

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WEB TABANLI BİR SCADA SİSTEMİNİN GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

ÇAĞRI KARAKAŞ

071131101

Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi

Programı: Bilgisayar Sistemleri

Danışman: Doç. Dr. Ömür AYDOĞMUŞ

ELAZIĞ - 2016

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WEB TABANLI BİR SCADA SİSTEMİNİN GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ
ÇAĞRI KARAKAŞ

Tezin Enstitüye Verildiği Tarih : 15 Haziran 2016

Tezin Savunulduğu Tarih : 13 Temmuz 2016

Tez Danışmanı : Doç. Dr. Ömür AYDOĞMUŞ (F.Ü.)

Diğer Jüri Üyeleri : Doç. Dr. Mehmet GEDİKPINAR (F.Ü.)

Yrd. Doç. Dr. Ömer ÇELİK (T.Ü.)

ELAZIĞ - 2016

ÖNSÖZ

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Sayın Doç. Dr. Ömür AYDOĞMUŞ' a teşekkürlerimi sunarım.

Çağrı KARAKAŞ

ELAZIĞ-2016



İÇİNDEKİLER

Sayfa No

ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLOLAR LİSTESİ	XI
SEMBOLLER LİSTESİ	XII
KISALTMALAR LİSTESİ	XIII
1.GİRİŞ.....	1
1.1. Tezin Amacı	1
1.2. Tezin Organizasyonu.....	4
2. ENDÜSTRİYEL OTOMASYON.....	5
2.1. Endüstriyel Veri İletişimi	6
2.1.1. OSI Modeli	7
2.1.2. RS-232 Arayüz Standardı.....	8
2.1.3. Fiber optik	8
2.1.4. Modbus	10
2.1.5. Data Highway Plus /DH485	13
2.1.6. HART	14
2.1.7. AS-i.....	15
2.1.8. DeviceNet.....	16
2.1.9. Profibus.....	17
2.1.10. Foundation Fieldbus	18
2.1.11. Endüstriyel Ethernet	19
2.1.12. TCP/IP	21
2.1.13. Profinet	22
2.2. Otomasyon Sistemlerinin Bileşenleri	23
2.2.1. Programlanabilir Mantıksal Denetleyici (PLC).....	23
2.2.2. HMI	25
2.2.3. SCADA.....	25
2.2.3.1. Giriş ve SCADA’ nın Tarihçesi.....	25
2.3. ARM Tabanlı İşlemciler.....	34
2.3.1. ARM Mimarisi	34
2.3.2. ARM İşlemcili Geliştirme Kartları.....	36
2.3.3. Gömülü Sistemler.....	41
3. WEB TEKNOLOJİLERİ	43
3.1. HTML.....	43
3.2. JavaScript	45
3.3. FLASH ve Actionscript.....	50
4. RASPBERRY PI.....	53
4.1. Donanım	54
4.2. Yazılım	58
4.3. I2C ve SPI Haberleşme.....	59
4.3.1. I2C	61

4.3.2. SPI	65
5. İŞLETİM SİSTEMİ VE PROGRAMLAMA DİLİ.....	67
5.1 Linux.....	67
5.2. Python Programlama Dili	68
5.3. Flask	72
6. WEB TABANLI SCADA SİSTEMİNİN GELİŞTİRİLMESİ	76
6.1. Sistemin Tanımlanması	76
6.2. Sistemin Elektronik Alt Yapısı.....	76
6.3. Yazılım Altyapısı.....	79
6.4. Web Arayüzü Tanıtımı	79
7. SONUÇLAR.....	87
KAYNAKLAR.....	89
ÖZGEÇMİŞ	92



ÖZET

Her geçen gün endüstriye olan ihtiyaç hızlı bir şekilde artmaktadır. Buna bağlı olarak kaliteyi arttırmak ve maliyetleri azaltmak için genellikle insan müdahalesinin olmadığı veya daha az olduğu otomatik olarak çalışabilen sistemlerin kullanılması gerekmektedir. Bundan dolayı, endüstriyel uygulamaların neredeyse tamamında otomasyon sistemleri kullanılmaktadır. Otomasyon sistemleri Programlanabilir Mantıksal Denetleyici (PLC), Merkezi Denetim ve veri toplama (SCADA) ve İnsan-Makine Arayüzü (HMI) gibi yüksek maliyetli cihazların birlikte kullanılmasından oluşmaktadır. Özellikle SCADA veya HMI uygulamaları sistem maliyetini büyük ölçüde arttırmaktadır. Ayrıca, PC tabanlı otomasyon sistemlerinde kullanılan işletim sistemi ve SCADA Runtime lisansları da ek bir maliyet oluşturmaktadır. Bilgisayar temelli SCADA sistemlerinde çalışma koşullarına bağlı olarak bilgisayardan dolayı bazı problemler oluşabilir.

Son zamanlarda, bazı firmalar tarafından piyasaya sürülen ARM (Advanced Risc Machine) tabanlı Linux işletim sistemi çalıştırabilen kartlar sayesinde çok düşük maliyetli çözümler elde etmek mümkün olmuştur. Bu cihazlar standart bilgisayarlar ile karşılaştırıldıklarında oldukça düşük boyut ve fiyata sahiptirler. Ayrıca, ARM-tabanlı cihazlar açık kaynak donanım/yazılım gibi avantajlara da sahiptirler. Fiyatları neredeyse bir PC'nin yüzde beşinden daha düşüktür. Bu ürünler, yüksek performans gerektirmeyen birçok uygulamada güvenilir bir şekilde kullanılabilirler.

