmada GSM modem ile uzaktan denetlenebilen bir taşınabilir kalp cihazı tasarlanmıştır. Yazılımlar için LabVIEW, donanım kartı olarak National firmasının DAQ kartı kullanılmıştır.

Sakarya Üniversitesi'nde Tunalı (2004) tarafından yapılan bir çalışmada telefon hattına bağlı DTMF sinyali gönderebilen bir telefon ile uzak bir noktada telefon hattına bağlı kontrol kartı aracılığıyla bir bilgisayarı açıp kapatabilen bir sistem tasarlanmış ve gerçekleştirilmiştir.

Yıldız Teknik Üniversitesi'nde Ünlü (2007) tarafından yapılan bir çalışmada bilgisayarla kablosuz haberleşen bir mobil robotun İnternet üzerinden denetimi tasarlanmıştır. İnternet üzerinden denetim için C++ programlama dili kullanılarak sunucu/istemci arasında TCP/IP protokolü kullanılmıştır.

Gazi Üniversitesi'nde Korkmaz (2004) tarafından yapılan bir çalışmada ev ve iş yerlerinde bulunan cihazların GSM ağları üzerinden uzaktan denetimi gerçekleştirilmiştir. Geliştirilen sistem 4 adet cihazın aç/kapat denetimini yapabilmekte ve kullanıcıları sesli mesaj ile bilgilendirmektedir.

Kocaeli Üniversitesi'nde Karaçor (2004) tarafından yapılan çalışmada, cep telefonu tabanlı mobil SCADA otomasyon sistemi gerçekleştirilmiştir. Bu sistem için özel geliştirilmiş Java uygulaması yüklü bir cep telefonu ile İnternet üzerinden uzaktaki SCADA sistemi denetlenebilmekte ve izlenebilmektedir. SCADA sistemi ile haberleşme Web sunucusu tarafından gerçekleştirilmektedir.

Yen-Shin (2002), bulanık mantık denetimli evirici sisteminin İnternet tabanlı izlenmesi ve denetimi için bir sistem önermektedir. Sistem istemci/sunucu tabanlıdır. İstemci tarafta Visual Basic ile hazırlanmış insan-makine arayüzü kullanılarak kullanıcıdan komut alınır. Aynı zamanda istemci, sunucunun oluşturduğu veritabanına Web tarayıcı kullanarak erişebilmektedir. Sunucu yine Visual Basic'te yazılmış yazılım ile bulanık mantık denetim, PLC ile iletişim ve istemciden komut alımı gerçekleştirir. Sunucu aynı zamanda evirici üzerinden aldığı verileri veritabanına kaydetmektedir. Çalışmada, eviricinin denetimi ve sıcaklık bilgisi okumak amacıyla PLC kullanılmıştır.

Uzaktan denetim ve izleme üzerine yapılan çalışmalar incelendiğinde çok uzun mesafelerden denetim sağlayan İnternet ve GSM teknolojilerinin kullanımının arttığı görülmektedir. İnternet tabanlı denetim sağlayan sistemler genellikle sunucu/istemci mantığı üzerinde kurulmaktadır. İstemci ve sunucu bilgisayarlar için hazırlanmış özel yazılımlar sayesinde denetim ve izleme yapılmaktadır. Kullanılan programlama dilleri, denetim kartları ve denetlenen makineler çeşitlilik göstermektedir. Bu tür sistemlerde Java teknolojilerinin kullanımının giderek arttığı görülmektedir. GSM üzerinden yapılan denetim sistemleri de benzer özellikler taşımaktadır. İstemci olarak, özel hazırlanmış arayüz yazılımını barındıran cep telefonu kullanılarak sunucu taraftaki yazılım ile veri-alışverişi gerçekleştirilmektedir. Bunun yanında

sabit bir cep telefonu ve ona baęlı kontrol kartını sunucu olarak kullanan sistemler de mevcuttur. Ancak bu sistemler sınırlı özelliklere sahip olduęu gibi genellikle aç/kapat denetim için kullanılmaktadır.

İnternet tabanlı ya da Web tabanlı denetim sistemleri içinde bilgisayara ihtiyaç duymadan doğrudan Ethernet aęına baęlanan bir arayüz sistemi bulunmamaktadır. Arayüz sistemi yerine bilgisayar kullanan sistemler virüslere açık olduęu için daha yüksek kurulum ve bakım masrafı gerektirir.

Mevcut sistemlerin çoęu güvenlik nedeniyle istemci tarafta sabit bir yazılım kullanmaktadır. Yazılıma sahip olmayan hiçbir bilgisayar sisteme erişememektedir. Web tarayıcıya sahip herhangi bir bilgisayar ile erişilebilen sistemlere az rastlanmaktadır. Böyle sistemler kullanıcı tanımlı olarak sunulduğunda diğer sistemlere nazaran olaęanüstü esneklik getirmektedir. Web tarayıcısı olmayan bilgisayar olmadığı düşünöldüğünde İnternete baęlı herhangi bir bilgisayar istemci olabilir. Tek bilgisayara baęlı kalınmaması bir arıza durumunda erişimin başka bir bilgisayardan yapılabilmesine olanak tanır.

Bu çalışmada, önceki çalışmalar ve mevcut gereksinimler göz önünde bulundurularak, İnternete baęlanabilen herhangi bir bilgisayar ile geliştirilen arayüz devresi üzerinden doğrudan makine denetimi ve izleme saęlayan bir arayüz sistemi hedeflenmiştir.

3. JAVA AĞ PROGRAMCILIĞI

Java ağ paketi, java.net ile sağlanan güçlü ağ erişim kütüphanesi, Java'nın İnternet ve genel anlamda ağ programcılığında kullanılmasının önemli nedenlerinden biridir. Bu paket yüksek seviyeli sınıflar içerdiği gibi düşük seviyeli sınıflar da içermektedir. Bu çalışmada düşük seviyeli soket programlama sınıfları üzerinde durulacaktır.

3.1 Java Programlama Dili

Java, Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış açık kodlu, nesneye yönelik, platformdan bağımsız, yüksek performanslı, çok işlevli, yüksek seviye, adım adım işletilen (interpreted) bir dildir (Anonim C).

Java programları derlendiklerinde çalıştırılabilir kod yerine baytkod (bytecode) denilen bir komut kümesi üretir. Oluşturulan baytkod'lar Java Sanal Makinesi (JVM, Java Virtual Machine) bulunduran herhangi bir bilgisayar ya da elektronik cihaz tarafından yürütülebilir. JVM baytkod'ları yorumladığından dolayı çalıştırılabilir kodlara göre biraz daha yavaş çalışır. Ancak bu düzen sayesinde platform bağımsızlığı özelliği sağlanır.

Java ilk çıktığında daha çok küçük cihazlarda kullanılmak için tasarlanmış ortak bir platform dili olarak düşünülmüştür. Ancak platform bağımsızlığı özelliği ve C++ dilinden çok daha üstün ve güvenli bir yazılım geliştirme ve işletme ortamı sunduğundan, hemen her yerde kullanılmaya başlanmıştır.

İnternet tabanlı olup veri tabanlarına bağlantılar yapabilen Java, birçok diğer programlama dilinden daha üstün özelliklere sahiptir. Servlet, Java Sunucu Sayfaları (JSP, Java Server Pages), Java Veritabanı Bağlanırlığı (JDBC, Java Database Connectivity), Kuruluş Java Bezelyeleri (EJB, Enterprise Java Beans) gibi Java teknolojileri bu tür uygulamalar yazmada büyük kolaylıklar sağlamaktadır.

Java teknolojisinin kaynak kodu 13 Kasım 2006'dan bu yana açıktır. Bu sayede gelişen teknolojiye rahatlıkla ayak uydurabilmektedir. Belirli kurallara uyulmak kaydıyla Java teknolojisini isteyen kişi veya kurumlar geliştirip değişiklik yapabilirler.

Sun Microsystems, JVM (Java Virtual Machine) veya Java programlama diline bağılı alt teknolojiler yazmak için belirli kurallar koymuştur; bu kurallar topluluğuna “belirtiler” denir. Alt teknoloji geliştirecek olan her kurum bu standart belirtiler çerçevesinde hareket etmek zorundadır. Bu belirtilere sadık kalan her yazılım firması Java sanal makinesi veya Java programlama diline bağılı alt teknolojiler yazabilir (örneğin Application Server - Uygulama Sunucusu). Eğer bu standart belirtilere sadık kalınmayıp standart dışı bir JVM veya Java programlama diline bağılı alt teknolojiler yazılmaya kalkılırsa hukuki bir sorun ortaya çıkabilir. Java bu özelliğiyile de gelişmeye açık bir teknolojidir (Anonim C).

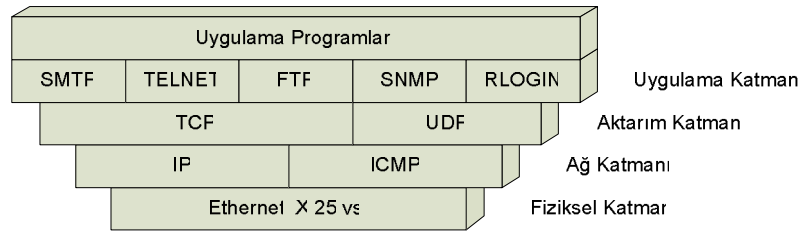
Java teknolojisinin diğere önemli bir özelliğiyile de “çöp toplama sistemi (garbage collector)”dir. Bu yapı da standart belirtiler arasındadır. Bu yapı sayesinde programcı program kodları arasında çok fazla yer tutan denetleme ve düzenleme işlemlerinin çoğunu takip etmekten kurtulur. Çöp toplama sistemi daha önceden oluşturulmuş, ancak şu an için kullanılmayan ve bellekte gereksiz yer işgal eden nesnelere belirleyerek otomatik olarak siler. Böylece bellek yönetim (memory management) yükü tasarımcıdan JVM’e geçmiş olur. Diğere dillerde, örneğiyile C++ programlama dilinde, oluşturulan nesnelere yok edilme sorumluluğuyile programcıya aittir. Unutulduğuyile veya doğru yapılmadığıyile ise programın kilitlenmesi hatta sistem kilitlenmesi ile karşılaşılabilir. Çöp toplayıcının ne zaman ortaya çıkıp temizleme yapacağı belirli değildir. Eğer bellekte JVM için ayrılan kısım dolmaya başlamışsa çöp toplayıcı devreye girerek kullanılmayan nesnelere bellekten siler (Anonim C).

3.2 İnternet İletişim Kuralları

İnternet dünya üzerindeki milyonlarca bilgisayarı bağlayan dünya çapında bir bilgisayar ağıdır. Günümüzde İnternet’e sadece bilgisayarlar değil pek çok cihaz bağlanabilmektedir. İnternet dilinde bu cihazlar uç sistemler olarak adlandırılır. Uç sistemler ve İnternet’in diğere parçaları, İnternet içinde bilgi göndermek ve almayı denetlemek için protokoller çalıştırır. İnternetin tüm protokollere TCP/IP olarak adlandırılır (Gümüşkaya, vd. 2003).

3.2.1 Temel TCP/IP protokolleri

TCP/IP içindeki protokoller günümüz İnternet hizmetlerinin alt yapısını oluşturur. TCP/IP protokol kümesinin sahip olduğu mimari katmanlı olup Şekil 3.1’de TCP/IP katmanları görülmektedir. Her bir katman altındaki katmandan hizmet bekler, üzerindeki katmana ise hizmet eder. Buna göre fiziksel katmanı ağ katmanına, ağ katmanı ise aktarım katmanına hizmet vermektedir.



Şekil 3.1 TCP/IP Katmanları

3.2.1.1 İnternet protokolü

Ağ katmanı seviyesinde olan en önemli protokol İnternet protokolüdür (IP). Hedef adresi gibi bilgileri içeren bir paket anahtarlamalı ağ üzerinden gönderilen bilgi paketi veya birimine datagram denir. IP, datagramların farklı fiziksel katmanlardan geçerek hedef adrese yönlendirilmesinden sorumludur. Bu işlemin gerçekleştirilebilmesi için aktarım katmanından gelen bilgi paketine IP başlığı eklenir. Şekil 3.2’de bir IP datagramı görülmektedir.

Sürüm	IHL	Servis tip	Toplam uzunluk	
Tanımlama			Bayrak	Parça offset
Yaşam süresi (TTL)		Protoko	Sağlama toplamı	
Kaynak adresi				
Hedef adres				
TCP başlığı ve iletilen bilg				

Şekil 3.2 IP datagramı

IP başlığı içinde göndericinin (kaynak) ve alıcının (hedef) İnternet adresleri, protokol numarası ve sağlama toplamı vardır. Alıcının İnternet adresi, iletinin gönderilmek istendiği bilgisayarın adresidir. Bu adres sayesinde, aradaki yönlendiriciler ve geçit yolları bu datagramı nereye yönlendireceklerini bilir. Protokol

numarası bu datagramı TCP'ye vermesi gerektiğini söyler. Sağlama toplamı IP başlık bilgisinin yolda bozulup bozulmadığını ya da yanlış adrese gidip gitmediğini denetlemek için kullanılır (Baykal, 2005).

İnternet protokolü, İnternet'e bağlı her bilgisayar ve yönlendirici arabiriminin bir IP adresi olmasını gerektirir. TCP/IP kullanan bir ağda her IP adresi 32 bit uzunluğunda ve benzersiz olmalıdır. Okunmasını kolaylaştırmak için bu adresler dört adet onluk düzende sayı(192.168.34.48 gibi) ile ifade edilir (Kaplan, 2000).

İnternet protokolünün hizmet verdiği aktarım katmanında TCP ve UDP olmak üzere iki protokol bulunmaktadır.

3.2.1.2 İletim denetimi protokolü

İletim denetimi protokolü (TCP) İnternet'te kullanılan temel protokoldür ve bağlantı-tabanlıdır. Bilgi gönderilmeden önce iki makine arasında bir bağlantı kurulur, daha sonra bilgi gönderilir. Gönderim tamamlanınca da bağlantı kesilir. TCP, dosya aktarımları ve uzaktan bağlantılar gibi kritik öneme sahip görevler için tercih edilir.

TCP'nin temel görevleri şöyle sıralanabilir:

- Bir üst katmandan gelen verinin uygun uzunluktan parçalara bölünmesi
- Alıcı tarafta aynı biçimde sıraya koyulabilmesi için her parçaya sıra numarası verilmesi
- Kaybolan veya bozuk gelen parçaların tekrarlanması

TCP protokolü bu görevleri yerine getirmek için veri parçalarının önüne başlık ekler. Başlık ve veri parçası TCP parçası olarak anılır. TCP parça yapısı Şekil 3.3'de gösterilmiştir.

Gönderici port numarası		Alıcı port numarası	
Sıra numarası			
Onay numarası			
Başlık Uzunluğu	Ayrılmaç	Bayrak	Pencere
Sağlama toplam		Acil işaret	
Kullanıcı verisi			

Şekil 3.3 TCP parça yapısı

Gönderici bağlantı noktası numarası iletiyi gönderen uygulamayı, alıcı bağlantı noktası numarası ise iletiyi alması istenen uygulamayı işaret eder. 32 bitlik sıra numarası ile 32 bitlik onay numarası, gönderici ve alıcı tarafından güvenilir veri aktarımı için kullanılır.

4 bitlik başlık uzunluğu, TCP başlığının uzunluğunun kaç tane 32 bitlik sözcükten ibaret olduğunu belirtir. TCP başlığı, en az 5 en fazla 15 sözcük uzunluğunda olabilir. Bayrak alanı 6 bayrak içerir. Bilgisayarlar arasındaki denetim kodlarının gönderilmesi için kullanılır. Pencere alanı, akış denetimi için veri paketlerinin boyunu denetlemek amacıyla kullanılır. Sağlama toplamı verinin ve başlığın hatasız aktarılıp aktarılmadığını sınamak için kullanılır. Acil işareti, ivedi olarak aktarımı sonlandırma vb. durumlarda kullanılır. Acil veri alıcının uygulama katmanında öncelikle değerlendirmesi gereken veridir. Kullanıcı verisi, ivedi olarak değerlendirilmesi istenen verinin bölüm içindeki yerini işaret eder.

3.2.1.3 Kullanıcı datagram protokolü

Kullanıcı datagram protokolü (UDP, User Datagram Protocol) ve TCP aynı katmana ait protokollerdir. UDP'nin TCP'den farkı, sorgulama ve sınama amaçlı, küçük boyutlu verinin aktarılması için kullanılıyor olmasıdır. UDP'de veri küçük boyutlu olduğu için parçalamaya gerek duyulmaz (Çölkesen, vd. 2002).

Ayrıca TCP'den farklı olarak UDP'de veri iletimi bağlantısızdır. Paketler gönderim sırasında kaybolabilir veya sıraları bozuk olarak hedefe ulaşabilir. UDP Hızlı olması sebebiyle oyunlarda ve çoklu ortam uygulamalarında UDP yaygın olarak kullanılır (Gümüşkaya, vd. 2003).

3.2.1.4 Haberleşme bağlantı noktaları

TCP protokolü bir makineden çalışan bir uygulamaya gelen veya bu uygulamadan dışarı giden veri için haberleşme bağlantı noktalarını (bağlantı noktası) kullanır. Verinin doğru makineye gitmesi IP adresi ile sağlanırken makine içinde doğru uygulamaya gitmesi bağlantı noktaları aracılığıyla sağlanır. Bağlantı noktaları, 0 ile 65535 arasında değişen 16 bit tamsayılar ile tanımlanır. 0 ile 1023 arasındaki bağlantı noktaları iyi-bilinen bağlantı noktası numaraları olarak adlandırılır ve genellikle Hiper Metin Aktarımı İletişim Kuralı (HTTP, Hyper Text Transfer Protocol), Dosya Aktarımı İletişim Kuralı (FTP, File Transfer Protocol), Uzak

İletişim Ağı (TELNET, Telecommunication Network) , vb hizmetler için kullanılır. Bu nedenle iyi-bilinen bağlantı noktası numaraları kullanıcı uygulamaları için kullanılmamalıdır.