Bu tezde, küçük ölçekli endüstriyel uygulamalarda kullanılabilecek ARM tabanlı bir otomasyon sistemi geliştirilmiştir. Böylece, tek cihaz ile SCADA, PC ve PLC cihazları olmaksızın kontrol sağlanabilmektedir. Ayrıca WEB-tabanlı olması sayesinde başka cihazlar ile erişim sağlayabilme özelliğine sahip düşük maliyetli bir otomasyon sistemi elde edilmiştir.

Anahtar Kelimeler: Endüstriyel otomasyon, ARM-işlemci, SCADA, HMI, PLC

SUMMARY

THE DEVELOPMENT OF A WEB-BASED SCADA SYSTEM

The need for the industry is increasing rapidly with each passing day. Consequently, it is typically requiring the use of automated systems that are operable without human intervention to improve the quality and reduce costs. Therefore, automation systems are used in almost all industrial applications. Automation systems consist of combined use of high-cost equipment such as Programmable Logic Controller (PLC), Supervisory Control and Data Acquisition (SCADA) and Human–Machine Interface (HMI). In particular, SCADA and HMI applications increase the system cost. Furthermore, the operating system and SCADA Runtime licenses are an additional cost. Some problems may occur depending on operating conditions because of the computer in the PC-based SCADA systems.

Recently, some solutions have been achieved with very low cost thanks to ARM (Advanced Risc Machine) based cards operated with Linux operating system which are released by some companies. These devices have quite small size and low price when compared to standard computers. In addition, the ARM-based devices have some advantages such as open-source hardware/software. Their prices are almost five percent lower than a PC. These products can be used reliable in many applications not requiring high performance. In this thesis, ARM-based automation system has been developed that can be used in small-scale industrial applications. Thus, a single device can provide the control of the system without SCADA, PC and PLC devices. In addition, a WEB-based automation system is obtained with low cost and providing access to other devices thanks to WEB-based.

Keywords: Industrial automation, ARM-Processor, SCADA, HMI, PLC.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1. Türkiye ve gelişmiş ülkelerdeki otomasyon pazarının dağılımı	2
Şekil 1.2. Zorlu endüstriyel koşullar altında çalışan bilgisayarlı sistemler	2
Şekil 2.1. Otomasyon sistemlerinin bileşenleri	6
Şekil 2.2. Open Systems Interconnection (OSI) modeli.....	7
Şekil 2.3. Tipik bir seri veri iletişim bağlantısı	8
Şekil 2.4. Işın bir optik fiber boyunca hareket ediyor.	9
Şekil 2.5. Profibus ve endüstriyel ethernet için fiber optik kablo	10
Şekil 2.6. MODBUS haberleşmesine ait bir uygulama	12
Şekil 2.7. Modbus – Ethernet dönüştürücü	13
Şekil 2.8. Allen Bradley firmasına ait Data Highway haberleşmesi	14
Şekil 2.9. HART sinyalizasyon sisteminin frekans dağılımı.....	14
Şekil 2.10. Siemens firmasına ait bir As-i bağlantı şeması	16
Şekil 2.11. DeviceNet haberleşme ağı örneği	17
Şekil 2.12. Profibus haberleşme örneği	18
Şekil 2.13. Foundation Fieldbus haberleşme örneği.....	19
Şekil 2.14. Endüstriyel ethernet ağı ile ilgili bir örnek.....	20
Şekil 2.15. TCP-IP protokolü ile OSI protokolünün katmanları	22
Şekil 2.16. Profinet ile ilgili bir örnek	23
Şekil 2.17. Çeşitli PLC Görüntüleri.....	24
Şekil 2.18. PLC’ nin genel yapısı	25
Şekil 2.19. Bilgisayarın PLC veya DCS’ ye bağlantısı	27
Şekil 2.20. Modem kullanılarak bilgisayardan RTU’ ya bağlantı	29
Şekil 2.21. Bir SCADA sisteminin bileşenleri	29
Şekil 2.22. APC Rock genel görünüm.....	37
Şekil 2.23. BeagleBone Black genel görünüm	37
Şekil 2.24. Cubieboard2 genel görünüm	38
Şekil 2.25. ODROID-XU genel görünüm	39
Şekil 2.26. Wandboard genel görünüm	39
Şekil 2.27. Pandaboard genel görünüm	40
Şekil 2.28. Bir gömülü sistemin yapısı.....	42
Şekil 3.1. JavaScript altyapısı.....	48
Şekil 3.2. REST düzeni	50
Şekil 3.3. Flash çalışma alanı bölümleri.....	51
Şekil 4.1. Raspberry Pi üst görünüm	54
Şekil 4.2. RPI bileşenlerinin yerlerini belirten bir diyagram.....	55
Şekil 4.3. Raspberry Pi B Rev2 blok şeması	55
Şekil 4.4. Raspberry Pi’ nin pinlerini gösteren üstten görünümü.....	56
Şekil 4.5. GPIO, güç ve topraklama pinleri.....	56
Şekil 4.6. Basit bir elektrik devresi	57
Şekil 4.7. Raspberry Pi’ nin pin yapısı	58
Şekil 4.8. Raspbian ekran görüntüsü	59
Şekil 4.9. Seri port ile iki cihazın iletişimi	62
Şekil 4.10. SPI haberleşme protokolü master - slave bağlantısı.....	63
Şekil 4.11. Tipik bir I2C bağlantısı	64
Şekil 4.12. İki iletim hattındaki pull-up dirençleri	64

Şekil 4.13. I2C sinyalleri	65
Şekil 4.14. Tipik bir SPI bağlantısı	65
Şekil 4.15. SPI veri döngüsü	66
Şekil 5.1. Python ile anında GUI.....	71
Şekil 6.1. MCP3208 pin yapısı.....	77
Şekil 6.2. MCP23017 pin yapısı.....	78
Şekil 6.3. Sistemin elektronik altyapısı	78
Şekil 6.4. Kullanılan çevresel elemanların Raspberry Pi ile bağlantısı.....	79
Şekil 6.5. Web arayüzü giriş ekranı.....	80
Şekil 6.6. Karşılama ekranı	80
Şekil 6.7. Led kontrolleri ekranı.....	81
Şekil 6.8. Sistem üzerinde led kontrolleri	81
Şekil 6.9. Anlık sıcaklık değeri ekranı	82
Şekil 6.10. Sıcaklık grafiği ekranı	82
Şekil 6.11. Alarm ekranı pasif.....	83
Şekil 6.12. Alarm ekranı aktif	83
Şekil 6.13. Alarm ekranı – buton basıldı	84
Şekil 6.14. Analog giriş ekranı	84
Şekil 6.15. Mesaj iletim ekranı.....	85
Şekil 6.16. Sistem üzerinde mesaj iletimi	85
Şekil 6.17. Ayarlar ekranı.....	86
Şekil 6.18. Yeni kullanıcı adı ve şifre belirleme ekranı	86

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. Linux tabanlı geliştirme kartları	40
Tablo 4.1. I2C ve SPI seri veri yollarına genel bakış	60



SEMBOLLER LİSTESİ

- N_1 : Kabuk bölümün kırılma indisi
 N_2 : Öz bölümün kırılma indisi
 θ_i : Geliş açısı
 θ_r : Yansıma açısı



KISALTMALAR LİSTESİ

ARM	: Acorn Risc Machine – Advanced Risc Machine
ARP	: Adres çözümleme protokolü
ASCII	: Bilgi değişimi için Amerikan standart kodlama sistemi
CPU	: Merkezi işlem birimi
DCS	: Dağıtılmış kontrol sistemi
FTP	: Dosya aktarım protokolü
GPIO	: Genel amaçlı giriş/çıkış
HMI	: İnsan makine arayüzü
HTTP	: Hiper metin transfer protokolü
I2C	: İnter enterge devre
IP	: İnternet protokolü
JSON	: Javascript nesne gösterimi
LAN	: Yerel alan ağı
MAC	: Ortam erişim kontrolü
MIPS	: Saniyedeki milyon komut
OSI	: Açık sistemler ara bağlantısı
PC	: Kişisel Bilgisayar
PCB	: Baskılı devre kartı
PLC	: Programlanabilen Mantıksal Denetleyici
RARP	: Geri adres çözümleme protokolü
RTU	: Uzak terminal ünitesi
SCADA	: Merkezi denetleme kontrol ve veri toplama
SCL	: Seri saat sinyali
SDA	: Seri veri hattı
SMTP	: Elektronik posta aktarım protokolü
SPI	: Seri çevresel ara yüz
TCP	: Aktarım kontrol protokolü
UDP	: Kullanıcı veri bloğu protokolü
XML	: Genişletilebilir işaretleme dili

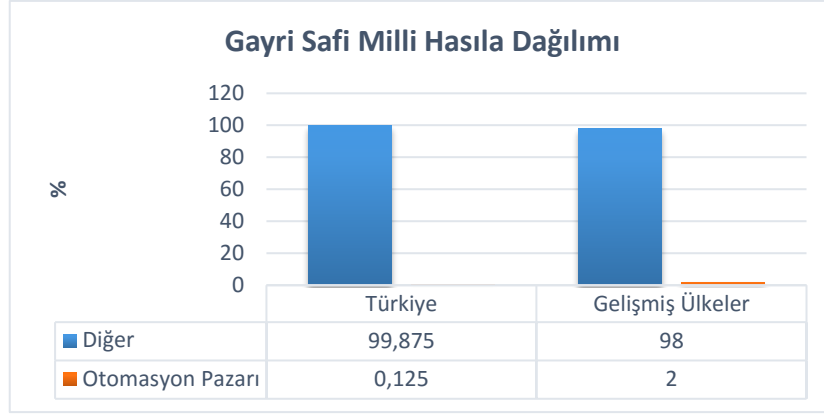
1.GİRİŞ

1.1. Tezin Amacı

Masaüstü bilgisayarların yoğun bir şekilde kullanıldığı 1990'lı yıllardan günümüze kadar olan süreç içerisinde, mikroişlemci teknolojisinin gelişmesiyle birlikte iş dünyasının ve güncel hayatın da çehresi değişmiştir. Bu cihazlar yerini zamanla, dizüstü bilgisayarlara, akıllı cep telefonlarına ve tablet bilgisayarlara bırakmıştır. Kullanılan tümleşik devre yapılarının kabiliyeti artmış ve hacimsel olarak da küçülerek kompakt bir hal almışlardır. Bütün bu değişimler, daha kolay, hızlı ve ucuz bir şekilde programlama yapabilme arzusunun getirisiidir.

Bu nedenle otomasyon sektöründe de kullanılan benzer teknolojilerin değişime ayak uydurması kaçınılmazdır. Özellikle gömülü sistem mantığına dayalı ve beraberinde büyük hacimli bilgisayarlar gerektirmeyen kontrolör yapılarının, mevcut otomasyon sistemlerine entegrasyonu, gerek ekonomik gerekse teknolojik anlamda yeni ufuklar açacaktır.