3.3 Soket Programcılığı

İstemciler ve sunucular soketler yardımıyla bağlantı ve iletişim kurarlar. Bağlantılar İnternet üzerine TCP kullanılarak oluşturulan iletişim bağlantılarıdır. Bazı istemci/sunucu uygulamaları bağlantısız UDP üzerine inşa edilir. Bu uygulamalar da iletişim için soketler kullanır. Soketler İnternet iletişiminin uç noktalarıdır. İstemciler, istemci soketleri oluşturur ve onları sunucu soketlerine bağlar. Bu bağlantı üzerinden veri alışı-verişi gerçekleştirilir (Javorski,1997).

Soketler, istemci veya sunucu programını içeren makinenin (host) IP adresi ve istemci ya da sunucu programı tarafından kullanılan bir bağlantı noktası ile tanımlanır. Soket bilgisi IP:Bağlantı noktası (örneğin 212.188.14.23:80) düzeninde gösterilir.

Java ağ paketi, java.net, İnternet veya bir TCP/IP ağı üzerindeki haberleşmeler için bir arabirim sağlar. Java.net paketinde bulunan Socket ve ServerSocket sınıfları, bir istemci ve bir sunucu arasında TCP protokolü tabanlı platform-bağımsız güvenilir haberleşme bağlantıları oluşturmada kullanılır. Aynı paket içerisinde yer alan DatagramPacket ve DatagramSocket sınıfları ise UDP protokolü ile platform bağımsız paket tabanlı haberleşme sağlar.

3.3.1 TCP soket programcılığı

TCP/IP soketleri, İnternet'te ana sistemler arasında, güvenilir, çift yönlü, kalıcı, noktadan-noktaya ve akış tabanlı bağlantılar için kullanılır.

Java'da iki tip TCP soketi vardır. Biri sunucular, diğeri istemciler içindir. Java ağ paketinde yer alan Socket sınıfı, ağ üzerindeki bir Java programı ile diğere uçtaki uygulama arasında iki yönlü bir haberleşme yolunun bir ucunu belirtir. ServerSocket sınıfı bir sunucunun istemci bağlantı isteklerini dinlediği sokettir ve sadece sunucularda kullanılır.

Bir soket 7 temel işlemi gerçekleştirebilir:

- Uzaktaki bir makineye bağlanma
- Veri gönderme

- Veri alma
- Bir bağlantıyı kapama
- Bir yerel bağlantı noktasına bağlanma
- Gelen veriyi dinleme
- Bağlantı noktasına uzak makinelerden gelen bağlantıları kabul etme

Hem istemciler hem de sunucular tarafından kullanılan Socket sınıfı, yukarıdaki işlemlerden ilk dördüne karşılık gelen metotlara sahiptir. Son üç işlem, istemcilerin bağlantı yapmalarını beklemeleri için, sadece sunucular tarafından gerekmektedir. Bu son üç işlem sadece sunucularda kullanılacak ServerSocket sınıfı tarafından sağlanmaktadır.

Soket tabanlı haberleşmede sunucu öncelikle belirli bir bağlantı noktası numarasından bağlantı isteği bekler. Kendisine bağlı bulunan istemcilere hizmet verebilmek ve aynı zamanda yeni istekleri de dinleyebilmek için her bir bağlantı için yeni bir Socket nesnesi oluşturur. İstemci tarafında ise istemci, sunucunun çalıştığı makinenin adını veya IP numarasını ve sunucunun bağlı olduğu bağlantı noktası numarasını bilir. Bir bağlantı isteği yapmak için, istemci sunucu adresi (IP veya adı) ve bağlantı noktası numarasını belirterek, sunucu ile bir bağlantı talebinde bulunur. Eğer sunucu çalışıyor ve istenen hizmet de belirtilen bağlantı noktası numarasında veriliyorsa, sunucu bağlantı isteğini kabul eder. Bağlantı kabulden hemen sonra, sunucu istemci ile veri iletişimi için farklı bir bağlantı noktasına için bir soket açar. Sunucu, bağlanan istemcinin isteklerine bakabilmesi ve aynı zamanda bağlantı isteklerini dinlediği özgün soketi dinlemeye devam edebilmek için yeni bir sokete gereksinim duyar. İstemci tarafında, eğer bağlantı kabul edilmişse, bir istemci soketi oluşturulur ve istemci sunucu ile haberleşmesinde bu soketi kullanır. Bu işlemlerden sonra istemci ve sunucu soketi üzerinde oluşturulan giriş-çıkış akışları (stream) üzerinden bilgi alış-verişi yapılabilir.

İstemci soketleri oluşturmak için kullanılan iki kurucu metot aşağıda gösterilmiştir:

Socket(String anaSistemAdi, int bağlantı noktası)

Socket(InetAddress ipAdresi, int bağlantı noktası)

İlk kurucu metot, adı ve bağlantı noktası numarası belirtilen yerel ana sisteme bağlanan bir soket oluşturur. Bu esnada parametrelerde bir sorun meydana gelirse

sorunun kaynağına bağlı olarak `UnknownHostException` veya `IOException` fırlatabilir.

İkinci kurucu metot ise, mevcut bir `InetAddress` nesnesi ve bir bağlantı noktası kullanarak soket oluşturur. Bu esnada parametrelerde bir sorun meydana gelirse `IOException` fırlatabilir.

Bir soket aşağıdaki metotlar kullanılarak herhangi bir anda onunla ilişkilendirilen adres ve bağlantı noktası için incelenebilir:

`InetAddress getAddress()` :Socket nesnesiyle ilişkili `InetAddress`'i döndürür.

`int getPort()`: Socket nesnesiyle ilişkilendirilen `InetAddress`'i döndürür.

`int getLocalPort()` : Socket nesnesinin bağlandığı yerel bağlantı noktasını döndürür.

Bir `Socket` nesnesi oluşturulduğunda, onunla ilişkilendirilen girdi ve çıktı akışlarına erişmek için de incelenebilir. Bu metotların her biri soketlerin ağ bağlantısı geçersiz olduğunda `IOException` fırlatır.

`InputStream getInputStream()`: Çağrılan soketle ilişkilendirilen `InputStream`'i döndürür.

`OutputStream getOutputStream()`: Çağrılan soketle ilişkilendirilen `OutputStream`'i döndürür.

`InetAddress` sınıfı IP adresini simgeler. `InetAddress` sınıfının statik `getByName()` metodu bir `InetAddress` nesnesi döndürmektedir. Böylece web-sayfalarının isimleri kullanılarak IP adresi alınabilir. `InetAddress` sınıfı `getByName()` gibi işlevsel pek çok metoda sahiptir.

`InetSocketAddress` sınıfı bir IP soket adresini (IP ve bağlantı noktası) simgeler.

3.3.2 UDP soket programcılığı

Java ağ paketinde UDP haberleşmeleri için üç temel sınıf bulunmaktadır. Bunlar `DatagramPacket`, `DatagramSocket` ve `MulticastSocket` sınıflarıdır. `DatagramPacket` sınıfı, veri baytlarını UDP datagram paketlerine doldurmaktadır. Bir `DatagramSocket` UDP datagramları hem göndermekte hem de almaktadır. `MulticastSocket`, birçok makine tarafından dinlenen bir sokettir ve çoklu yayın (multicasting) olarak adlandırılan haberleşme modelinde kullanılır.

3.4 Java Applet

Appletler, bir İnternet sunucusu üzerinden erişilen, İnternet üzerinden taşınan ve Web sayfalarıyla bütünleştirilebilen Java programlarıdır. Applet içeren bir Web sayfası tarayıcıda görüntüleneceği zaman applet yüklenir ve yürütülür. Applet çıkışı, tarayıcı görüntü alanında önceden ayarlanan yerinde gösterilir.

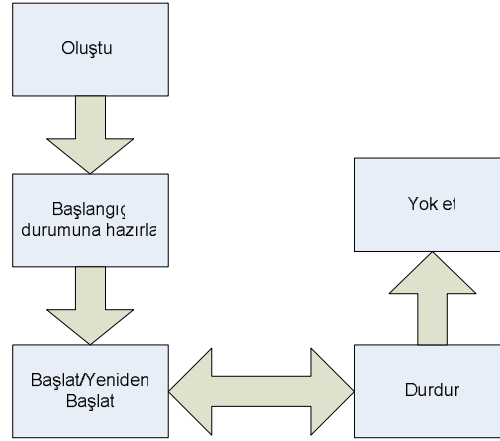
Appletler pek çok güvenlik sınırlamalarına sahiptir. Örneğin bir applet yerel dosya sistemine herhangi bir yoldan erişemez, bir yerel dosyayı okuyamaz, dosyaya yazamaz, dosyalar hakkında herhangi bir bilgiye ulaşamaz, istemcideki diğer programları çalıştıramaz, kullanıcının makinesi hakkında bilgiye ulaşamaz, vs. Sadece indirildikleri sunucuya bir soket bağlantısı açabilirler. Bazı özel durumlarda tüm bu güvenlik sınırlamaları bir Java Arşiv (JAR, Java Archive) dosyasına bir sayısal imza iliştilerilerek kaldırılabilir (Flanagan, 1997).

Java.applet paketi Java Uygulama Programlama Arayüz'leri (API, Application Programming Interface) içindeki en küçük pakettir. Tüm appletler Java.applet paketi içinde yer alan applet sınıfının bir alt sınıfı olarak tanımlandıkları için java.applet paketini içe aktarmak (import) zorundadırlar (Schildt, 2003).

Appletler Java uygulamaları gibi bir *main()* metoduna sahip değildir. Bir applet yüklendiği zaman ilk önce appletin kurucu metodu daha sonra *init()* metodu ve daha sonra *start()* metodu otomatik olarak çalışır. Applet yüklendikten sonra ekrana çizilmesini *paint()* metodu sağlar. Bir applet sonlandırıldığında ise *stop()* ve *destroy()* metodları çalışır (Anonim D).

- *init()*: Çağrılacak ilk metottur. Değişkenlere ilk değer ataması yapılan yerdir. Appletin çalışma zamanı boyunca bir kez çağrılır.
- *start()*: *init()* metodundan sonra çağrılır. Appleti barındıran Web sayfasının her görüntülenişinde yeniden bu metot çağrılır.
- *paint()*: Applet ilk çalıştırıldığında ve applet çıktısının yeniden çizilmesi gerektiği durumlarda çağrılır.
- *stop()*: Web tarayıcısı applet içeren sayfayı terk ettiğinde çağrılır. Applet görünür değilken çalışmasına gerek olmayan kanalları askıya almak için kullanılır.
- *destroy()*: Appletin tamamen bellekten çıkarılması gerektiğinde çağrılır. Bu metottan önce her zaman *stop()* metodu çağrılır.

Şekil 3.4'te bir appletin yaşam döngüsü görülmektedir.



Şekil 3.4 Applet yaşam döngüsü

Bir appletin bir Web tarayıcısında yürütülmesi için özel Java ekleri içeren bir HTML belgesi oluşturulur. APPLET imi, HTML belgesine konulan applet'e isim vermek ve özelliklerini belirlemek için kullanılır. Applet imi için gerekli belirleyiciler CODE, WIDTH ve HEIGHT'dır. CODE belirleyicisi, appletin class dosyasını içeren dosyanın adını belirtir. WIDTH ve HEIGHT belirleyicileri appletin gösterileceği alanının piksel cinsinden genişliğini ve yüksekliğini belirtir. Appletler HTML belgesinin bulunduğu dizin dışında başka bir dizine konulacaksa CODEBASE belirleyicisi kullanılır. Bu belirleyici appletin URL yolunu gösterir. APPLET iminin isteğe bağlı belirleyicileri de bulunmaktadır. Bunlar, ALIGN, HSPACE, VSPACE, NAME ve ALT'tır. Appletin gösterileceği alanı ve belgenin yerleşimini iyileştirmek için kullanılır.

APPLET bloğu içinde PARAM imi kullanılarak appletlere genel amaçlı parametreler geçirilebilir. Her PARAM imi bir parametreye değer atamak için kullanılır. PARAM iminin NAME ve VALUE olmak üzere iki belirleyicisi vardır. NAME belirleyicisi parametre adını, VALUE ise parametre değerini belirlemek için kullanılır. Appletler PARAM imi içerisindeki parametrelere erişebilmek için *getParameter()* metodunu kullanır.

Aşağıdaki örnek APPLET ve PARAM imlerinin kullanımı göstermektedir.

```

<APPLET CODE="OrnekApplet.class" WIDTH=300 HEIGHT=200>
<PARAM NAME=SEHIR VALUE="MUGLA"
</APPLET>
  
```

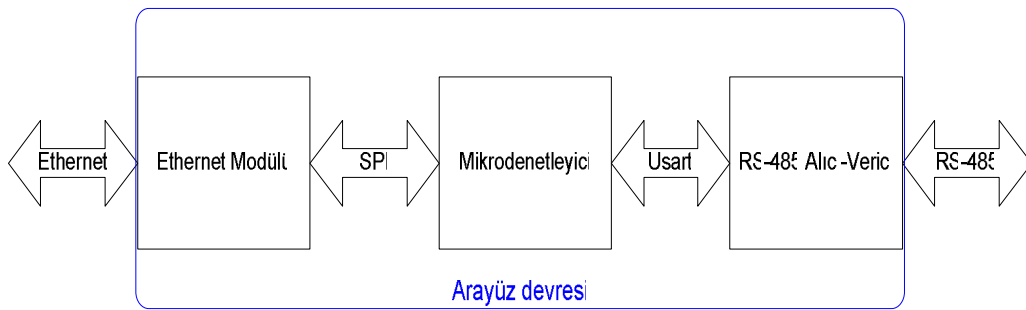
4. UZAKTAN DENETİM ARAYÜZÜ TASARIM VE UYGULAMASI

Uzaktan denetim arayüzü (UDA), donanım ve yazılım olmak üzere iki bileşenden oluşmaktadır. Yazılım, uygulama ve arayüz yazılımlarından oluşmaktadır. Uygulama yazılımı, mikrodenetleyicinin Ethernet modülü ve RS485 alıcı-vericisi arasındaki iletişimini sağlamaktadır.

Uygulamada, uzaktan denetimin bir sunucu bilgisayar üzerinden gerçekleştirilmesiyle yüksek maliyet ve bakım gerektiren durumların (virüs bulaşması, işletim sistemi kaynaklı sorunlar) daha sık oluşması gibi sakıncalar göz önünde bulundurularak sunucu taraf için donanım tasarlanmıştır. Arayüz yazılımı kullanıcının donanıma uzaktan erişimini sağlamaktadır. Arayüz yazılımının herhangi bir bilgisayardan erişilebilmesi için Web sayfasına gömülebilir olması ve etkileşimli olması düşülerek Applet teknolojisi kullanılmıştır.

4.1 Donanım

UDA devresi bir Ethernet ağına bağlanılarak gerek yerel ağdan gerekse İnternet'e bağlı herhangi bir cihazdan denetlenebilir. UDA devresi donanımsal olarak bir Ethernet modülü, bir mikrodenetleyici ve bir de RS485 alıcı-vericisinden oluşmaktadır. Şekil 4.1'de uzaktan denetim arayüz devresi blok diyagramı görülmektedir



Şekil 4.1 Uzaktan denetim arayüz devresi blok diyagramı

UDA devresinin Ethernet ağına bağlantısını sağlama, TCP ve UDP protokolleri ile veri alma ve gönderme gibi işlevleri yerine getirmesi için bir Ethernet modülü kullanılmıştır. Ethernet modülünü ve veriler doğrultusunda çıkışı denetlemek için bir

mikrodenetleyici kullanılmıştır. Çıkış hattında ise çok sayıda iletişim kuralı tarafından desteklenen RS485 fiziksel arayüzü için RS485 alıcı-vericisi kullanılmıştır.

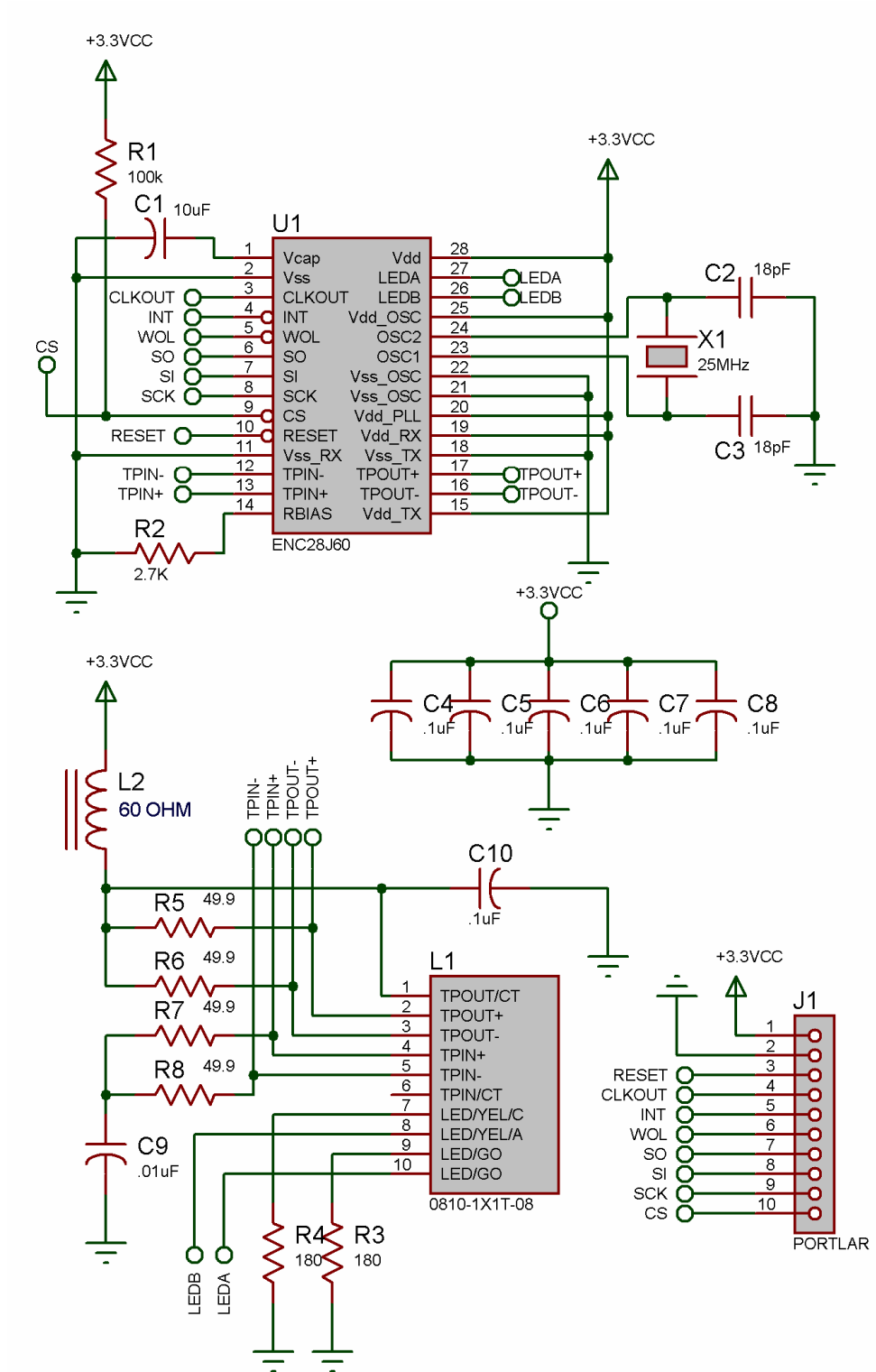
Tez çalışmasında tasarlanan arayüzün uygulamasına farklı bir örnek oluşturması amacıyla arayüzün bir asenkron motor sürücü/denetleyicisinin İnternet üzerinden programlanması hedeflenmiştir. Sürücü olarak Siemens Micromaster 420 motor sürücüsü seçilmiştir.

4.1.1 Ethernet Modülü

UDA devresini Ethernet ağına bağlamak için EDTP Electronics Inc. firmasının Frame thrower Ethernet modülü kullanılmıştır. Ethernet modülü Microchip firması tarafından üretilen SPI arayüzlü Ethernet denetleyicisi ENC28J60 üzerine kurulmuştur. Şekil 4.2’de Ethernet modülünün devre şeması verilmiştir.

Ethernet modülünün üzerinde Ethernet ağıyla fiziksel bağlantısını sağlayan RJ-45 konnektörü ile denetim ve besleme için bağlantı noktaları bulunmaktadır. Bağlantı noktaları ve işlevleri aşağıda verilmiştir.

- **3.3V:** +3,3V besleme girişi.
- **GND:** Toprak ucu.
- **RESET:** Düşük etkin sıfırlama girişidir. Dışarıdan yongayı sıfırlamak için eşzamansız bir metot sağlar.
- **CLKOUT:** Ana denetleyici saati olarak ya da sistemde bulunan başka bir devre için saat kaynağı olarak kullanılabilen programlanabilir saat işareti sağlar.
- **INT:** Düşük etkin kesme çıkış ucu. Yongada kesme meydana gelip gelmediğini ana denetleyiciye bildirmek için kullanılır.
- **WOL:** Ayrılmış işlev. Daima bağlantısız bırakılır.
- **SO:** SPI arayüz için veri çıkış ucu.
- **SI:** SPI arayüz için veri giriş ucu.
- **SCK:** SPI arayüz için saat giriş ucu. SCK’nın düşen kenarlarında veri alış-verişi gerçekleşir.
- **CS:** SPI arayüz için düşük etkin yonga seçici giriş ucu. Mantıksal 0’da iken çalışma gerçekleşebilir.



Şekil 4.2 Ethernet modülü devre şeması

Ethernet denetçisi:

Ethernet modülünde, Ethernet denetçisi olarak Microchip firmasının ürettiği ENC28J60 yongası kullanılmaktadır. ENC28J60 bir endüstri standardı olan Seri Çevresel Arayüz (SPI, Serial Peripheral Interface) destekli bir bağımsız Ethernet denetçisidir (Anonim B).

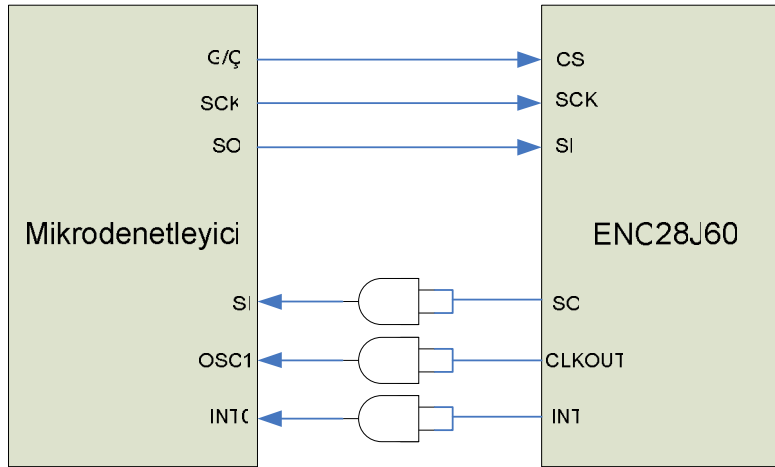
ENC28J60 yongasının tipik özellikleri şunlardır:

- IEEE 802.3'ye tam uyumlu
- Tümüleşik MAC
- Otomatik kutup saptama ve düzeltmeli bir 10Base-T bağlantı noktası desteği
- Tam ve yarım-çift yön modu desteği
- Çarpışma olunca programlanabilir otomatik yeniden gönderim
- Programlanabilir dolgu ve CRC üretimi
- Hatalı paketler için programlanabilir otomatik ret
- 20Mhz'e kadar saat hızıyla SPI arayüzü
- LINK, TX, RX, çarpışma ve tam/yarım-çift yön durumları için programlanabilir iki LED çıkışı
- 6 kesme kaynağı
- Hızlı veri taşınımı için dâhili DMA
- Değişik ağ protokolleri için donanım destekli sağlama-toplamı hesaplama
- Tekli-yayın, çoklu-yayın ve genel-yayın paketleri
- Programlanabilir alıcı paket süzme

ENC28J60 3,3V'luk besleme gerilimi ile çalışan yonga olmasına rağmen üretici firma tarafından 5V'luk sistemlerle kolayca tümleştirilebilecek şekilde tasarlanmıştır. ENC28J60 SPI girişleri, CS, SCK, SI ve RESET ucu 5V toleranslıdır. Dolayısıyla bu uçlar doğrudan 5V'ta çalışan ana denetleyici bağlantı noktalarına bağlanabilir. Ancak, ana denetleyicinin SPI ve kesme girişleri 3,3V toleranslı değildir. Bu nedenle mikrodenetleyici ENC28J60 üzerindeki 3.3V CMOS çıkışları

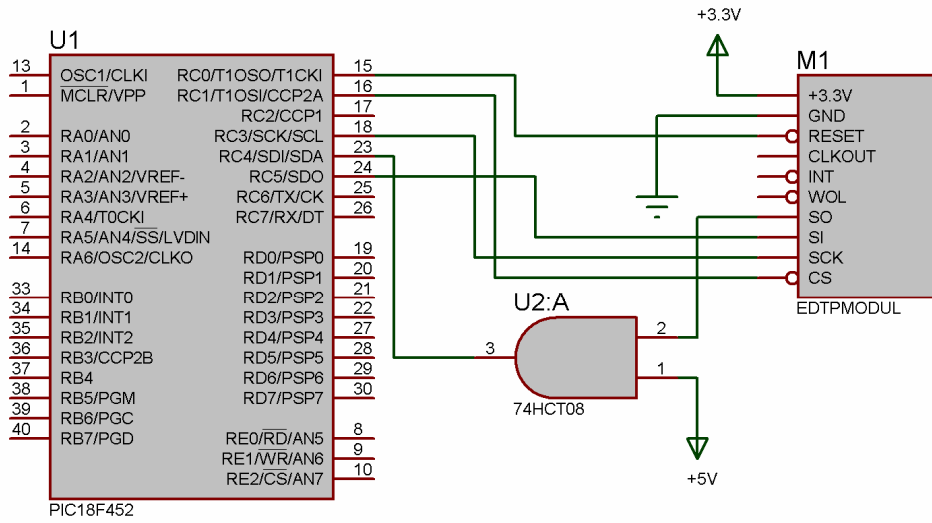
sağlıklı olarak okuyamayacaktır. Sorunu önlemek için SPI ve kesme girişlerinin tekyönlü seviye dönüştürücü üzerinden bağlanması gereklidir (Anonim B).

Seviye dönüştürmek amacıyla bir 74HCT08 (4'lü VE kapısı), bir 74ACT125 (4'lü 3-durumlu tampon) ya da TTL seviye girişli tampon ile diğer 5V CMOS tümleşik devreler kullanılarak ekonomik bir çözüm elde edilebilir. 3-durumlu tamponlar kullanmak SPI hattını birden fazla aygıtın paylaşabilmesini sağlar. Şekil 4.3'te VE kapıları ile yapılan seviye kaydırma görülmektedir (Anonim B).



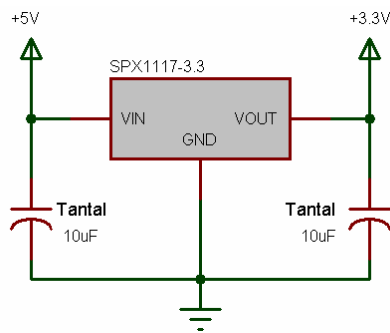
Şekil 4.3 VE kapıları ile seviye kaydırma

UDA devresinde Ethernet modülünün mikrodeneleyici ile bağlantısı Şekil 4.4'te gösterilmiştir. Ethernet modülünün (EDTPMODUL) CS, SCK, SI ve SO uçları SPI iletişim için yeterlidir. Veri çıkış ucu, SO'nun seviyesini 5V'a kaydırmak için 74HCT08 (VE kapısı) tümleşik devresi kullanılmıştır. 74HCT08, 4 adet iki girişli VE kapısı içerir. Sadece SO çıkışı VE kapılarından birinin 1 numaralı girişine bağlanır, diğer giriş (2 numaralı giriş) +5V besleme ucuna bağlanır. Alternatif olarak VE kapısı girişleri kısa devre edilip SO ucuna bağlanabilir. Olası bir kararsızlık durumuna mahal vermemek için kullanılmayan VE kapıları girişlerine +5V (mantıksal 1) uygulanmalıdır.



Şekil 4.4 Ethernet modülünün mikrodenetleyici ile bağlantısı

Ethernet modülü için besleme, sabit 3,3V çıkış gerilimi sağlayan SPX1117-3.3 doğrusal regülâtör tümleşik devresiyle sağlanmaktadır. Regülâtör giriş ve çıkış uçlarına 10 μ F tantal kondansatörler bağlanmıştır. SPX1117'nin kararlılığını korumak için en az 2,2 μ F (tantal ya da seramik) ya da 10 μ F (alüminyum) kondansatör gereklidir. Koşulsuz kararlılık için 10 μ F tantal kondansatör tercih edilmiştir. Şekil 4.5'te Ethernet modülünün besleme devresi görülmektedir.



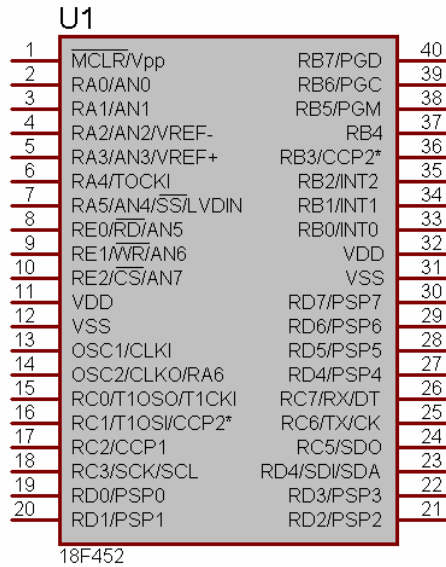
Şekil 4.5 Ethernet modülü besleme devresi

4.1.2 Mikrodenetleyici

Mikrodenetleyiciler tek başına bir bilgisayarda bulunması gereken giriş/çıkış birimleri, MİB, RAM ve kalıcı hafıza gibi birimleri üzerinde barındıran tümleşik devrelerdir.

Farklı pek çok üretici olmasına rağmen en çok tercih edilen Microchip firmasının ürettiği PIC mikrodenetleyicileridir. PIC mikrodenetleyicileri PIC10, PIC12, PIC16, PIC17, PIC18 (8-bit), PIC24F, PIC24H, dsPIC30, dsPIC33 (16-bit) ve PIC32 (32-bit) aileleri gibi farklı sınıflarda üretilmektedir. Her PIC ailesi farklı uygulama alanlarına hitaben geliştirilmiştir. Genel amaçlı pek çok uygulama için 8 bitlik mikrodenetleyiciler yeterlidir. PIC18 ailesi 16-bitlik komut seti, yüksek çalışma hızı, A/D çevrim, PWM, zamanlayıcılar, 32 kesmeye imkân vermesi, CAN arabirimi, RS232 arabirimi, seri ve paralel haberleşme protokolleri ile en güçlü 8-bit mikrodenetleyici ailesidir. Bu nedenle bu uygulama için yeterli donanımsal özelliklere sahip olan bir mikrodenetleyici, PIC18F452 seçilmiştir.

Şekil 4.6'da DIP paketine sahip bir PIC18F452 mikrodenetleyicinin bağlantı uçları göstermektedir.



Şekil 4.6 DIP pakete sahip PIC18F452 bağlantı uçları

PIC18F452, aşağıda belirtilen fiziksel ve donanımsal özelliklere sahiptir:

- **Çalışma Frekansı:** DC-40Mhz
- **Program Hafızası:** 32 Kbayt yada 16384 komut

- **Veri Hafızası:** 1536 bayt
- **Veri EEPROM Hafızası:** 256 bayt
- **Kesme Kaynakları:** 18 adet
- **G/Ç portları:** Bağlantı noktası A,B,C,D,E
- **Zamanlayıcı:** 4 adet
- **Yakala/Karşılaştır/PWM modülü:** 2 adet
- **Seri İletişim:** MSSP, Adreslenebilir USART
- **Paralel İletişim:** PSP
- **10 bit ADC Modülü:** 8 giriş kanalı
- **Sıfırlayıcılar:** POR, BOR, RESET Komutu, Yığın Dolu, Yığın boş (PWRT,OST)
- **Programlanabilir Düşük Gerilim Saptayıcı:** Var
- **Programlanabilir Gerilim Düşmesi Sıfırlaması:** Var
- **Komut Seti:** 75 komut
- **Paketler:** 40-bacak DIP, 44-bacak PLCC, 44-bacak TQFP

PIC18F452, PORTA, PORTB, PORTC, PORTD ve PORTE olmak üzere 5 adet giriş/çıkış (G/Ç) portuna sahiptir. PORTA 7-bit, PORTB, PORTC ve PORTD 8'er bit ve PORTE 3-bit genişliğindedir. Dolayısıyla her birine farklı işlevler atanmış toplam 34 adet iki yönlü uç bulunmaktadır. Her bağlantı noktası, G/Ç uçlarının yönünü belirlemeye yarayan bir TRIS yazmacına sahiptir. Bir G/Ç ucunu giriş olarak yapılandırmak için TRIS yazmacının ilgili biti 1, çıkış olarak yapılandırmak için 0 yapılır. G/Ç uçları birden fazla işlevi yerine getirecek şekilde çoğullanmıştır. İlgili yazmaç ve bayraklar yardımıyla kullanılmak istenen işlev etkinleştirilebilir. Ancak herhangi bir kararsızlık durumuyla karşılaşmamak için kullanılmayan işlevler edilgin durumda tutulmalıdır (Microchip, 2007).

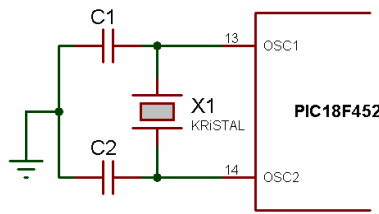
PIC18F452 aşağıda belirtilen 8 farklı osilatör modunda çalışabilir.

- **LP** Düşük güçlü kristal
- **XT** Kristal/Rezonatör
- **HS** Yüksek hızlı kristal/rezonatör
- **HS+PLL** Yüksek hızlı kristal/rezonatör ve PLL etkin

- **RC** Harici direnç kondansatör
- **RCIO** Harici direnç kondansatör ve G/Ç bacağı etkin
- **EC** Harici saat
- **ECIO** Harici saat ve G/Ç bacağı etkin

Osilatör modu, mikrodenetleyicinin programlanması sırasında kullanılan kristal ya da rezonatör frekansına göre 3 yapılandırma biti (FOOSC2,FOOSC1 ve FOOSC0) kullanılarak ayarlanır.

XT, LP, HS ya da HS+PLL modlarında, mikrodenetleyicinin OSC1 ve OSC2 uçlarına bir kristal yada rezonatör bağlanır. Kristal kullanıldığında OSC1 ve OSC2 uçları ile şase arasına, kapasitesi kristal frekansına bağlı olarak seçilen kondansatörler bağlanır. 4 MHz ve üzerindeki frekanslarda kristal kullanılıyorsa 15-33pF arası kondansatör kullanılabilir. Şekil 4.7’de PIC18F452 osilatör bağlantısı görülmektedir.