Modern otomasyon sistemlerinin ülkemize gelişi, gelişmiş ülkelere göre geç olsa da, Türkiye'nin global pazara açılması, ihracat hacminin artması ve uluslararası rekabette söz sahibi olması yadsınamaz bir gerçektir. Türkiye'de mevcut otomasyon şirketlerinin bir kısmı proses endüstrisine (kağıt-selüloz, petro-kimya, kimya, elektrik enerji üretimi, su-atık su, çimento, cam, tekstil, ilaç, maden-mineral, yiyecek-içecek) diğer kısmı ise kesikli üretime (makine, otomotiv, beyaz eşya ve ambalaj) yönelik çalışmaktadır. Bununla beraber kullanılan endüstriyel ekipmanlar yurtdışından ithal edildiği için ekonomik anlamda firmalara yük getirmektedir. Ülkemizde ekonominin dışa bağımlılığını azaltmak adına, sanayi ve dolayısıyla otomasyon sektörüne önem verilmelidir. Gelişmiş ülkelerin gayri safi milli hasıllarının ortalama %1-2 civarında değişen otomasyon pazarı Türkiye'de %0.125 gibi çok düşük bir düzeydedir. Bu dağılım Şekil 1.1' de gösterilmiştir.



Şekil 1.1. Türkiye ve gelişmiş ülkelerdeki otomasyon pazarının dağılımı

Bu problemin üstesinden gelmek için, öncelikle işe küçük ve orta ölçekli sanayicimizden başlamak gerekmektedir. Bu tez çalışmasında, ucuz, kullanıcı dostu, enerji ve maliyet açısından avantaj sağlayan bir prototip gerçekleştirilmiştir.

Endüstriyel uygulamalarda makineler ile insanlar arasında haberleşmenin gerçekleştirilmesi için bazı sistemler geliştirilmiştir. Bu sistemlerin en basit olanı butonlar ve lambalardan oluşmuş bir kontrol panosudur (Mimic Diyagram). Temel olarak kumanda mantığıyla çalışan bu sistemlerin özellikle karmaşık kablo hatları ve arıza verme sıklıklarının fazla olması gibi dezavantajları vardır. Bu sistemler yerini daha gelişmiş olan HMI (Human Machine Interface) ve SCADA (Supervisory Control and Data Acquisition) sistemlerine bırakmaya başlamıştır. Ancak bu uygulamalar PLC (Programmable Logic Controller) cihazlarına ihtiyaç duymaktadır. Bu sistemlerin kurulum maliyetlerinin oldukça yüksek olmasından dolayı ülkemizde halen birçok işletmede eski sistemler kullanılmaya devam etmektedir. Özellikle PLC ve SCADA sistemlerinin birlikte kullanıldıkları uygulamalarda bilgisayar kullanılması hem maliyet hem de bilgisayarların Şekil 1.2’de gösterildiği gibi zorlu endüstriyel koşullarda kullanılmasının doğuracağı problemleri de beraberinde getirmektedir.



Şekil 1.2. Zorlu endüstriyel koşullar altında çalışan bilgisayarlı sistemler

Endüstride kullanılan birçok cihaz insan kontrolü altında olması gerekmektedir. Endüstriyel süreç denetiminde basınç, sıcaklık, ağırlık, adet, hız, akım, gerilim vb. gibi ölçülen verilerin grafikleştirilmesi ve/veya görselleştirilmesi gerekmektedir. Bunun yanında ölçülen bu verilerin kayıt altında tutulması gerekmektedir. Ayrıca kullanıcının sistem parametrelerine müdahale etmesi gerekmektedir. Bazı durumlarda ise süreç denetimini sağlayan algorithmadan bağımsız olarak sistem manuel olarak çalıştırılabilmesi gerekmektedir. Tüm bunları yerine getirebilmek için SCADA sistemi kullanılmaktadır.

Kapsamlı, entegre bir kontrol ve gözetleme sistemi olan SCADA kontrol sistemi sayesinde, bir tesis veya işletmeye ait tüm ekipmanların kontrolü gerçekleştirilebilir, üretim planlaması kolaylıkla yapılabilir, çevre kontrol üniteleri ve yardımcı işletmeler dahil tüm birimlerin kontrolü ve gözlenmesi sağlanabilir. Ayrıca anlık olaylar ve alarmlarla ilgili veriler saklanarak geçmişte meydana gelen olaylar istenilen tarih ve saatte gözlemlenebilir [1,2].

SCADA sistemler, üç ana bileşene sahiptirler: uzak terminal birimleri, merkez istasyonlar ve haberleşme ortamları [3]. SCADA sistemi izleme, danışma, uzaktan kontrol ve sistemle ilgili veri toplama işlevlerini yerine getirir. Danışma ve uzaktan kontrol işlevi, belli cihazları veya tüm bir tesisi uzaktan kontrol edebilmek, bu cihazların kontrol konutlarına uygun bir şekilde çalışmasını sağlayabilmek ve davranışlarının yine kontrol komutlarına uygun olup olmadığını denetleyebilmektir. Danışmanlı kontrol sistemi, çoğullama tekniği kullanılarak bir iletişim kanalı üzerinden, geniş bir coğrafi bölgeye yayılmış ve birbirine uzak çok sayıda cihazın sistem operatörü tarafından işletilmesi olarak tanımlanır. SCADA sistemlerinden yararlanılarak uygulama yazılımları geliştirmek için protokollerin ve veri tabanı yapısının tanımlanmış olması gerekmektedir. SCADA işletmede bilgi omurgası görevini icra eder ve bu görevi yapması için iletişim protokolleri aracılığıyla kendi aralarında iletişim kurması gereken birimlerin haberleşmesini sağlamaktadır [4,5]. SCADA sistemlerinde kullanılabilecek birçok iletişim ortamı vardır [6,7].

Bu tez çalışmasında; WEB üzerinde çalışabilen bir SCADA yazılımı tasarlanmıştır. Tasarlanan SCADA sistemine ARM işlemci tabanlı bir elektronik devre ile erişim ve sistemin yönetilmesi sağlanmıştır. Sistemin kullanıcı tarafından kontrol edilebilmesi için WEB tabanlı bir yazılım hazırlanarak, sunucu tarafında ise WEB yazılımına bağlı çalışan bir sunucu yazılımı geliştirilmiştir. WEB üzerinden sistemi kontrol etmek isteyen istemci komutlarını, sunucu yazılımı yorumlayacak ve tasarlanan elektronik devreye aktaracaktır.

Bu sayede ucuz ve etkili bir şekilde web üzerinden kontrol edilebilen bir WEB-Tabanlı SCADA sistemi geliştirilmiş olacaktır. Ayrıca sistemde PC ve PLC kullanılmadan aynı işlevleri yerine getirebilen LINUX işletim sistemi kullanan bir ARM işlemci ile hem SCADA hem de kontrol senaryosu aynı işlemci üzerinde çalıştırılması sağlanmıştır. Mevcut otomasyon sistemlerinde kullanılan bilgisayarlı sistemlerin problemlerini ortadan kaldırarak çok daha ucuza ve sağlam otomasyon sistemlerinin kurulması hedeflenmiştir.

1.2. Tezin Organizasyonu

Tezin bölümleri ile ilgili kısa açıklamalar aşağıdaki gibidir.

İkinci bölümde, endüstriyel kumanda, kontrol ve veri iletişim sistemleri ile ilgili genel bilgilere yer verilmiştir. Daha sonra veri iletişim standartlarına kısaca değinilmiştir. PLC, HMI ve özellikle bu tezde de kullanılan SCADA gibi otomasyon sistemlerinin bileşenleri üzerinde durulmuştur ve ayrıca ARM tabanlı işlemciler ve ARM işlemcileri kullanan geliştirme kartlarına yer verilmiştir.

Üçüncü bölümde, tezde kullanılan web teknolojileri ile ilgili genel bilgilere yer verilmiştir.

Dördüncü bölümde, Raspberry Pi ile ilgili bilgiler yer almaktadır. Donanım özellikleri, kullandığı yazılım ve iletişim standartlarına değinilmiştir.

Beşinci bölümde, web tabanlı SCADA sistemi geliştirilirken kullanılan Python programlama dili ve bir web framework olan flask üzerinde durulmuştur. Ayrıca bu programlama dilinin uyumla çalışabildiği ve geliştirme kartının işletim sisteminin de bir türevi olduğu Linux'a değinilmiştir.

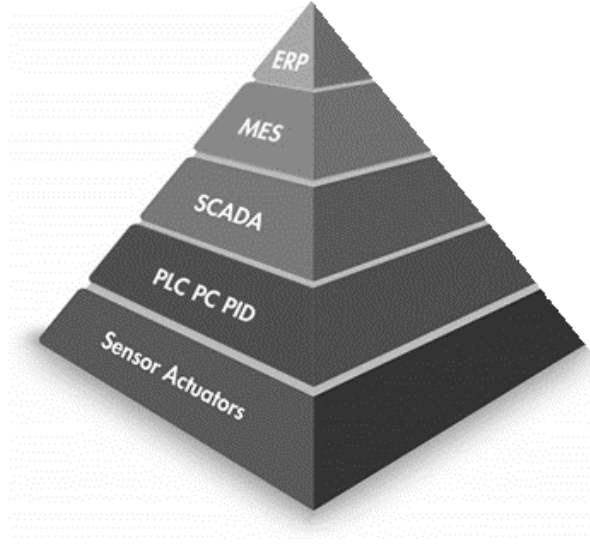
Altıncı bölümde ise oluşturulmuş olan SCADA sistemi ile ilgili bilgilere yer verilmiştir. Sistemin elektronik alt yapısı hakkında açıklamalar yapılmıştır ve bu kısımda kullanılan elektronik bileşenler ile ilgili bilgiler verilmiştir. Daha sonra sistem tasarlanırken ihtiyaç duyulan yazılımlara kısaca değinilmiştir. Ayrıca geliştirilen web tabanlı yazılımın tanıtımı yapılmıştır. Yine bu bölümde yazılımın kullanımı ile ilgili açıklamalar ve ekran görüntüleri yer almaktadır.

2. ENDÜSTRİYEL OTOMASYON

Günümüzde insan müdahalesine gerek duymayan, işlemleri otomatik olarak yapabilen, kararlı çalışan sistemler gün geçtikçe artmakta ve önem kazanmaktadır. Üretim sistemlerinde, işlemlerin bu şekilde istenildiğinde amaca uygun olarak kendi kendine yürütülmesi, kontrollerinin kolay ve hızlı bir şekilde sağlanması, sistemlerle ilgili verilere çabuk ulaşılması ve ilgili birimlere aktarılabilmesi amacıyla otomasyon sistemlerinden çoğu zaman faydalanılmıştır. Endüstri alanında çokça kullanılan bu sistemler genelde üç alt bölümden oluşur[8]. Endüstriyel Kumanda Sistemleri; bu tür sistemlerde birimlerin çalışma koşulları bazı mantıksal kurallara göre düzenlenir ve gerçekleştirilir. Kullanılan devrelerde çeşitli mantıksal işlevler, zaman ve sayaç işlevleri kullanılarak sistemin amacına uygun şekilde kumanda komutları üretilir. Endüstriyel Kontrol Sistemleri; endüstride sıkça kullanılan bu sistemlerde, amaç sistemin kararlı çalışmasını sağlamaktır. Kontrol sistemleri, herhangi bir nedenle sistemde meydana gelebilecek hataları kontrol ederek hatayı belirli sınırlar arasında tutmaya çalışır. Endüstriyel Veri İletişim Sistemleri; birimler arasındaki iletişimin güvenli ve hızlı bir şekilde sağlanmasını sağlayan sistemlerdir. Bu tür sistemler, SCADA gibi özel hazırlanmış yazılımlar sayesinde gerçek zamanlı işlemlerde, uzaktan kumanda edilen modüllerde ve kontrol işlemlerinde kullanılırlar [8].