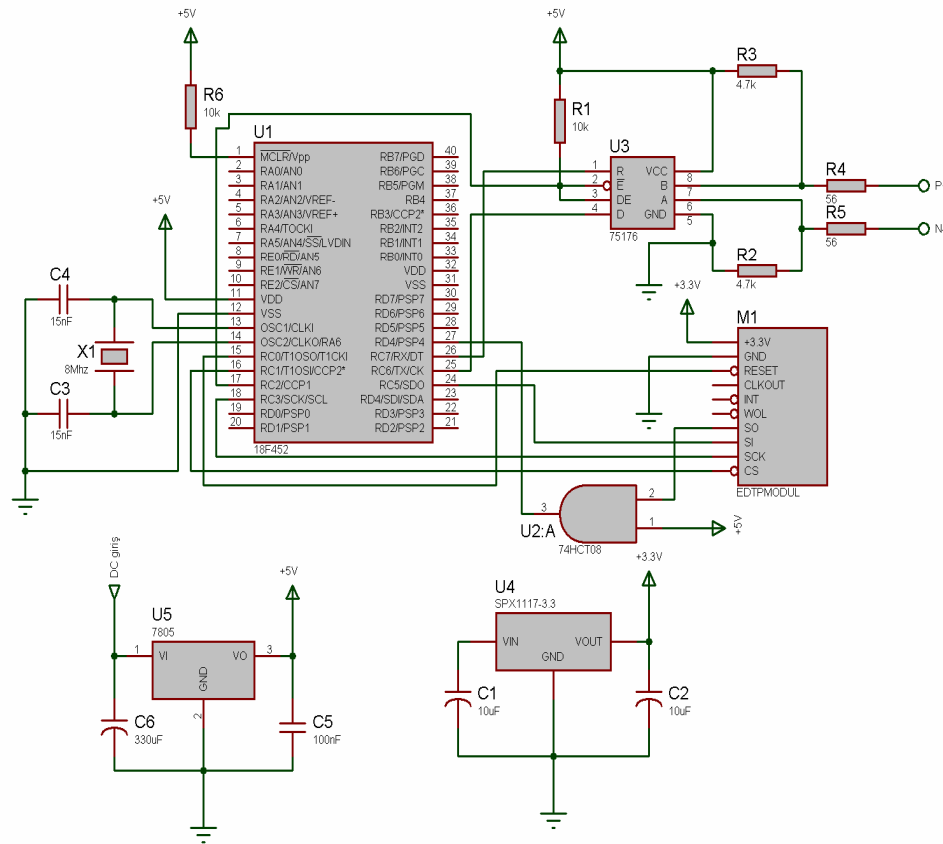


Şekil 4.7 PIC18F452 osilatör bağlantısı

PIC18F452 içerisinde bulunan ve etkinliği seçilebilen bir PLL devresi, kristal osilatör frekansının 4 katının saat frekansı olarak kullanılabilmesini sağlar. 10 Mhz’lik bir giriş saat frekansı için dâhili saat frekansı 40 Mhz’e yükselecektir. Yüksek frekanslı kristallerin neden olduğu Elektromanyetik Girişimden (EMI, Electromagnetic Interference) kaygılanan tasarımcılar için kullanışlı bir özelliktir.

PLL, osilatör yapılandırma bitleri HS modu için programlandığında etkinleştirilebilir. Eğer diğer modlar için programlanmış ise PLL etkinleşmez ve sistem saati doğrudan OSC1’den gelir.

Şekil 4.8’de tasarlanan arayüz devresinin şeması görülmektedir. Ethernet modülü blok halinde gösterilmiştir.



Şekil 4.8 Arayüz devre şeması

Seri İletişim:

Mikrodenetleyicide seri iletişim, tek bir hat üzerinden gönderilecek ya da alınacak olan verinin bitlerinin belli bir iletişim kuralı esasına göre gönderilmesi veya alınması işlemidir. Seri veri iletişimde eşzamanlı ve eşzamansız olmak üzere iki yöntem kullanılmaktadır. Eşzamanlı iletişimde bir veri bloğu saat pulsleri ile eşzamanlanarak iletilirken, eşzamansız iletişimde bir bayt'lık veri belirlenmiş zaman dilimlerinde iletilir. Her iki iletişim yöntemi donanımsal ve yazılımsal olarak gerçekleştirilebilir. PIC18F452 donanımsal seri iletişim için MSSP ve USART olmak üzere iki adet modül içerir.

MSSP modülü:

Ana Eşzamanlı Seri Bağlantı Noktası (MSSP, Master Synchronous Serial Port) modülü çevresel birimler veya diğer mikrodenetleyiciler ile seri olarak haberleşmeyi sağlayan bir arabirimdir. MSSP modülünün SPI ve I²C olarak iki tip çalışma modu bulunmaktadır.

SPI (Seri çevresel arayüz) modu 8-bit uzunluğundaki verinin aynı anda eşzamanlı olarak alınması ve gönderilmesine olanak tanır. SPI modunda haberleşme temel olarak 3 bacak ile sağlanır:

- **SDO** (Seri Veri Çıkışı)
- **SDI** (Seri Veri Girişi)
- **SCK** (Seri Saat)

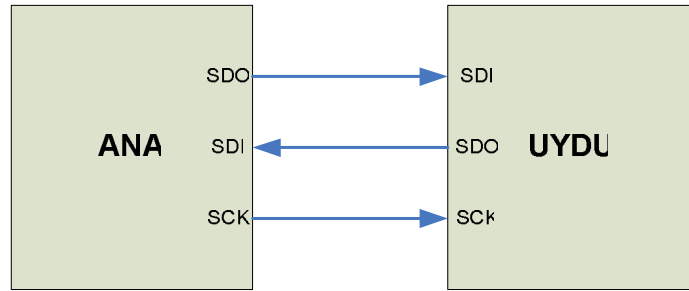
MSSP modülü, SPI modu için SSPCON1, SSPSTAT, SSPBUF ve SSPSR olmak üzere 4 yazmaca sahiptir. SSPCON1 ve SSPSTAT, SPI modu için denetim ve durum yazmaçlarıdır. SSPSR, veriyi içe ya da dışa kaydırmak için kullanılan kaydırmalı yazmaçtır. SSPBUF ise veri baytları için tampon yazmacıdır.

SPI modunda işlem yapabilmek için birkaç ayarın yapılması gereklidir. Bu ayarlar SSPCON1<5:0> ve SSPSTAT<7:6> denetim bitleri programlanarak yapılabilir. Bu denetim bitleri aşağıda belirtilen durumlara izin verir:

- Ana(master) modu (SCK saat çıkışı)
- Uydu(slave) modu (SCK saat girişi)
- Saat kutbu (SCK'nın boştaki hali)
- Veri girişi örnekleme fazı (Veri çıkış zamanının ortası ve sonunda)
- Saat kenarı (SCK'nın düşen ya da yükselen kenarında veri aktar)
- Saat hızı (Yalnız ana modunda)
- Uydu seçme modu (Yalnız uydu modunda)

Seri portu etkinleştirmek için öncelikle SSPEN biti(SSPCON1<5>) 1 yapılmalıdır. SPI modunun sıfırlamak ya da yeniden yapılandırmak için SSPEN biti 0 yapılmalı, SSPCON yazmacı yeniden düzenlenmeli ve SSPEN tekrar 1 yapılmalıdır. Bu SDI, SDO, SCK ve SS bacaklarını seri bağlantı noktası uçları olarak yapılandırır. Seri bağlantı noktası işlevinin kullanılabilmesi için SDO bacağı çıkış, SCK bacağı ana/uydu moduna göre çıkış ya da giriş, SS bacağı giriş olarak kullanıcı tarafından yapılandırılmalıdır. SDI bacağının yönü SPI modülü tarafından belirlenir.

SPI veri iletişimde birimlerden biri ana diğeri uydu modunda çalışır. Şekil 4.9’da SPI veri iletişimde ana uydu bağlantısı görülmektedir. Ana birim veri aktarımını SCK işareti göndererek başlatır. Programlanan saat kenarlarında her iki kaydırmalı yazmaçtan veri dışarı kaydırılır ve saatin karşı kenarında mandallanır. Her iki birimin aynı anda veri gönderip alabilmesi için aynı saat kutbuyla programlanması gerekir.



Şekil 4.9 SPI veri iletişimde ana/uydu bağlantısı

USART modülü:

Evrensel Eşzamanlı Eşzamansız Alıcı Verici (USART,) modülü verilerin eşzamanlı ve eşzamansız iletişimi için kullanılan ve belli bir protokol dâhilinde işlem yapan bir birimdir. USART kişisel bilgisayarlar gibi çevresel birimler ile iletişim yapabilen tam çift yönlü eşzamansız sistem olarak yapılandırılabilir ya da seri EEPROM, ADC veya DAC gibi çevresel birimler ile iletişim kurabilen yarı çift yönlü eşzamanlı sistem olarak yapılandırılabilir. USART aşağıdaki modları destekler:

- Eşzamansız (tam çift yönlü)
- Eşzamanlı-Ana (yarı çift yönlü)
- Eşzamanlı-Uydu (yarı çift yönlü)

RC6/TX/CK ve RC7/RX/DT bacaklarını USART olarak yapılandırmak için:

- SPEN biti (RCSTA<7>) 1 yapılmalı,
- TRISC<6> 0 yapılmalı,
- TRISC<7> 1 yapılmalıdır.

USART modülü iletişim denetimi için TXSTA ve RCSTA yazmaçlarına sahiptir.

Bit/s olarak ifade edilen seri iletişim hızına Baud Oranı adı verilir. USART modülünün iletişim hızı Baud Oranı Üretici (BRG) tarafından belirlenir. SPBRG

yazmacı bağımsız çalışan 8-bitlik bir zamanlayıcının periyodunu denetler. Eşzamansız moda BRGH biti (TXSTA<2>) ayrıca baud oranını denetler. Eşzamanlı modda bu bit kullanılmaz.

Eşzamanlı modda istenen baud oranı,

$$Baud = \frac{F_{osc}}{4(X + 1)}$$

eşitliği ile hesaplanmaktadır. Eşzamansız modda istenen baud oranı,

BRGH biti 0 iken

$$Baud = \frac{F_{osc}}{64(X + 1)}$$

eşitliği ile, BRGH biti 1 iken

$$Baud = \frac{F_{osc}}{16(X + 1)}$$

eşitliği ile hesaplanır. Eşitliklerde;

F_{osc} : Osilatör frekansı,

X: SPBRG yazmacı içindeki değerdir.(0..255)

Baud: İstenilen baud oranıdır.

Eşzamansız modda USART standart NRZ formatını (bir START biti, sekiz ya da 9 veri biri ve bir STOP biti) kullanır. En yaygın veri formatı 8-bittir. Yonga üzerinde bulunan 8-bit baud oran üretici osilatörden standart baud oranı türetmek için kullanılabilir. USART ilk önce LSB'yi iletir ya da alır. USART alıcı ve verici işlevsel olarak birbirinden bağımsızdır ancak aynı veri formatını ve baud oranını kullanır. Eşlik, donanım tarafından desteklenmez ancak yazılım tarafından gerçekleştirilebilir. Eşzamansız mod uykusu sırasında durdurulur. Eşzamansız mod SYNC biti (TXSTA<4>) sıfırlanarak seçilir.

Bir eşzamansız iletimi kurmak için;

- SPBRG yazmacı uygun baud oranı değeriyle yüklenir. Yüksek hızlı baud oranı istenirse BRGH 1 yapılır.
- SYNC biti 0 yapılarak ve SPEN biri ayarlanarak eşzamansız seri bağlantı noktası etkinleştirilir.
- Kesmeler kullanılacaksa TXIE yetki biti 1 yapılır.

- 9-bit iletim isteniyorsa TX9 iletim biti 1 yapılır. Adres/veri biti olarak kullanılabilir.
- TXEN biti 1 yapılarak iletim etkileştirilir. Aynı zamanda TXIF biti 1 yapılmalıdır.
- 9-bit iletim seçildiyse 9. bit TX9D bitine yüklenmelidir.
- Veri TXREG yazmacına yüklenir (İletim başlar).

Bir eşzamansız alımı kurmak için ise;

- SPBRG yazmacı uygun baud oranı değeriyle yüklenir. Yüksek hızlı baud oranı istenirse BRGH 1 yapılır.
- SYNC biti 0 yapılarak ve SPEN biti ayarlanarak eşzamansız seri bağlantı noktası etkinleştirilir.
- Kesmeler kullanılacaksa RCIE yetki biti 1 yapılır. (Alım tamamlandığında RCIF bayrağı 1 olacak ve RCIE yetki biti 1 yapıldığından bir kesme üretilecektir)
- 9-bit alım isteniyorsa RX9 biti 1 yapılır.
- CREN biti ayarlanarak alım etkinleştirilir.
- 9. biti almak için RCSTA yazmacı okunur ve alım sırasında herhangi bir hata meydana gelirse tespit edilir.
- RCREG yazmacını okunarak alınan 8-bit veri okunur.
- Herhangi bir hata meydana geldiğinde CREN yetki biti sıfırlanarak hata temizlenir.
- Kesmeler kullanılıyorsa INTCON yazmacı içindeki GIE ve PEIE bitlerinin 1 olduğundan emin olunmalıdır.

4.1.3 RS485 Alıcı-Verici

RS485, RS232 gibi bilgisayarlar ve cihazlar için bir seri iletişim yöntemidir. İki hat, yarı-çift yönlü, çok noktalı seri bağlantı sağlayan OSI modeli fiziksel katmanında elektriksel bir tanımlamadır. Birçok düğümün birbiri ile iletişeceği veri toplama ve denetim uygulamalarında yaygın bir şekilde kullanılan bir iletişim arabirimidir (Anonim A).

RS232'nin ana sorunlarından biri işaret hatları üzerinde gürültü bağıışıklığı yoksunluğudur. Alıcı ve verici, veri ve tokalaşma hatlarının gerilimlerini bir ortak

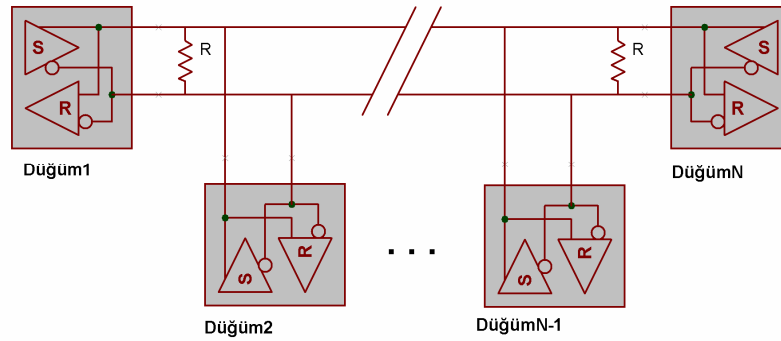
sıfır hattı ile karşılaştırır. Toprak seviyesinden kaymalar kötü sonuçlar doğurabilir. Gürültü kolayca yükselebilir ve bu durumda iletişim hızı ve maksimum aralığın her ikisi sınırlanır. Bu nedenle RS232 arabiriminin eşik seviyesi nispeten yüksektir ($\pm 3V$ 'tur).

RS485'te RS232'nin aksine bir işaret referansı olarak bir ortak sıfır yoktur. RS485 alıcı ve vericinin toprak seviyesindeki birkaç volt fark herhangi bir sorun oluşturmamaktadır. RS485 işaretleri kayardır ve her bir işaret P+ ve N- hatları üzerinden gönderilir. RS485 alıcı bir işaret hattı üzerindeki mutlak gerilim seviyesinin yerine her iki hat arasındaki gerilim farkını karşılaştırır. Bu düzen gayet iyi çalışır ve bir ortak iletişim sorunu kaynağı olan toprak döngülerinin oluşmasını engeller. En iyi sonuç P+ ve N- hatlarının kabloları bükülü olursa alınır. Yüksek gürültü bağışıklığı gerekiyorsa korumalı bükülü çift (STP) ve yapraklı bükülü çift (FTP) ağ kabloları kullanılabilir. Fark işaretleri ve bükülü çift RS485'in, RS232'nin sağladığından daha uzun iletişim mesafelerinde haberleşme yapılabilmesine olanak tanır. RS232'de iletişim mesafesi en fazla 15m iken RS485'ninki 1200m'dir. Farksal sinyal hatları ayrıca farksal olmayan bağlantılarda sağlanandan daha yüksek bit oranlarına izin verir. RS232 20kbit/s hızı desteklerken RS485 12m'ye kadar 35Mbit/s, 1200m'ye kadar da 100kbit/s hızları desteklemektedir (Anonim G).

RS485 aynı ağ içindeki çok sayıda alıcı ve verici arasında ağ kurma yeteneğine sahip arabirimdir. $12K\Omega$ 'luk giriş direnci ile varsayılan RS485 alıcılar kullanıldığında ağa 32 adet aygıt bağlanabilir. Direnç değeri daha da artırılarak RS485 girişleri düğüm sayısı 256'ya çıkartılabilir. Eğer sayı daha da artırılsın istenirse RS485 tekrarlayıcılar kullanmak gerekir (Anonim G).