Otomasyon denilince günümüzde akla ilk olarak PLC gelmektedir. Fakat otomasyon sistemleri esasında sadece PLC' den ibaret değildir. Aslında esnek bir üretim sistemi tasarlanmak istenirse, sistem çalışırken nelerin olup bittiği görülebilen, takip edilebilen, denetlenebilen, elde edilen verilerin gerekirse kaydedilebileceği ve kaynakların planlamasının yapılabilmesi bir mekanizma oluşturmak gerekir. Bu bağlamda otomasyon sistemlerini tasarlarlarken ERP/MES (Enterprise Resource Planning/ Management Enterprise System), SCADA, DCS (Distuributed Control Ssystem), PLC, RTU (Remote Terminal Unite), HMI, sensör ve aktüatörler gibi bileşenler kullanılmaktadır [9].

Günümüzde var olan otomasyon sistemlerine bakıldığında bileşenlerin Şekil 2.1' deki gibi gruplar oluşturduğu görülmektedir.



Şekil 2.1. Otomasyon sistemlerinin bileşenleri

Bir otomasyon sistemi bu bileşenlerin hepsini ihtiva edebildiği gibi sadece birkaçı da bulunabilir. Bu tez çalışmasında otomasyon sistemlerinde var olan bu bileşenlerin bazıları ve özellikle SCADA sistemler üzerinde durulmuştur.

2.1. Endüstriyel Veri İletişimi

Veri iletişimi noktadan noktaya bilgi transferlerini kapsar. Birçok iletişim sistemi analog verileri işler. Telefon sistemleri, radyo ve televizyon buna örnek olarak verilebilir. Modern cihazların neredeyse tamamı dijital veri transferi ile ilgilidir. İletişim sistemleri bilgi aktarımı için bir vericiye, bilgi alımı için bir alıcıya ve ikisi arasında bir bağlantıya ihtiyaç duyar. Bu bağlantı şekli kablolu veya kablosuz olabilir.

Dijital veri bazen öncelikle analog verileri işlemek üzere tasarlanmış bir sistemi kullanarak aktarılabilir. Örneğin bir modem, telefon kablolarından gelen analog verileri dijital verilere dönüştürerek çalışır. Diğer bir modem ise bu işlemin tersini yapar. Zaten modem kelimesi de modülatör ve demodülatör kelimelerinden türetilmiştir.

Alıcı ve vericiler, verinin nasıl kodlandığı konusunda bir uyum içerisinde çalışmalıdır. Alıcı mutlaka vericinin göndermiş olduğu verileri anlayabilmelidir. Cihazlar arasındaki iletişimi sağlamak amacıyla verileri düzenlemeye yarayan iletişim kurallarına protokol deniyor.

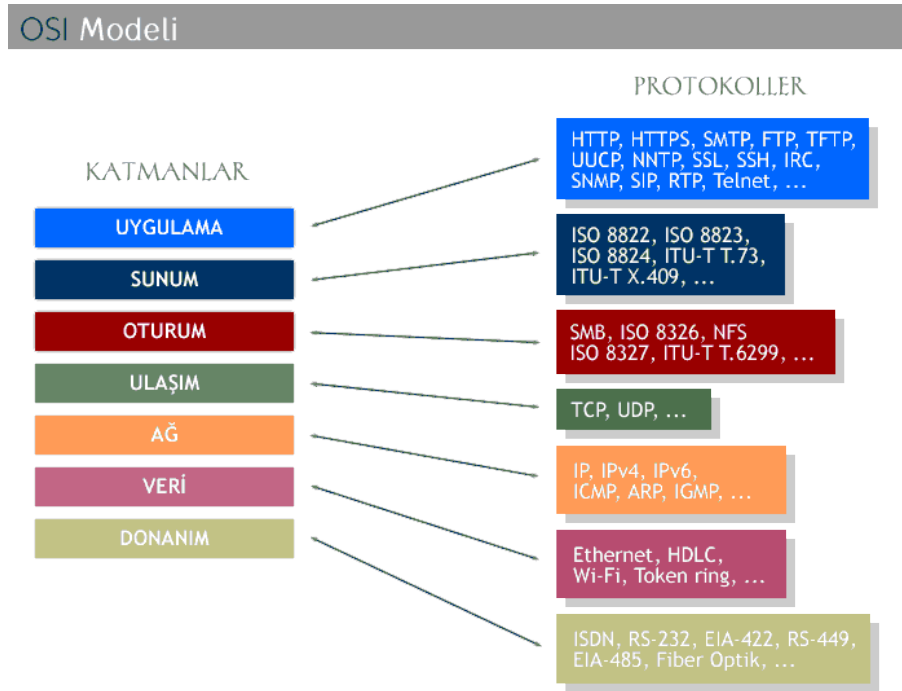
Ethernet standardı geçtiğimiz yıllarda büyük miktarda ilgi uyandırdı. Diğer bir protokol olan TCP/IP ise günümüzde popüler ve çokça kullanılmaktadır. Son zamanlarda

endüstriyel sistemler veri iletişimi üzerine kurulmaya başlamıştır. Özellikle endüstriyel otomasyon sistemlerinde kullanılan aygıtlar için veri haberleşme standartları geliştirilmiştir. Bunun yanında günlük yaşantımızda kullandığımız Ethernet haberleşme protokolleri de otomasyon cihazlarında popüler olmaya başlamıştır. Bu sayede Rj-45 standartlarındaki soketler yardımıyla birçok cihaz ekstra dönüşüm kitlerine ihtiyaç duymadan yerel hatlar üzerinden birbirlerine kolay bir şekilde bağlanabilmektedirler.