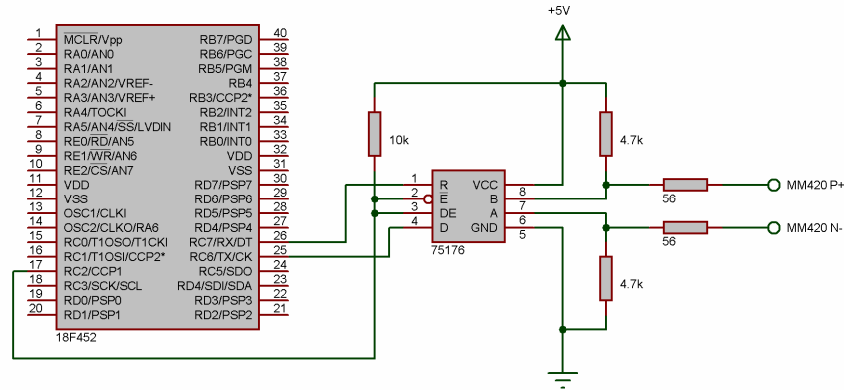
RS485'in, bilimsel ve teknik uygulamalarda bilgisayarlar, PLC'ler, mikrodenetleyiciler ve akıllı algılayıcılar için popüler olmasının sebebi olarak yazılımsal olarak kolay gerçekleştirilebilmesi, akıllı ağ donanımı gerektirmemesi, herhangi bir veri iletişim kuralı belirtmemesi ve önermemesi sayılabilir.

Şekil 4.10'da RS485'in genel ağ topolojisi gösterilmiştir. Çok noktalı RS485 ağına N tane düğüm bağlanır. Daha yüksek hız ve uzun hat için yansımaları önleyen sonlandırma dirençlerinin hatların her iki ucunda olması gereklidir. Her iki uçta 100Ω 'luk dirençler uygundur (Anonim G).



Şekil 4.10 RS485 genel ağ topolojisi

RS485, Profibus, Modbus, gibi pek çok iyi bilinen standart için elektriksel katman olarak kullanılır. Bu nedenle tez çalışmasında UDA devresinde iletişim yöntemi olarak RS485 kullanılmıştır. UDA devresinde RS485 alıcı-verici olarak Texas Instruments firmasının ürettiği SN75176A tümleşik devresi kullanılmıştır. SN75176A tümleşik devresi iki yönlü bir farksal hat alıcı vericisidir. Şekil 4.11'de SN75176A tümleşik devresinin mikrodenetleyici ile bağlantısı gösterilmiştir.



Şekil 4.11 SN75176A'nın mikrodenetleyici ile bağlantısı

4.1.4 Asenkron Motor

Asenkron motorlar endüstriyel denetim, otomasyon ve cihazlar için yaygın olarak kullanılan motorlardır. Sağlam, güvenilir ve uzun ömürlüdürler. Endüstriyel uygulamaların çoğu, motor hızının değiştirilebilmesini gerektirir. Bu işlevi yerine getirmek için mekanik sistemler kullanılabileceği gibi elektronik cihazlar da kullanılmaktadır. Bu elektronik motor sürücüler, motor hızını denetlemek dışında

motorun dinamik ve kararlı hal karakteristiklerini geliřtirmek, sistemin ortalama g¼c t¼knetimi ve motorun g¼r¼lt¼ ¼retimi azaltmak gibi ilave yetenekleri de vardır.

Asenkron motor denetimi, motorun doęrusal olmayan ¼zellikleri nedeniyle karmařıktır ve farklı ¼st¼nl¼klere sahip bir¼ok y¼ntem ile yapılabilmektedir. Bařlıca denetim y¼ntemleri; Deęiřken gerilim-deęiřken frekans (sabit V/f oranı ya da Volt-Hertz) denetim, Vekt¼r denetim ve Doęrudan Tork denetimidir.

A¼ık d¼ng¼ i¼inde en yaygın hız denetim y¼ntemi sabit V/f denetimdir. Bu y¼ntem konum denetimine gerek duyulmayan ya da y¼ksek doęrulukla hız denetimine gerek duyulan uygulamalar i¼in en uygundur.

Vekt¼r denetim, ger¼ek zamanlı hız hesaplama i¼in karmařık algoritmalar kullanır. Geribesleme devresi barındırdıęı i¼in daha pahalıdır ancak sabit V/f denetime nazaran daha iyi tork yanıtı, DC s¼r¼ře yakın bařarım ve doęru hız denetimi sunar.

Doęrudan tork denetimi geribeslemesiz, uyarlamalı motor modeli kullanan bir y¼ntemdir. Hızlı yanıt, daha az mekanik hata ve DC s¼r¼ře yakın bařarım gibi ¼st¼nl¼klerine karřın y¼ksek tork ve akı dalgalanması gibi sakıncalara sahiptir.

4.1.5 Motor S¼r¼c¼

End¼stride asenkron motor s¼r¼c¼s¼ Siemens ¼r¼nleri yaygın olarak kullanılmaktadır. Siemens farklı uygulamalar i¼in farklı s¼r¼c¼ sınıfları sunmaktadır.

Genel olarak bu sınıflar;

- Micromaster Vector
- Micromaster
- Micromaster Eco
- Combimaster olarak sıralanabilir.

Micromaster Vector, genel ama¼lı uygulamalar i¼in y¼ksek bařarım sunan s¼r¼c¼ sınıfıdır. Micromaster ise daha basit uygulamalar i¼in ¼zellikleri azaltılmıř bir s¼r¼c¼ sınıfıdır. Micromaster Eco, ısıtma, havalandırma ve iklimlendirme uygulamaları i¼in ¼zel tasarlanmıř bir s¼r¼c¼ sınıfıdır. Combimaster sınıfı ise üzerine invert¼r yerleřtirilmıř bir ind¼ksiyon motorudur.

Bu uygulama, Micromaster 420 s¼r¼c¼s¼ kullanılarak yapılmıř olmasına raęmen dięer Siemens s¼r¼c¼leri i¼in uyarlanabilir.

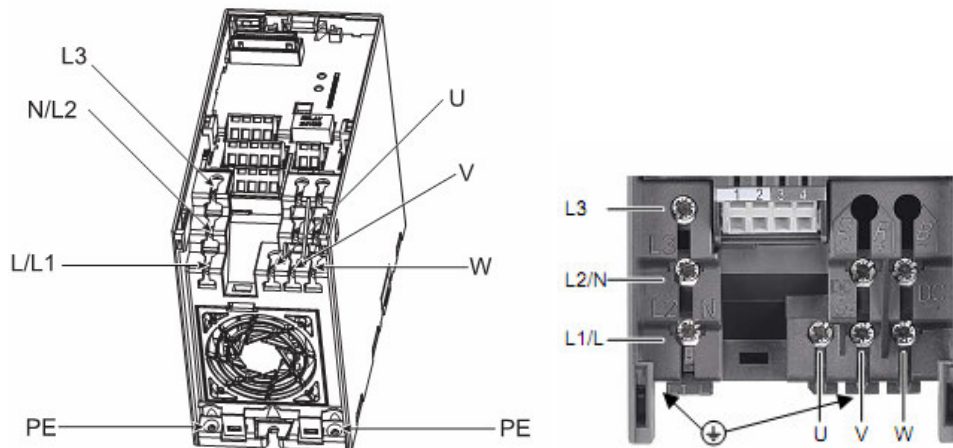
Micromaster 420 cihazı, üç fazlı AC motorların devirlerini kontrol etmek için kullanılan bir dizi frekans invertörüdür. 120 W'lık tek fazlı girişten 11 kW'lık üç fazlı girişe kadar değişik özellikte modelleri mevcuttur (Anonim E).

Micromaster 420, gelişmiş motor kontrol uygulamaları için seviyelere ayrılmış kapsamlı parametre listesine sahiptir. Bu parametrelerin varsayılan değerleri pek çok uygulama için uygun olmasına rağmen istendiğinde panel üzerinden doğrudan ya da seri arayüz ile değiştirilebilir. P0003 parametresi vasıtasıyla seçilebilen Standart, Genişletilmiş, Uzman ve Bakım olmak üzere kullanıcı tarafından erişilmesi mümkün olan dört seviye bulunmaktadır (Anonim E).

- **P0003=0** : Kullanıcı tarafından tanımlanan parametre listesidir.
- **P0003=1, Standart** : En sık kullanılan parametrelere erişim sağlar. Varsayılan seviyedir.
- **P0003=2 , Genişletilmiş**: Sürücü giriş-çıkış işlevlerine genişletilmiş erişim sağlar.
- **P0003=3, Uzman**: Sadece uzmanların kullanımı içindir.
- **P0003=4, Bakım** : Sadece yetkili bakım personelinin kullanımı içindir ve şifre korumalıdır.

4.1.5.1 Motor sürücü kurulumu

Micromaster 420 sürücüsü üç fazdan beslenilebileceği gibi tek fazdan da beslenilebilir. Micromaster 420 sürücüsünün güç besleme ve motor bağlantı uçları Şekil 4.12'de gösterilmiştir. Şekil 4.12'de L3, L2 ve L1 uçları üç faz besleme için L ve N uçları tek faz besleme için kullanılır. PE ucu sürücüyü topraklamak için kullanılır. U, V ve W uçları asenkron motor sargılarına bağlanacak 3-faz uçlarıdır. Micromaster 420 sürücünün denetim terminalleri ve işlevleri Şekil 4.13'de gösterilmiştir (Anonim E).



Şekil 4.12 MM420 güç terminalleri

Terminal	Gösterim	İşlev
1	-	+10V çıkış
2	-	0V çıkış
3	ADC+	Analog giriş(+)
4	ADC-	Analog giriş(-)
5	DIN1	Sayısal giriş 1
6	DIN2	Sayısal giriş 2
7	DIN3	Sayısal giriş 3
8	-	+24V yalıtılmış çıkış
9	-	0V yalıtılmış çıkış
10	RL1-B	Sayısal çıkış
11	RL1-C	Sayısal çıkış
12	DAC+	Analog çıkış(+)
13	DAC-	Analog çıkış(-)
14	P+	RS485 bağlantı noktası
15	N-	RS485 bağlantı noktası



Şekil 4.13 MM420 denetim terminalleri

4.1.5.2 Seri arayüz

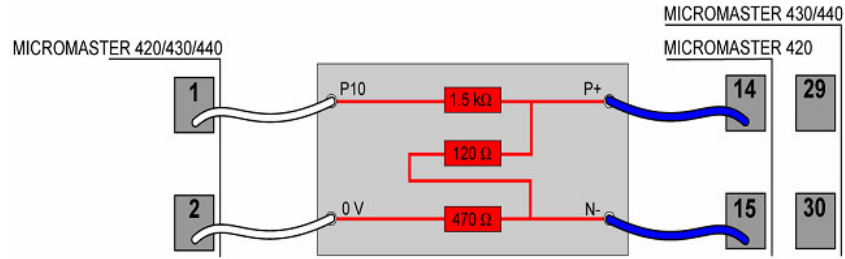
Tüm Siemens sürücüler standart olarak bir seri bağlantı arayüzü içerir. Seri bağlantı arayüzü kullanmanın başlıca üstünlükleri vardır:

- Kabloleme oldukça azaltılabilir.

- Denetim işlevleri yeniden bağlantı yapılmadan değiştirilebilir.
- Parametreler arayüz tarafından ayarlanabilir ve değiştirilebilir.
- Başarım sürekli bir şekilde izlenebilir ve denetlenebilir.

Seri arayüz, endüstriyel uygulamalar için tasarlanmış bir RS485 iki-hat bağlantı kullanır. En fazla 31 tane sürücü bir tek RS485 hattına bağlanabilir. Tüm sürücülere özel olarak erişim yapılabileceği gibi hepsine birden de erişim yapılabilir. Uydu (slave) olarak davranan sürücülerin denetlenebilmesi için RS485 hattına bir tane ana (master) denetleyicinin bağlanması gereklidir.

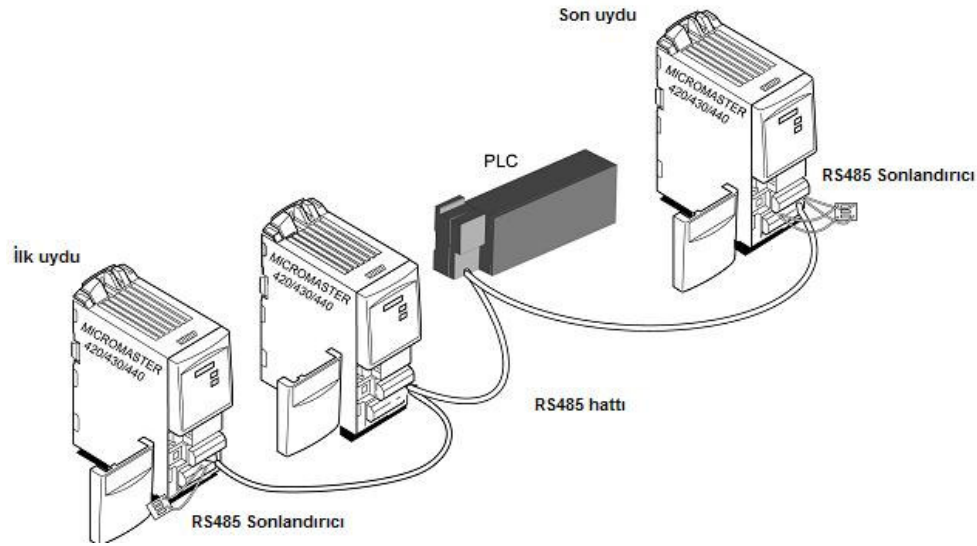
Micromaster 420 sürücüsünü RS485 iletişim ile kullanma, hattın (P+ ve N- arasındaki) her iki ucunda uygun sonlandırma ve hattın en az bir ucunda doğru yükleme ve çökeltme direnci gerektirir (örneğin P+'dan P10'a ve N-'den 0V'a). Micromaster hat üzerindeki son uydunya ve hat üzerinde hiçbir yükleme ve çökeltme direnci yok iken RS485 sonlandırıcı Şekil 4.14'te olduğu gibi bağlanmalıdır (Anonim E).



Şekil 4.14 RS485 sonlandırıcı

Micromaster hat üzerindeki ilk uydunya (Şekil 4.15'te soldaki) RS485 sonlandırıcı sadece P+ ve N- uçları kullanılarak bağlanabilir. Hat son sürücü tarafından beslenecektir (Anonim E).

RS485 iletişim başlatılmadan önce yükleme ve çökeltme dirençleri için besleme sağlanmalıdır (Anonim E).



Şekil 4.15 RS485 hattı sonlandırma

4.2 Yazılım

UDA sistemi, mikrodeneleyici yazılımı ve applet arayüz yazılımı olmak üzere iki yazılım içermektedir. Uygun kütüphanelere sahip olmasından dolayı mikroElektronika firmasının PIC C derleyicisi olan mikroC, UDA mikrodeneleyici yazılımı için uygun görülmüştür. UDA devresinin kullanıcı ile bağlantısını sağlayan arayüz applet'ini geliştirmek için ise Borland firmasına ait Java geliştirme ortamı olan JBuilder yazılımı uygun görülmüştür.

4.2.1 Mikrodeneleyici Yazılımı

PIC18F452 Ethernet modülü ile haberleşmek için SPI, Micromaster 420 sürücüsü ile haberleşmek için USART kullanır. Ethernet modülü ile iletişim için SPI Ethernet kütüphanesi kullanılmıştır. MM420 ile haberleşme için USART kütüphanesi USS protokolüne uyarlanarak kullanılmıştır.

SPI Ethernet kütüphanesi aşağıdaki fonksiyonlara sahiptir:

SPI_Ethernet_Init :

İlkörnek	<code>void SPI_Ethernet_Init(unsigned char *resetPort, unsigned char resetBit, unsigned char *CsportPtr, unsigned char Csbit, unsigned char *mac, unsigned char *ip, unsigned char fullDuplex);</code>
Dönüş	Yok
Açıklama	SPI ve ENC denetçisini başlatır. Bu fonksiyon hafıza kısa geldiğinde yardımcı bağ

	<p>için iki kısma ayrılmıştır.</p> <p>resetPort : Sıfırlama bacağı portuna işaretçi</p> <p>resetBit : resetPort üzerindeki sıfırlama bacağı sayısı</p> <p>CSPort : CS bacağı portuna işaretçi</p> <p>mac : 6 karakterlik MAC adres dizisine işaretçi</p> <p>ip : 4 karakterlik IP adres dizisine işaretçi</p> <p>fullDuplex : Yarı çift yönlü iletişim için SPI_Ethernet_HALFDUPLEX yada tam çift yönlü iletişim için SPI_Ethernet_FULLDUPLEX.</p>
Gereksinimler	SPI Ethernet başlatılmadan önce SPI_Init() çağrılmalıdır.
Örnek	SPI_Ethernet_Init(&PORTC, 0, &PORTC, 1, myMacAddr, myIpAddr, SPI_Ethernet_FULLDUPLEX)

SPI_Ethernet_doPacket

İlkörnek	<code>void SPI_Ethernet_doPacket();</code>
Dönüş	Yok.
Açıklama	Erişilebilir ise gelen bir paketi işler. Bu fonksiyon public'tir.
Gereksinimler	Bu fonksiyon çağrılmadan önce SPI_Ethernet_Init fonksiyonu çağrılmalıdır.
Örnek	<code>SPI_Ethernet_doPacket();</code>

SPI_Ethernet_putByte

İlkörnek	<code>void SPI_Ethernet_putByte(unsigned char v);</code>
Dönüş	Yok.
Açıklama	v= depolanacak değer Mevcut EWRPT ENC konumuna bir bayt depolar. Bu fonksiyon public'tir.
Gereksinimler	Bu fonksiyon çağrılmadan önce SPI_Ethernet_Init fonksiyonu çağrılmalıdır.
Örnek	<code>SPI_Ethernet_putByte(0xA0);</code>

SPI_Ethernet_getByte

İlkörnek	<code>unsigned char SPI_Ethernet_getByte();</code>
Dönüş	Belirtilen adresin bayt değeri
Açıklama	Mevcut ERDPT ENC konumundan sonraki baytı alır. Bu fonksiyon public'tir.
Gereksinimler	Bu fonksiyon çağrılmadan önce SPI_Ethernet_Init fonksiyonu çağrılmalıdır.
Örnek	<code>b=SPI_Ethernet_getByte();</code>

SPI_Ethernet_UserTCP

İlkörnek	unsigned int SPI_Ethernet_UserTCP(unsigned char *remoteHost, unsigned int remotePort, unsigned int localPort, unsigned int reqLength);
Dönüş	HTTP yanıtının bayt sayısını ya da gönderim yoksa 0 döndürür.
Açıklama	Bu işlev kütüphane tarafından çağrılır. Kullanıcı HTTP isteğine peş peşe SPI_Ethernet_getByte() çağrıları yoluyla erişir. Kullanıcı peş peşe SPI_Ethernet_putByte() çağrıları yoluyla gönderim tamponuna veri koyar. İşlev HTTP isteği bayt sayısını ya da gönderim yoksa 0 döndürmelidir.
Gereksinimler	Bu fonksiyon çağrılmadan önce SPI_Ethernet_Init fonksiyonu çağrılmalıdır.
Örnek	Bu fonksiyon kütüphane tarafından çağrılır ve kullanıcı kodu tarafından çağrılmamalıdır.