2.1.1. OSI Modeli

ISO (International Organization for Standardization) tarafından geliştirilen Open Systems Interconnection (OSI) modeli ile, bir ağa bağlanabilen farklı cihazların ve uygulamaların birbirleriyle iletişimi nasıl kuracakları tanımlanır. OSI Modeli, herhangi bir donanıma veya ağ tipine göre değişiklik göstermez.

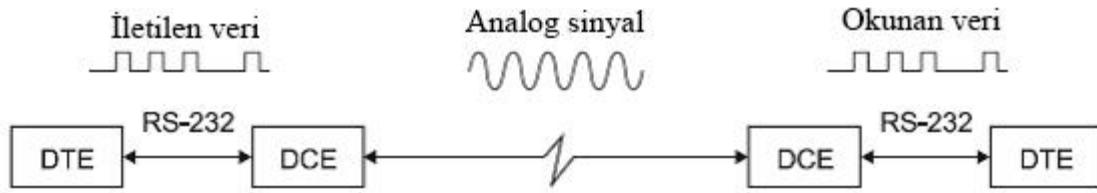
Ağ üzerinde iletişim sağlarken izlenen karmaşık yolu sadeleştirmek ve belirli görevlerde uzmanlaşılmasının sağlanması amacıyla ISO standardı yedi alt göreve (katman) ayrılmıştır. Şekil 2.2’ de OSI modeli olarak tanımlanan bu yedi katman gösterilmiştir [13].



Şekil 2.2. Open Systems Interconnection (OSI) modeli

2.1.2. RS-232 Arayüz Standardı

RS-232 bir seri iletişim standardıdır. Bu standart, veri terminal ekipmanı (DTE) ile veri taşıma ekipmanı (DCE) arasındaki seri ikili tek sonlu veri iletimi için kullanılır. RS-232 daha çok bilgisayarda mevcut olan seri portlarda kullanılır. Bu standart, sinyal anlamlarını ve zamanlamalarını, elektriksel karakteristikleri, bacak çıkışlarını ve konnektörlerin fiziksel büyüklüklerini kapsamaktadır. Tipik bir seri veri iletişim bağlantısı Şekil 2.3' te gösterilmiştir.

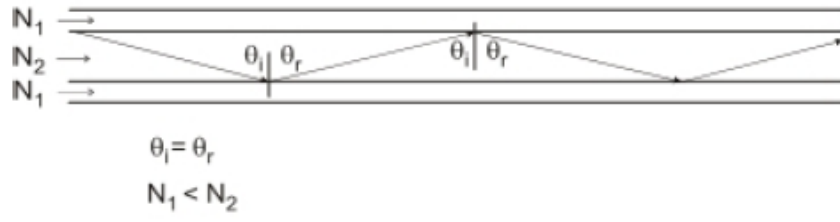


Şekil 2.3. Tipik bir seri veri iletişim bağlantısı

2.1.3. Fiber optik

Fiber optik haberleşme, fiber bir çekirdek boyunca güdümlü ışık sinyallerini kullanır. Fiber optik kablolar ışık için birer dalga kılavuzudur. Bütün enerji kablonun merkez çekirdeği boyunca güdümlüdür. Işık, iç kısımdaki yansımalar aracılığıyla fiberin merkezinde bir dalga kılavuzu şeklinde ilerlemektedir. Çapraz modları veya çoklu yayınma hatlarını destekleyen fiberler çok modlu fiberler olarak adlandırılır. Eğer fiber sadece tek bir modu destekliyorsa bu tür fiberlere ise tek modlu fiberler denilir. Geniş çaplı bir merkeze sahip olan çok modlu fiberler, genellikle daha çok gücün iletilmesine ihtiyaç duyulan kısa mesafeli iletişim hatlarında tercih edilirler. Tek modlu fiberler ise uzun mesafeli iletişim hatları için tercih edilirler [14].

Çekirdek ve kaplama, ışını çekirdek içerisinde tutar. Işın, çekirdeğe kritik açıdan daha büyük bir açıyla girer. Işın daha sonra Şekil 2.4' teki gibi fiberin çekirdeği boyunca çok düşük bir güç kaybıyla yoluna devam eder.



Şekil 2.4. Işın bir optik fiber boyunca hareket ediyor.

Sağladıkları avantajlar açısından, optik tabanlı haberleşme sistemleri kısa süre içerisinde geniş kullanım alanı bulmuştur. Fiber optik haberleşmenin kullanım alanları aşağıdaki gibidir:

- TV sistemleri
- Veri iletimi
- Elektronik cihazların birbirleriyle iletişimi
- Yüksek gerilim hatları
- Askeri bağlantılar
- Trafik kontrolleri
- İnternet bağlantıları
- Vücut iç organlarının görüntülenmesi