SPI_Ethernet_UserUDP

İlkörnek	unsigned int SPI_Ethernet_UserUDP(unsigned char *remoteHost, unsigned int remotePort, unsigned int destPort, unsigned int reqLength);
Dönüş	UDP yanıtının bayt sayısını ya da gönderim yoksa 0 döndürür.
Açıklama	Bu işlev kütüphane tarafından çağrılır. Kullanıcı UDP isteğine peş peşe SPI_Ethernet_getByte() çağrıları yoluyla erişir. Kullanıcı peş peşe SPI_Ethernet_putByte() çağrıları yoluyla gönderim tamponuna veri koyar. İşlev UDP yanıtı bayt sayısını ya da gönderim yoksa 0 döndürmelidir.
Gereksinimler	Bu fonksiyon çağrılmadan önce SPI_Ethernet_Init fonksiyonu çağrılmalıdır.
Örnek	Bu fonksiyon kütüphane tarafından çağrılır ve kullanıcı kodu tarafından çağrılmamalıdır.

Yukarıda sözü edilen işlevler kullanılmadan önce ENC28J60 Ethernet denetçisi için bir MAC adresi ve IP adresi tanımlanmalıdır. Aşağıda MAC ve IP adres tanımlama örneği gösterilmiştir.

```
unsigned char myMacAddr[6]={0x00,0x14,0xA5,0x76,0x19,0x3F};
```

```
unsigned char myIPAddr[4]={192,168,20,60};
```

ENC28J60 denetçisi statik IP adresi ile çalıştığı için bağlandığı Ethernet ağına uygun ve benzersiz bir IP atanmasına dikkat edilir.

USART kütüphanesi aşağıdaki işlevleri sunmaktadır:

Usart_Init

İlkörnek	void Usart_Init(const long baud_rate);
Açıklama	USART donanım modülünü istenilen baud oranında başlatır. Belirli Fosc'ler için

	izin verilen baud_oranları için yonganın veri sayfasına başvurun. Desteklenmeyen bir baud oranı belirlerseniz derleyici bir hata bildirecektir. Bu işlev diğer USART kütüphanesi işlevleri kullanılmadan önce çağrılmalıdır.
Gereksinimler	USART donanımlı bir PIC mikrodenetleyici.
Örnek	Usart_Init(2400);

Usart_Data_Ready

İlkörnek	char Usart_Data_Ready(void);
Dönüş	Veri hazırsa 1 değilse 0 döndürür.
Açıklama	Verinin gönderime hazır olup olmadığını sınamak için kullanılan işlevdir.
Gereksinimler	Bu işlev çağrılmadan önce USART donanım modülü başlatılmalı ve iletişim kurulmalıdır.
Örnek	<pre>int receive; if(Usart_Data_Ready()) receive=Usart_Read(); //Veri hazırsa oku ve receive değişkenine kaydet.</pre>

Usart_Read

İlkörnek	char Usart_Read(void);
Dönüş	Alınan baytı döndürür. Bayt alınmamışsa 0 döndürür.
Açıklama	İşlev USART vasıtasıyla bir bayt alır. Öncelikle verinin hazır olup olmadığını sınamak için Usart_Data_Ready işlevini kullanın.
Gereksinimler	Bu işlev çağrılmadan önce USART donanım modülü başlatılmalı ve iletişim kurulmalıdır.
Örnek	<pre>int alveri; if(Usart_Data_Ready()) alveri=Usart_Read(); //Veri hazırsa oku ve alveri değişkenine kaydet.</pre>

Usart_Write

İlkörnek	char Usart_Write(char data);
Açıklama	İşlev USART yoluyla bir bayt(data) gönderir.
Gereksinimler	Bu işlev çağrılmadan önce USART donanım modülü başlatılmalı ve iletişim kurulmalıdır.
Örnek	<pre>int gonveri; Usart_Write(gonveri); //gonveri verisi USART üzerinden gönderilir</pre>

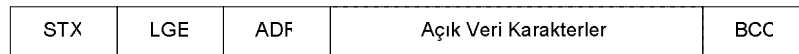
4.2.1.1 Evrensel seri arayüz protokolü

Evrensel seri arayüz protokolü (USS) bir seri hat üzerinden iletişim için ana-uydu ilkesine göre erişim tekniği tanımlar. Elektriksel ve mekaniksel arayüz özellikleri (donanım) protokol özelliklerinin bir parçası değildir.

USS protokolü ile sürücüye komut vermek, parametre değerlerini okumak/değiřtirmek ya da motor karakteristiklerini izlemek için telegamlar kullanılır.

Telegram yapısı:

Her bir telegram STX karakteri (02h) ile başlar, onu uzunluk belirten karakter (LGE) ve adres baytı (ADR) izler. Açık veri karakterleri bu 3 bayt'tan sonra gelir. Telegram blok denetim karakteri (BCC) tarafından sonlandırılır (Anonim I) (Şekil 4.16).



Şekil 4.16 Telegram yapısı

Telegram uzunluğu deęiřkendir. Telegram uzunluğu ikinci telegram baytında belirlenmiştir. Yapılanıřa baęlı olarak sabit telegram uzunluğu belirlenebilir. Sabit telegram uzunlukları yerine farklı telegram uzunlukları hat üzerindeki her bir uydu için kullanılabilir. Bir telegramın maksimum toplam uzunluğu 256 bayttır. Toplam telegramın gerçek uzunluğu ilk iki karakter (STX ve ADR) sayılmadıęı için LGE'de belirtilen uzunluktan iki karakter daha uzundur. Sadece açık veri karakterleri (n tane), adres baytı (ADR) ve blok denetim karakteri (BCC) telegram uzunluęunu oluřturmaktadır. Telegram başına maksimum 252 bayt açık veri karakter aktarılabilir (Anonim I).

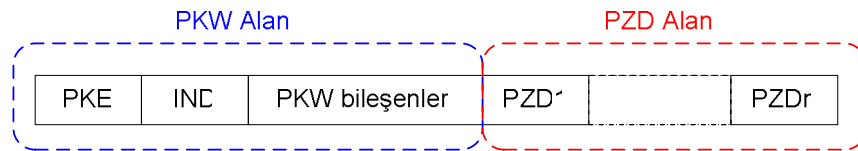
Deęiřken uzunluktaki telegramlarda açık veri karakterlerinin sayısı özel görevlere baęlıdır. Önceden belirlenmiş sabit uzunluktaki telegram aktarımında veri karakterlerinin sayısı sabittir.

Adres alanı bir bayt'tır ve uydu düęümün adresini içerir. Uydu düęümünün adresi (0-31) ikilik olarak adres alanın 0-4 bitlerin yazılır. Adres alanın 5. biti genel yayın bitidir. Bu bit 1 yapılırsa ileti hat üzerindeki tüm uydular tarafından alınır.

Adres alanının 6. biti 1 yapılırsa uydu düğüm aldığı telegramı aynen geri gönderir (ayna telegram). Ayna telegram ana/uydu arasındaki veri aktarımını sınamak için kullanılabilir.

BCC alanı telegramları geçerli kılmak için kullanılan bir bayt-boyutlu sağlama toplamıdır. Telegram içindeki her baytın ilk iki bayttan başlanarak birbiri ile ÖZEL-VEYA işlemine tabi tutulmasıyla elde edilen değer telegramın sağlama toplamıdır.

Açık veri alanı Şekil 4.17’de görüldüğü gibi PKW ve PZD olmak üzere iki alana ayrılmıştır. PKW alanı parametre değerlerini okuyup yazmak için kullanılır. PZD alanı sürücünün denetlenmesi ve izlenmesi için kullanılır.



Şekil 4.17 Açık veri bloğu

PKW alanının ilk iki sözcüğü (word) PKE ve IND, ana birim tarafından istenen göreve ilişkin ya da yanıt telegramının tipine ilişkin bilgi verir. Ayrıca telegramla erişilecek parametre numarasını (PNU) tanımlar. PKE sözcüğünün 12-15 bitleri görev yada yanıt tipi içi ayrılmıştır. 0-10 arası bitleri parametre numarası için ayrılmıştır. 11.bit daima sıfırdır. IND sözcüğünün 12-15 bitleri parametre numarası alanı uzantısıdır. 0-7 bitler indeks için ayrılmıştır. 8-11 bitler kullanılmaz.

Parametre numaralarının tamamı PKE ve IND sözcüklerinin ilgili bitleri ile tanımlanır. Parametre numaralarının karşılığı bulunurken parametre numarası 2000’e bölünür. Bölme işleminde elde edilen bölüm ilgili IND bitlerine, kalan ise PKE bitlerine ikilik düzene çevrilerek kaydedilir. Bölüm ilgili IND bitlerine yazılırken 15.bit LSB olacak şekilde işlem yapılır. Numarası 1999’a kadar olan parametreler için ilgili PKE bitlerine ikilik düzende parametre numarası (kalan ile aynı olacağından) doğrudan yazılır. Bölüm 0 olduğundan ilgili IND bitlerine 0 yazılır. Numarası 2000-3999 arasında olan parametreler için IND sözcüğünün 15. bitine 1 yazılır. Kalan yine ilgili PKE bitlerine yazılır. Daha büyük parametre numaraları içinde bu yöntem geçerlidir. Ancak MM420 3999’dan daha büyük parametrelere sahip değildir.

Örneğin, P0700 parametresi için ilgili PKE bitlerine 700’ün ikilik karşılığı olan 01010111100 (onaltılık 2BC) yazılır. İlgili IND bitlerine ise 0 yazılır. P2012

parametresi için ilgili PKE bitlerine 12'nin ikilik karşılığı olan 00000001101 (onaltılık 00C) yazılır. İlgili IND bitlerine ise 1 yazılır.

PKW bileşenleri genellikle PWE1 ve PWE2 adında iki sözcükten ibarettir. Erişilecek parametrenin değerini saklarlar. MM420'de tam sayı değerler, onlu değerler ve indeksli değerler gibi farklı parametre değer tipleri bulunmaktadır. Parametre değerleri, P2013 parametresinin ayarına ve parametre değer tipine bağlıdır. P2013 parametresi PKW alanının sabit mi değişken mi olacağını belirler.

Ana birim tarafından bir parametre değeri istenirse görev telegramının PWE1 ve PWE2 değerleri evirici tarafından göz ardı edilir.

P2013 kullanılarak PKW uzunluğu (3 yada 4 olarak) sabitlenirse ana birim daima PKW alanında ya 3 yada 4 sözcük gönderir (aksi halde uydu yanıt vermez). Aynı şekilde uydu yanıtının PKW alanı 3 yada 4 sözcük olacaktır. Değişken PKW uzunluğu için ana birim, PKW alanında görev için sadece gerekli sayıda sözcük gönderir. Yanıt telegramının uzunluğu da gerektiği kadar olacaktır.

PKE sözcüğünün 15–12. lu bitleri görev ya da yanıt kimliği için kullanılır. Örneğin parametre değeri okunacağı zaman görev kimliği 1 olarak gönderilir. Döndürülen yanıt kimliğinden de parametre değerinin uzunluğu bilinebilir. Örneğin çift sözcük parametre değeri döndürülecekse yanıt kimliği 2'dir. Tanımlı görev kimliği Çizelge 4.1'de, tanımlı yanıt kimlikleri ise Çizelge 4.2'de görülmektedir.

Çizelge 4.1 Tanımlı görev kimlikleri

Görev kimliği	Anlamı	Olumlu Yanıt Kimliği	Olumsuz Yanıt Kimliği
0	Görev yok.	0	-
1	Parametre değeri iste.	1, 2	7
2	Parametre değeri değiştir(sözcük)[yalnız RAM]	1	7,8
3	Parametre değeri değiştir(çift sözcük)[yalnız RAM]	2	7,8
4	Tanım bileşeni iste.	3	7
5	MM420 desteklemiyor.	-	-
6	Parametre değeri iste.(dizi) örn. İndeksli parametre	4,5	7
7	Parametre değeri değiştir.(dizi, sözcük) [Yalnız RAM]	4	7,8
8	Parametre değeri değiştir.(dizi, çift sözcük) [Yalnız RAM]	5	7,8
9	Dizi elemanlarının numarasını iste.	6	7
10	Ayrılmış.	-	-
11	Parametre değeri sakla.(dizi, çift sözcük)[RAM ve EEPROM]	5	7,8
12	Parametre değeri sakla.(dizi, sözcük)[RAM ve EEPROM]	4	7,8
13	Parametre değeri sakla.(çift sözcük)[RAM ve EEPROM]	2	7,8
14	Parametre değeri sakla.(sözcük)[RAM ve EEPROM]	1	7,8
15	MM420 desteklemiyor.	-	-

Çizelge 4.2 Tanımlı yanıt kimlikleri

Yanıt kimliği	Anlamı	Görev kimliği
0	Görev yok.	0
1	Parametre değeri aktar.(sözcük)	1, 2,14
2	Parametre değeri aktar(çift sözcük)	1,3,13
3	Tanım bileşeni aktar	4
4	Parametre değeri aktar(dizi, sözcük)	6,7,12
5	Parametre değeri aktar(dizi, çift sözcük)	6,8,11
6	Dizi eleman sayısını aktar.	9
7	Görev yürütülemedi(hata değeri ile)	1-15
8	Parametre arabirimi için hakları değiştirme.	2,3,5,7,8,11-14,15
9-12	Kullanılmaz.	-
13	Ayrılmış.	-
14	Ayrılmış.	-
15	Kullanılmaz.	15

Açık veri bloğunda yer alan PZD alanı sürücünün izlenmesi ve denetlenmesi için tasarlanmıştır. PZD işleme PKW işleminden daha önceliklidir. MM420 0-4 sözcükten oluşan PZD'lerle çalışabilmesine rağmen normalde 2 sözcük PZD kullanır. P2012 parametresinin değeri değiştirilerek PZD uzunluğu ayarlanabilir.

PZD görev telegramının ilk sözcüğü, sürücü denetim sözcüğüdür (STW) ve sürücünün USS üzerinden denetlenmesini sağlar. Çizelge 4.3'te belirtilen denetim sözcüğü görevleri bitlerin uygun şekilde ayarlanması ile gerçekleştirilebilir.

Çizelge 4.3 Denetim sözcüğü görevleri

Bit0	AÇ/KAPAT1(rampalı)	0-Hayır, 1-Evet
Bit1	KAPAT2(elektriği kes)	0-Evet, 1-Hayır
Bit2	KAPAT3(Hızlı dur)	0-Evet, 1-Hayır
Bit3	Darbe yetkisi	0-Hayır, 1-Evet
Bit4	RFG yetkisi	0-Hayır, 1-Evet
Bit5	RFG başla	0-Hayır, 1-Evet
Bit6	Ayar noktası yetkisi	0-Hayır, 1-Evet
Bit7	Hata onayı	0-Hayır, 1-Evet
Bit8	Sağa dürt(JOG)	0-Hayır, 1-Evet
Bit9	Sola dürt	0-Hayır, 1-Evet
Bit10	PLC'den denetim	0-Hayır, 1-Evet
Bit11	Ayar noktasını ters çevir	0-Hayır, 1-Evet
Bit12	Atanmamış	
Bit13	Motorpoti yükselt	0-Hayır, 1-Evet
Bit14	Motorpoti düşür	0-Hayır, 1-Evet
Bit15	Yerel/uzak	0-P0719 ind 0 1-P0719 ind 1

PZD görev telegramının ikinci sözcüğü ana ayar noktasıdır (HSW). P2009'un değerine bağlı olarak belirlenebilen iki farklı yol vardır. Eğer parametre 0 olarak

ayarlanırsa deęer onaltılık bir sayı olarak gönderilir. Parametre 1 olarak ayarlanırsa deęer mutlak onluk bir sayı olarak gönderilir. Örneęin onluk 4000 (onaltılık 0FA0) deęeri 40.00Hz gibi yorumlanır.

PZD yanıt telegramının ilk sözcüęü sürücü durum sözcüęüdür (ZSW). Durum sözcüęünün bitlerine atanmış geri bildirimler Çizelge 4.4'te görölmektedir.

Çizelge 4.4 Durum sözcüęü geri bildirimleri

Bit0	Sürücü hazır	0-Hayır, 1-Evet
Bit1	Sürücü çalışmaya hazır	0-Hayır, 1-Evet
Bit2	Sürücü çalışıyor	0-Hayır, 1-Evet
Bit3	Sürücü hatası etkin	0-Evet, 1-Hayır
Bit4	KAPAT2 etkin	0-Evet, 1-Hayır
Bit5	KAPAT3 etkin	0-Hayır, 1-Evet
Bit6	Engel etkini anahtarla	0-Hayır, 1-Evet
Bit7	Sürücü uyarı etkin	0-Hayır, 1-Evet
Bit8	Ayar noktası/gerçek deęer sapması	0-Evet, 1-Hayır
Bit9	PZD denetim	0-Hayır, 1-Evet
Bit10	Azami frekansa ulaşıldı	0-Hayır, 1-Evet
Bit11	Uyarı: Motor akım sınırı	0-Evet, 1-Hayır
Bit12	Motor tutma freni etkin	0-Evet, 1-Hayır
Bit13	Motor aşırı yüklemesi	0-Evet, 1-Hayır
Bit14	Motor sağa doğru dönüyor	0-Hayır, 1-Evet
Bit15	Sürücü aşırı yüklenmesi	0-Evet, 1-Hayır

PZD yanıt telegramının ikinci sözcüęü ana gerçek deęer (HIW) dir. Bu, normal olarak sürücü çıkış frekansı olarak tanımlanır. P2009 vasıtasıyla yapılan normalizasyon bu deęere ayrıca uygulanır.

Veri aktarım yordamı:

Başlangıç karakteri STX (onaltılık 02) yalnız başına bir telegramın başlangıcını açıkça tanımlamak için yeterli deęildir. Çünkü onaltılık 02 sayısı ayrıca veri karakterlerinde de oluşabilir. Bu yüzden ana birim tarafından STX'ten önce en az 2 karakter yürütme zamanı kadar başlangıç gecikmesi belirlenir. Örneęin iletişim 9600 baud oranı hızında ise 2,3ms, 19200 baud oranında ise 1,15ms başlangıç gecikmesi olmalıdır. Başlangıç gecikmesi ya da aralığı görev telegramın bir parçasıdır ve uydu düęüm tarafından izlenir. Sadece önceki başlangıç aralığı ile birlikte bir STX geçerli bir telegram başlangıcını tanımlar (Anonim H).

Görev telegramın son karakteri (BCC) ile yanıt telegramın başlangıcı (STX) arasındaki zaman aralığı yanıt gecikme süresi olarak anılır. Yanıt gecikme süresi ana

birim tarafından izlenir. İzin verilebilecek maksimum yanıt gecikme süresi 20ms'dir. Süre aşılsa hata üretilir (Anonim H).

Bir telegram alındığında, doğru telegram başlangıcı ilk olarak tanımlanmalı (başlangıç aralığı+STK) ve ondan sonra uzunluk (LGE) değerlendirmelidir. Eğer uzunluk bilgisi sabit telegram uzunluğunda seçilen değere karşılık gelmiyorsa ya da değişken bir telegram uzunluğu için geçersiz ise telegram reddedilir. Telegram kabulü sırasında ve öncesinde zaman izlenmek zorundadır (Anonim H).

Blok denetim karakteri (BCC) kabul sırasında üretilir ve tüm telegramdan okunduktan sonra alınan BCC ile karşılaştırılır. Bunlar uyuşmuyor ise telegram değerlendirilmez.

Bir karakter çerçeve hatası yada bir eşlik hatası alınan karakterlerin hiçbirinde meydana gelmez ise alınan telegramın düğüm numarası (ADR) değerlendirilebilir.

Adres baytı (ADR) düğüm numarası (uydu için) ile ya da beklenen uydu düğüm numarası (ana için) ile uyuşmaz ise telegram reddedilir.

USS için parametre ayarları:

MM420 USS iletişim için kullanabileceği iki seri arabirime sahiptir: RS232 ve RS485. RS232 seçimli bir modül kullanılarak gerçekleştirilir. RS485 arabirimi P+ ve N- hatları için sırayla 14 ve 15 nu.lu terminaleri kullanır. USS parametreleri genellikle iki indis içerir. İndeks 0 RS485, indeks 1 RS232 içindir.

USS vasıtasıyla iletişim kurmak için hangi arabirimin kullanılacağına karar verilmesi gerekir.

- **P0003=2** (2. seviye parametrelere erişmek için gerekli)
- **P2010=USS baud oranı.** Ana birim tarafından kullanılacak Baud oranı ile aynı olmalıdır. Desteklenen maksimum Baud oranı 57600 Baud'tur.
- **P2011=USS düğüm adresi.** Sürücü için eşsiz bir uydu adresidir.

Bu parametreler ayarlanır ayarlanmaz iletişim kurulabilir. Ana birim parametre okuyup yazabilir (PKW alanı) ve ayrıca sürücünün durumunu ve gerçek frekansını (PZD alanı) izleyebilir.

- **P0700=4 yada 5.** Bu, sürücünün USS vasıtasıyla denetlenmesine izin verir. Normal RUN ve OFF1 komutları sırayla onaltılık 047F ve 047E'dir.

- **P1000=4 yada 5.** Bu, ana ayar noktasının USS yoluyla gönderilmesine izin verir. Varsayılan olarak bu, P2000 kullanılarak normalize edilir. Önceki Micromaster sürücüleri ile geriye uyumluluk olması için P2009 kullanılarak normalizasyon yapılması mümkündür.

P0700 ve P1000 birbirinden bağımsızdır ve gerektiğinde ayrı olarak ayarlanmalıdır.

USS telegramı özel uygulamalar için aşağıdaki parametreler (sadece 3. seviye) ile isteğe göre uyarlanabilir.

- **P0003=3**, uzman parametrelerinin erişimine izin verir.
- **P2012=USS PZD uzunluğu.** Normal PZD uzunluğu önceden açıklandığı gibi 2 sözcüktür. Bu parametre kullanıcının denetim ve izleme amaçları için farklı bir uzunlukta PZD seçebilmesine izin verir. Örneğin 3 sözcüklük bir PZD ikinci bir ayar noktası ve gerçek değer kullanımına izin verir. İkinci gerçek değer, örneğin sürücü çıkış akımı olarak ayarlanabilir (P2016 yada P2019 indeks 3=27).
- **P2013=USS PKW uzunluğu.** Varsayılan olarak bu parametre 127'ye (değişken) ayarlıdır. Bunun anlamı PKW uzunluğunun değiştirilebileceği ve ayrıca yanıt telegram uzunluğunun değişken olabileceğidir. Bu toplam USS telegram uzunluğunu etkiler. Bir denetim programı yazılıyorsa sabit telegram uzunluğu faydalı olabilir. Örneğin yanıt durum sözcüğü daima aynı konumda oluşur. MM420 için kullanışlı sabit PKW uzunluğu 4 sözcüktür. Bu tüm parametrelerin yazılıp okunmasına izin verir.
- **P2016 ve P2019:** Bunlar kullanıcının yanıt PZD telegramlarında dönecek durum sözcükleri ve gerçek değerlerin tanımlayabilmesine izin verir (sırayla RS232 ve RS485 için). Bu parametreler aşağıdaki gibi ayarlanabilen indekslere sahiptir:

İndeks0=durum sözcüğü 1 (ZSW) (varsayılan 52=sürücü durum sözcüğü)

İndeks 1=gerçek değer 1 (HIW) (varsayılan 21=çıkış frekansı)

İndeks 3=gerçek değer 2 (HIW2) (varsayılan 0)

İndeks 4=durum sözcüğü 2 (ZSW2) (varsayılan 0)

4.2.2 Kullanıcı arayüz yazılımı

Kullanıcı arayüz yazılımı, kullanıcının uzaktan denetim arayüz devresi ile bağlantısını kurmasını, komut göndermesini ve veri alabilmesini sağlamaktadır. Arayüz yazılımı için Java Applet teknolojisi kullanılmıştır. Güvenliği sağlamak amacıyla tanımlı kullanıcı ve şifre tanımlanmıştır. Geçerli kullanıcı adı ve şifresi girilmediği takdirde arayüz devresi ile bağlantı kurulamamaktadır. Arayüz yazılımının kontrol mantığı için Micromaster 420 sürücüsünün BOP paneli kullanımı model alınmıştır. Şekil 4.18’de kullanıcı arayüz appleti görüntüsü verilmiştir.

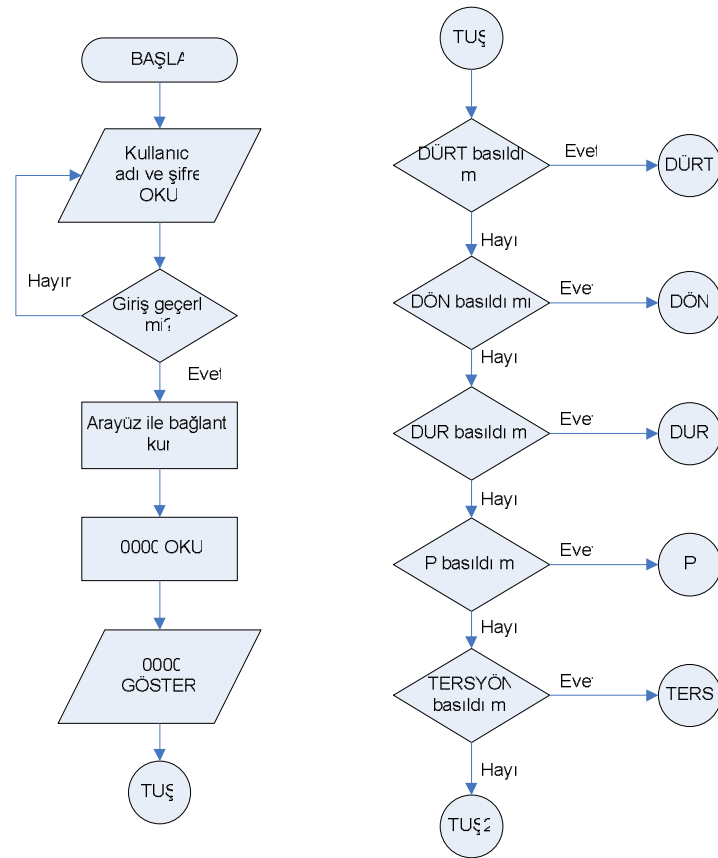


Şekil 4.18 Kullanıcı arayüz appleti

Kullanıcı adı ve şifresi girilip Bağlan tuşu tıklandığında geçerli bir giriş ise arayüz devresi ile bağlantı kurulur. Bağlantı kurulunca gösterge r0000 parametresinin değerini gösterecektir. P tuşu, parametre değerlerini görmek ve değişikliği kaydetmek için kullanılır. Artır ve Azalt tuşları istenilen parametre numarasına ulaşılmasını ve parametre değerinin ayarlanmasını sağlar. Parametreler arasında gezilirken r0000 parametresine dönmek için Fn tuşu kullanılabilir. Dürt (Jog) tuşu önceden belirlenmiş dürt frekansında motoru döndürür. Bu tuş basılı kaldığı sürece motor döner, bırakıldığında ise durur. Eğer motor duruyorsa bu tuş işe yaramaz. TersYön tuşu motorun dönüş yönünü değiştirmek için kullanılabilir. Dön tuşu, belirlenen frekansta motoru döndürür. Dur tuşuna bir kez basılarak Dur1, iki kez yada uzun basınca Dur2 moduna geçilir. Dur1 modu belirlenen oranda yavaşça motoru durdururken, Dur2 ise motorun gerilimini keserek durdurur.

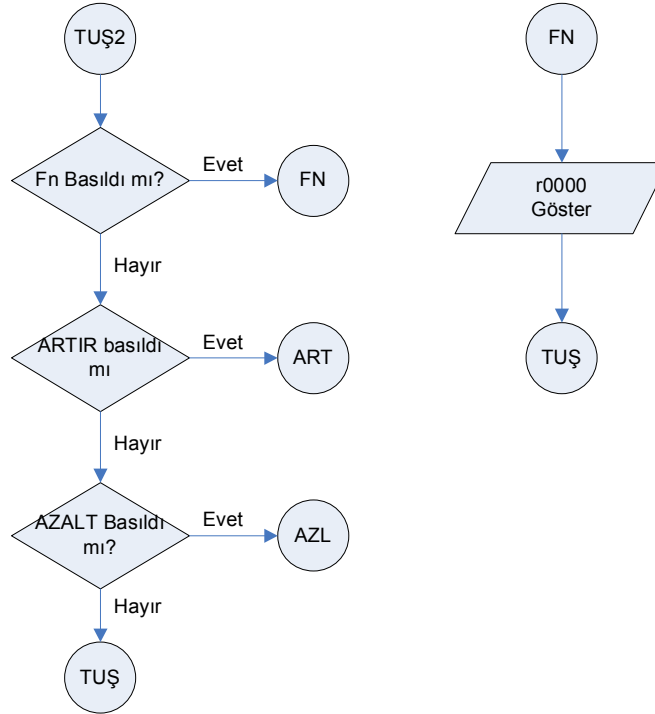
Şekil 4.19’da akış şeması ana bloğu görülmektedir. Akış şeması ana bloğunda öncelikle kullanıcı adı ve şifre okunmaktadır. Geçerli kullanıcı adı ve şifre girilmiş

ise arayüz donanımı ile bağlantı kurulmaktadır. Arayüz donanımı ile bağlantı kurulmasının ardından motor sürücüsünden r0000 parametresinin değeri okunmakta ve değer ekranda gösterilmektedir. Ardından sırayla Dürt, Dön, Dur, P, TersYön, Fn, Artır ve Azalt tuşlarının basılı olup olmadığını denetlenmektedir. İlgili tuş basılı ise ilgili alt akış şemaya dallanılmakta, basılı değil ise bir sonraki tuşun denetimine geçilmektedir. Tüm tuşlar denetlendikten sonra tuş denetiminin başladığı bölüme dallanılmaktadır.



Şekil 4.19 Akış şeması ana bloğu

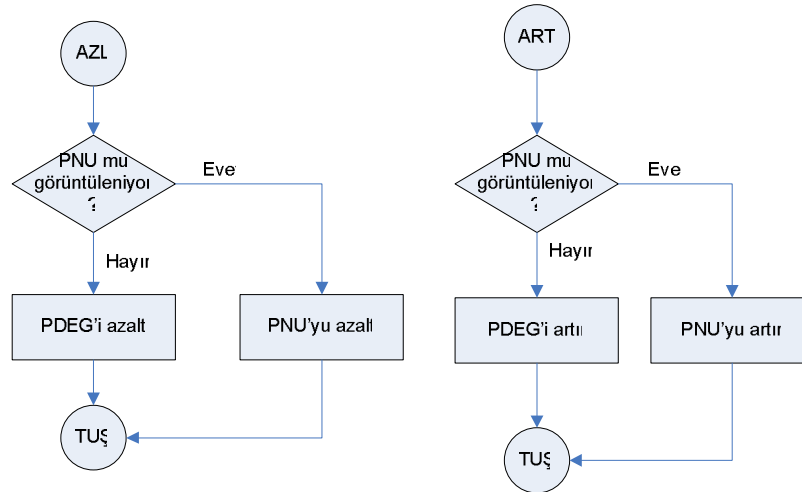
Şekil 4.20'de ana blok akış şemasının devamı ve Fn tuşu akış şeması görülmektedir. Fn tuşuna basıldığında ilgili alt yordama dallanılmakta ve ekranda r0000 parametresinin görüntülenmesini sağlamaktadır. Fn tuşu, çok büyük numaraya sahip bir parametre üzerinde işlem yapıldıktan sonra hızlı bir şekilde ilk parametreye dönülmesi amacıyla kullanılmaktadır.



Şekil 4.20 Ana blok akış şemasının devamı ve Fn tuşu akış şeması

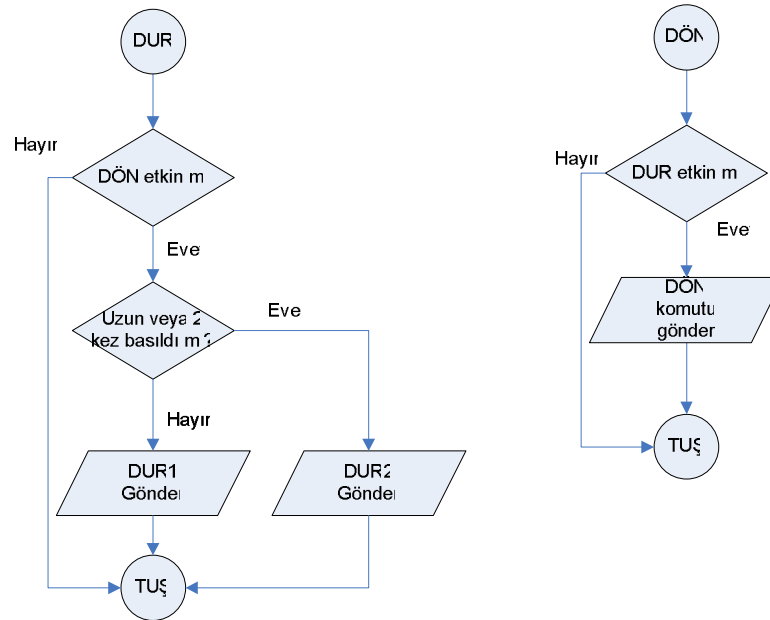
Şekil 4.21’de Azalt ve Artır tuşları için akış şemaları görülmektedir. Artır ve Azalt tuşlarına basıldığında öncelikle ekranda parametre numarasının görüntülenip görüntülenmediği sorgulanmaktadır. Eğer parametre numarası görüntüleniyorsa görüntülenen parametre numarası basılan tuşa bağlı olarak bir artırılmakta ya da azaltılmaktadır. Eğer parametre numarası görüntülenmiyor ise parametre değeri görüntülenmektedir. Bu nedenle basılan tuşa bağlı olarak parametre değeri bir artırılmakta ya da azaltılmaktadır. Ardından ana blok akış şemasında tuş denetimlerinin gerçekleştirildiği yere dallanılmaktadır.

Şekil 4.22’de Dur ve Dön tuşları için akış şemaları görülmektedir. Dur tuşuna basıldığında öncelikle motorun dönüp dönmediği sorgulanmaktadır. Motor dönmekte ise Dur tuşuna uzun basılıp basılmadığı veya 2 kez basılıp basılmadığı sorgulanmaktadır. Dur tuşu eğer bir kez kısa bir süre basılmış ise motor sürücüsüne Dur1 komutu gönderilmektedir. Dur1 komutu motorun ilgili parametrelerde belirtilmiş bir süre içinde yavaşça durmasını sağlamaktadır. Eğer Dur tuşuna 2 kez veya bir kez uzun bir süre basılmış ise motor sürücüsüne Dur2 komutu gönderilmektedir. Dur2 komutu, sürücünün motor gerilimini keserek motoru durdurmasını sağlamaktadır.



Şekil 4.20 Azalt ve Artır tuşları için akış şeması

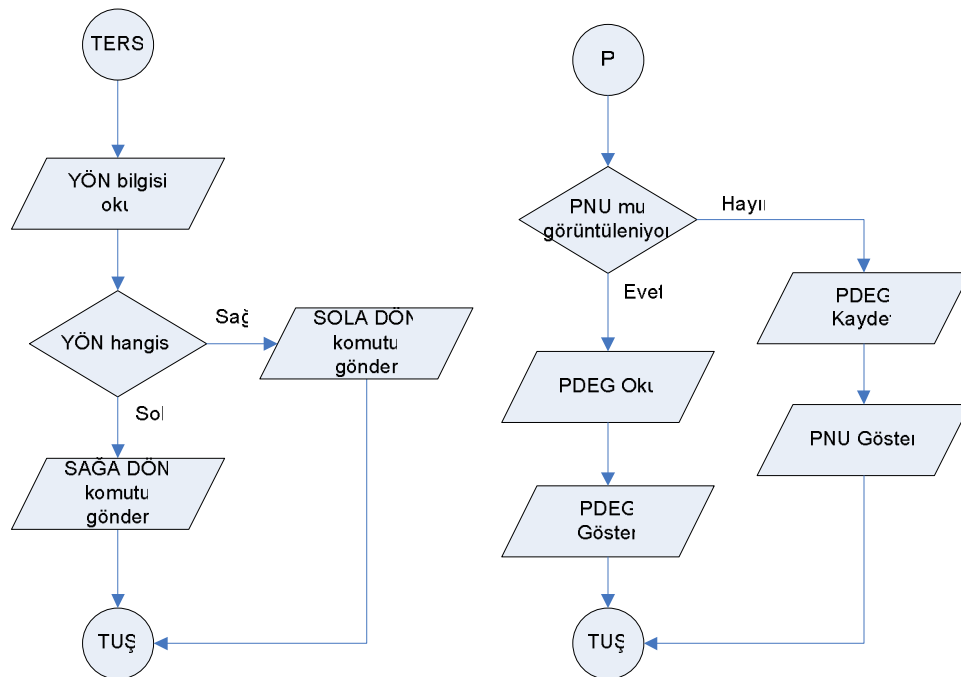
Dön tuşu basıldığında öncelikle motorun durur durmadığı sorgulanmaktadır. Eğer motor duruyor ise motor sürücüsüne motoru döndürme komutu gönderilerek motorun dönmesi sağlanmaktadır. Ardından ana blok akış şemasında tuş denetimlerinin yapıldığı bölüme dallanılmaktadır.



Şekil 4.21 Dur ve Dön tuşları akış şeması

Şekil 4.23'te TersYön ve P tuşlarının akış şeması görülmektedir. TersYön tuşu basıldığında öncelikle yön bilgisi okunmakta ardından motorun dönüş yönü sorgulanmaktadır. Motor sola dönüyor ise sağa dön, sağa dönüyor ise sola dön

komutu gönderilmektedir. P tuşuna basıldığında öncelikle ekranda parametre numarasının görüntülenip görüntülenmediği sorgulanmaktadır. Eğer ekranda parametre numarası görüntüleniyor ise ilgili parametrenin değeri okunmakta ardından okunan değer ekranda görüntülenmektedir. Eğer parametre numarası görüntülenmiyor ise görüntüleneni parametre değeri kaydedilmekte ve değeri görüntüleneni parametrenin numarası görüntülenmektedir. Ardından ana blok akış şemasında tuş denetimlerinin yapıldığı bölüme dallanılmaktadır.



Şekil 4.22 TersYön ve P tuşları akış şeması

5. BULGULAR

Yapılan kaynak araştırması sonucunda İnternet tabanlı uzaktan denetim sistemlerinin istemci ve sunucu bilgisayarlara yüklenen yazılımlar vasıtasıyla gerçekleştirildiği gözlenmiştir. Bazı sistemler bilgisayar yerine mobil iletişim cihazı kullansa dahi sistem yapısı aynı kalmakta ve özel bir istemci ile denetim gerçekleştirilmektedir. Çalışmada istemci taraf için tasarlanan uzaktan denetim arayüz devresi ile insan-makine arayüz yazılımının İnternet'e bağlı herhangi bir bilgisayar vasıtasıyla erişilebilmesi mümkün hale gelmiştir.

Diğer Ethernet denetçilerine nazaran maliyeti bu günkü koşullarda 5-6 yıl daha yüksek olmasına rağmen ENC28J60 mikrodenetleyiciler ile birlikte arayüz tasarımında süre ve iş yükü bakımından üstünlük sağlamaktadır. Ancak, mikrodenetleyici belirleme aşamasında Ethernet denetçisi ile haberleşebilmesi için SPI arayüzüne sahip, seri iletişim yöntemi olarak kullanılacak RS485'i destekleyen ve ağ programcılığı için yeterli program belleğine sahip bir mikrodenetleyici kullanılmalıdır. Ethernet yongasının IP ve MAC adresi program başlatıldığında mikrodenetleyici tarafından atanır. Yerel ağ üzerindeki diğer makinelerle IP çakışması olmayacak şekilde IP atanması kullanıcının sorumluluğundadır.

Uzaktan denetim uygulamasında mikrodenetleyici için sabit yazılımının hazırlanacağı derleyicinin belirlenmesi aşamasında aynı zamanda ENC28J60 için ve USART modülü için hazırlanmış komut kütüphanesine sahip olması tasarımcılara kolaylık sağlayacaktır. Derleyici kullanımında mikroC derleyicisi komutların açık kaynak kodlarını vermemektedir. Bu durum ENC28J60 kullanımında ve RS485 kullanımında tasarımcının iş yükünü artırabilir. Ancak mikroC derleyicisi oldukça gelişmiş komut kütüphanesine sahiptir. Bu durum kullanımda diğer derleyicilere üstünlük sağlayabilir.

Farklı makineler farklı fiziksel arayüzler ve protokoller kullanabilmektedir. Arayüz devresinde iletişimden ve denetimden sorumlu, hızlı ve yaygın kullanılan bir mikrodenetleyici olan PIC18F452, geniş ve yeniden programlanabilir program hafızası ile programın güncellenebilmesini veya başka makinelere uyarlanabilmesi üstünlüklerini sağlamaktadır.

Uzaktan denetim arayüz sisteminin kullanımına örnek olması açısından bir asenkron motor sürücüsü olan Micromaster 420'nin uzaktan denetimi gerçekleştirilmiştir. Sistem kolayca USS protokolüne sahip diğer Micromaster sürücüleri için uyarlanabilir. Ayrıca iletişim protokolü USS'ye çok yakın olan PROFIBUS ağı mikrodenetleyici programı uyarlanarak kullanılabilir.

Mikrodenetleyici ile Mikromaster 420 sürücüsü arasında iletişim, gürültü bağıklığı, çok düğümlü bağlantı, yüksek veri aktarım oranları ve uzun mesafe iletişim gibi üstünleri nedeniyle RS485 fiziksel arayüzü yerine RS232 fiziksel arayüzü kullanımı pek önerilmemektedir. Ancak, RS232 ile uçtan uca iletişim için RS485 alıcı-vericisine paralel bir RS232 alıcı-vericisi kullanılmalıdır.

Tasarlanan arayüz devresi mikrodenetleyici sabit yazılımı içine HTML ve Javascript kodu gömülerek mini sunucu olarak denenmiş ve başarılı sonuç alınmıştır. Kullanıcı arayüz yazılımı için Applet teknolojisi kullanımı Javascript kullanımına göre daha iyi sonuçlar vermektedir. Applet, mikrodenetleyici ile okunabilecek ilave bir belleğe yüklenerek arayüz devresi gelişmiş bir sunucu gibi kullanılabilir. Diğer yandan applet'ler yapısı gereği güvenlik nedeniyle çok sayıda kısıtlamaya sahiptir. Kısıtlamaları kaldırmak için yaygın olarak applet dosyası sayısal imza iliştilmiş bir JAR dosyası olarak kullanılmaktadır. Ancak appletin arayüze gömülmesi durumunda appletin güvenlik kısıtlamaları ortadan kalkmakta, dolayısıyla sayısal imzalama işlemine gerek kalmamaktadır.

6. SONUÇ VE DEĞERLENDİRME

Bu tez çalışmasında, uzaktan denetim ve izleme sistemlerine bir alternatif tasarlanmış ve gerçekleştirilmiştir. Uzaktan denetim ve izleme ortamı olarak İnternet, erişim yöntemi olarak İnternet'e bağlanabilen herhangi bir bilgisayar ile açılabilen web sayfasına gömülü bir arayüz appleti tercih edilmiştir. İnternete bağlı ve JVM yüklü herhangi bir bilgisayar ile uzaktan denetim ve izleme yapılabilmektedir. Denetlenecek makine tasarlanan arayüz devresine bağlanmaktadır. Arayüz devresi, eğer mikrodenetleyici kodunda arayüzün bağlanacağı yerel ağa uygun bir IP tanımlanmış ise çalışır çalışmaz belirtilen ağla doğrudan bağlantı kuracaktır. Dolayısıyla Ethernet ağı üzerinden TCP veya UDP paketlerini almak, göndermek için bir bilgisayar gerekmemektedir. Bilgisayar yerine tasarlanan arayüz devresinin kullanılmasıyla hem kurulum maliyeti düşürülmüş hem de virüs gibi çeşitli sorunlardan kaynaklanabilecek zaman ve bakım masrafı kayıpları önlenmiştir.

Önerilen bu sistemde kullanıcı arayüz appleti herhangi bir Web sunucusunda saklanmaktadır. Appletler indirildikleri sunucu dışında başka bir sunucu ile socket bağlantısı kuramadıklarından sayısal olarak imzalanarak kullanılmıştır.

Uzaktan denetlenecek ve izlenecek makineler iletişim protokolü olarak USS yerine Modbus, Profibus DP, FMS, PA, CanOPEN, vb. protokollerden biri ya da birkaçını destekliyor olabilir. Aynı zamanda RS485 yerine RS232, USB ya da optik fiziksel arayüzlerinden birini destekliyor olabilir. Önerilen bu sistem küçük donanımsal eklemeler ile farklı fiziksel arayüzlerine, yazılımsal değişiklikler veya eklemeler ile farklı iletişim protokollerine kolayca uyarlanabilir. Örneğin mikrodenetleyicinin USART modülüne bağlanacak FT232 yongası ile USB destekli makineler, benzer şekilde MAX232 alıcı-vericisi bağlanarak RS232 arayüzünü kullanan makineler denetlenebilir. USB veya RS232 fiziksel arayüzlerine uyarlama işi RS485-RS232 dönüştürücü kablo veya RS485-USB dönüştürücü kablo yardımı ile de gerçekleştirilebilir.

Son zamanlarda mikrodenetleyici programlarının, mikrodenetleyici devre üzerinde iken güncellenebilmesini sağlayan bootloader yazılımları geliştirilmektedir. Ethernet için uyarlanmış bir bootloader yazılımı kullanılarak arayüz sistemi uzaktan başka makinelere uygun hale getirilebilir. Uzaktan sabit yazılım güncelleyebilmek için arayüz sistemi biri Ethernet üzerinden veri-alışverişinden diğeri makine iletişim

protokolü ve fiziksel arayüzünden sorumlu iki mikrodenetleyicili olacak şekilde yeniden düzenlenebilir.

Tasarlanan ve gerçekleştirilen uzaktan denetim arayüz sistemi, kurulum maliyeti ve bakım maliyeti açısından mevcut arayüz sistemlerine göre daha uygundur. İnternet'e bağlanabilen herhangi bir bilgisayar yardımıyla denetim ve izleme yapılabilmesinden dolayı erişimi son derece kolaydır. Farklı iletişim iletişim kurallarına ve fiziksel arayüzlere sahip makinelere uyarlanabilir olmasından dolayı diğer uzaktan denetim ve izleme sistemlerinden ayrılmaktadır.

KAYNAKLAR

- Alp, U., Ayaz, H., Karadeniz, M., Dikici, Ç., Işıl Bozma, H. ,*İnternet Üzerinden Uzaktan Robot Erişimi*, Proc. of SIU 2003.
- Atabaş, İ., Arslan, M., Uzun, İ., *Isıtma Sistemlerinin Otomasyonu ve İnternet Üzerinden Kontrolü*, Akademik Bilişim Kongresi, 2007.
- Baykal, N., 2005. *Bilgisayar Ağları ikinci baskı*, SAS Bilişim Yayınları, Ankara, 542p.
- Çolak, İ., Irmak, E., Sefa, İ., Demirbaş, Ş., Bayındır, R. *Temel Elektrik Devrelerinin Analizi için İnternet Tabanlı Bir Eğitim Aracı*, 6th International Educational Technology Conference, 2006, Eastern Mediterranean University, pp. 452-461, April, Gazimağusa, Kuzey Kıbrıs Türk Cumhuriyeti.
- Çölkesen, R., Örencik, B., 2003. *Bilgisayar Haberleşmesi ve Ağ Teknolojileri*, Papatya Yayıncılık Eğitim Bilgisayar Sis. San. ve Tic. A.Ş., İstanbul, 447p.
- Eşme, E., *Uzaktan Kontrol Edilebilen Bir Kalp Cihazı Tasarımı*, Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 2006.
- Flanagan, D., 1997. *Java in a Nutshell a Desktop Quick Reference Second Edition*, O'REILLY Associates Inc., Sebastopol, 609p.
- Gümüşkaya, H., Boyacı, Ö., 2003. *Herkes için Java Ağ Programcılığı*, Alfa Basım Yayım Dağıtım Ltd. Şti., İstanbul, 720p.
- Javorski, J., 1997. *Java 1.1 Developer's Guide Second Edition*, Sams.net Publishing, Indianapolis, 1100p.
- Kaplan, Y., 2000. *Veri Haberleşmesi Temelleri*, Papatya Yayıncılık Eğitim Bilgisayar Sis. San. Ve Tic. A.Ş., İstanbul, 352p.
- Karaçor, M., *Cep Telefonu Tabanlı Scada Otomasyon Sisteminin Geliştirilmesi*, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, 2004.
- Korkmaz, F., *Elektrikli Cihazlar ve Güvenlik Sistemlerinin Cep Telefonuyla Uzaktan Denetlenmesi*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2004.
- Schildt, H., 2003. *Herkes için Java 2*, Alfa Basım Yayım Dağıtım Ltd. Şti., İstanbul, 992p.
- Tunalı, A., *Telefon Hatlarını Kullanarak PC ile Cihazların Uzaktan Kontrolü*, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya, 2004.
- Ünlü, B., *İnternet Üzerinden Mobil Bir Robotun Kontrolü*, Lisans Tezi, Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi, İstanbul, 2007.
- Yen-Shin, L., Jennshing, W., Zhong-Qing, L., Min-Hao, W., Su-Chen, T., 2002. *Internet-Based Monitoring and Control of Fuzzy-Controlled Inverter System*, IEEE

Anonim

- a. EIA-485, Wikipedia the free encyclopedia, <http://en.wikipedia.org/wiki/RS-485>, 2008
- b. ENC28J60 Stand-Alone Ethernet Controller with SPI Interface, Microchip, http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en533418
- c. Java Tutorials, Sun Microsystems, <http://java.sun.com/docs/books/tutorial/index.html>, 2007
- d. Java Network Programming FAQ, David Reilly, http://www.davidreilly.com/java/java_network_programming/, 2000
- e. Micromaster 420 0,12 kW - 11 kW, Siemens Industry Automation and Drive Technologies Service&Support, <http://support.automation.siemens.com>
- f. PIC18F452 datasheet, Microchip Technology, <http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>
- g. RS485 serial information, Lammert Bies, <http://www.lammertbies.nl/comm/info/RS-485.html>, 2008
- h. Universal Serial Interface Protocol USS Protocol, Siemens Industry Automation and Drive Technologies Service&Support, <http://support.automation.siemens.com>
- i. [Using USS protocol with MM420.pdf](#), Siemens Industry Automation and Drive Technologies Service&Support, <http://support.automation.siemens.com>

ÖZGEÇMİŞ

Muammer Kadir KARAGÖZ 1981 yılında Muğla'da doğdu. İlk ve orta öğretimi Muğla'da bulunan devlet okullarında tamamladı. 1998 yılında Muğla Endüstri Meslek Lisesinden okul ikincisi olarak mezun oldu. Aynı yıl Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Elektronik Öğretmenliği programında yüksek öğrenime başladı ve 2002 yılında mezun oldu. 2003 yılında Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim dalı tezsiz yüksek lisans programına başladı. 2006 yılında bu programı bırakarak Muğla Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim dalında tezli yüksek lisans programına başladı.

2003-2005 yılları arasında Kars GAMP Endüstri Meslek Lisesinde, 2005-2006 yılları arasında Ankara Dikmen Nevzat Ayaz Anadolu Meslek Lisesinde Elektronik Öğretmenliği yaptı. 2006 yılında MEGEP kapsamında Biyomedikal Cihaz Teknolojileri alanı modül inceleme çalıştaylarına katıldı ve iki modül yazımı gerçekleştirdi. İstanbul Hadımköy İbrahim Özyaydın Çok Programlı Lisesinde 2006 yılında başladığı Elektronik öğretmenliği görevini halen sürdürmektedir.