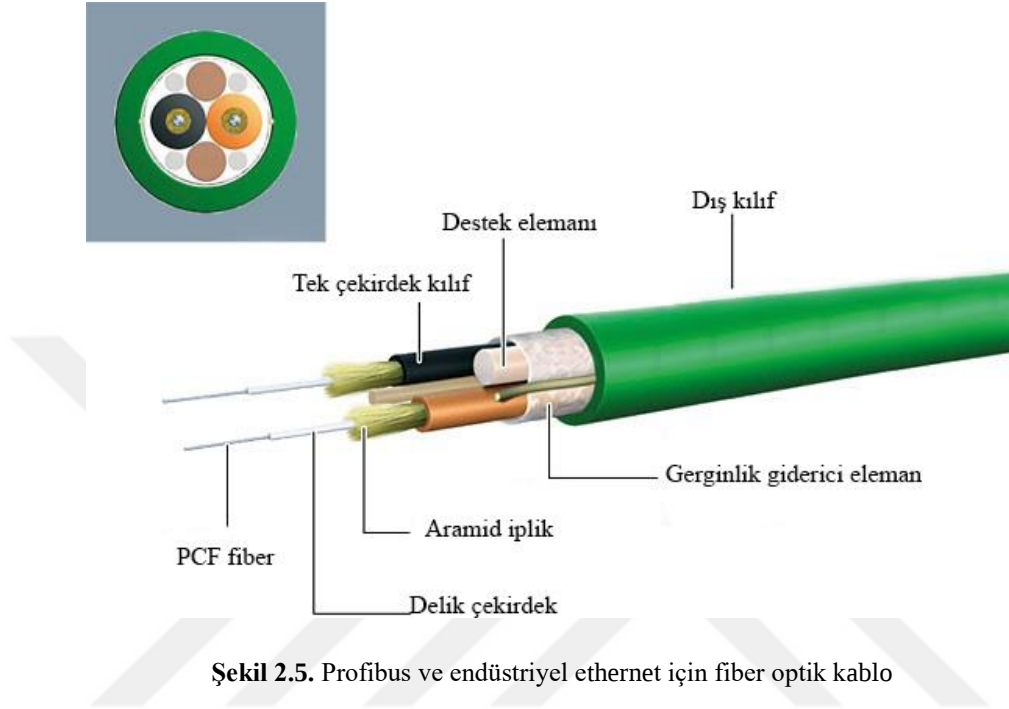
Fiber optik kabloların avantajları aşağıdaki gibidir:

- Band genişliği ve taşıma kapasitesi yüksektir.
- Farklı çap ölçülerine sahiptirler.
- Çok hafiftir.
- Veri iletim mesafesi çok fazladır.
- Kayıpları çok azdır.
- Bilgi hırsızlığı yapılması çok zordur.

Fiber optik kabloların dezavantajları aşağıdaki gibidir:

- Maliyetleri yüksektir.
- Detaylı işçilik gerektirir.
- Yüksek çıkış gücü yoktur [14].

Son zamanlarda fiber optik bağlantılar geniş çaplı endüstriyel uygulamalarda yüksek hız ve mesafe özelliğinden dolayı tercih edilmeye başlanmıştır. Şekil 2.5’ te endüstride kullanılması için tasarlanmış bir fiber optik kablounun iç detayları gösterilmiştir.



Şekil 2.5. Profibus ve endüstriyel ethernet için fiber optik kablo

2.1.4. Modbus

Modbus haberleşme protokolü, farklı türdeki ağlara bağlı cihazlar arasında istemci/sunucu iletişimini sağlayan bir uygulama katman (OSI katmanı 7) protokolüdür. Modbus haberleşme protokolü sadece bir protokoldür ve özel bir uygulama donanımı manasına gelmez.

PLC’ ler arasında seri haberleşme protokolü olarak kullanılan ve MODICON firmasının ortaya çıkarmış olduğu Modbus’ ın basit ve güçlü bir yapısı vardır ve bu sayede endüstri alanında en çok kullanılan protokoldür. Farklı cihazlar arasında haberleşmeyi sağlayabilir, sunucu-istemci tabanlıdır. Böylelikle bir cihazdan alınan veriler bir veri merkezinde toplanabilir. Dolayısıyla veriler bir cihazdan alınıp bir veri merkezinde toplanabildiği ağ sistemidir. Üreticilerin herhangi bir ücret ödmeden kullanabileceği açık bir protokoldür. Modbus, çeşitli aygıtları programlamak ve master/slave sistemlerini gözlemlemek için kullanılır. Bir Modbus ağında, 1 master ile birlikte 247’ ye kadar slave cihaz bulunabilir.

Master cihaz, slave cihazlardan veri alabilir ve bu verilerle slave cihazları yönetebilir ve bu cihazlara veri yazılmasını sağlayabilir.

Modbus, kısa mesafeli bağlantılar için RS232 seri haberleşme standardını kullanır ve uzun mesafeli bağlantılar için ise RS485 standardını kullanır. Veriler bu protokolde 0 ve 1'lerden oluşan bitler halinde taşınır. Algılayıcılar ve diğer cihazlar arasında bağlantı kurmak veya belirli bir alan içerisindeki cihazları uzaktan kontrol etmek için kullanılabilir. Esnekliği yüksek olan Modbus, gelişmiş sensörlerle birlikte de kullanılmaktadır.

Modbus haberleşme protokolünde iki seri iletim modu vardır. Bunlar ASCII ve RTU iletim modlarıdır. Denetleyiciler standart Modbus ağlarında iletişim kurmak için bu iki modu kullanarak kurulabilir. Her denetleyicinin yapılandırmasında kullanıcılar seri port iletişim parametreleriyle (baud hızı, parite modu, vb) birlikte istenilen modu seçerler. Modbus ağındaki tüm cihazlar için mode ve seri parametreler aynı olmalıdır.

ASCII veya RTU modunun seçimi sadece standart Modbus ağlarıyla ilgilidir. Bu da bu ağlar üzerinde seri iletilen mesaj alanlarının bit içeriklerini tanımlar ve bilginin mesaj alanlarına nasıl paketleneceğini ve çözümleneceğini belirler [15].

ASCII modunda; bir Modbus ağında denetleyiciler, iletişim için ASCII (American Standard Code for Information Interchange) modunu kullanarak kurulduğunda, bir mesajdaki her 8 bitlik bayt iki ASCII karakter olarak gönderilir. Bu modun ana avantajı bir hataya neden olmadan bir saniyeye kadar karakterler arasında olan zaman aralıklarına izin vermesidir [15].

ASCII modundaki her bayt formatı:

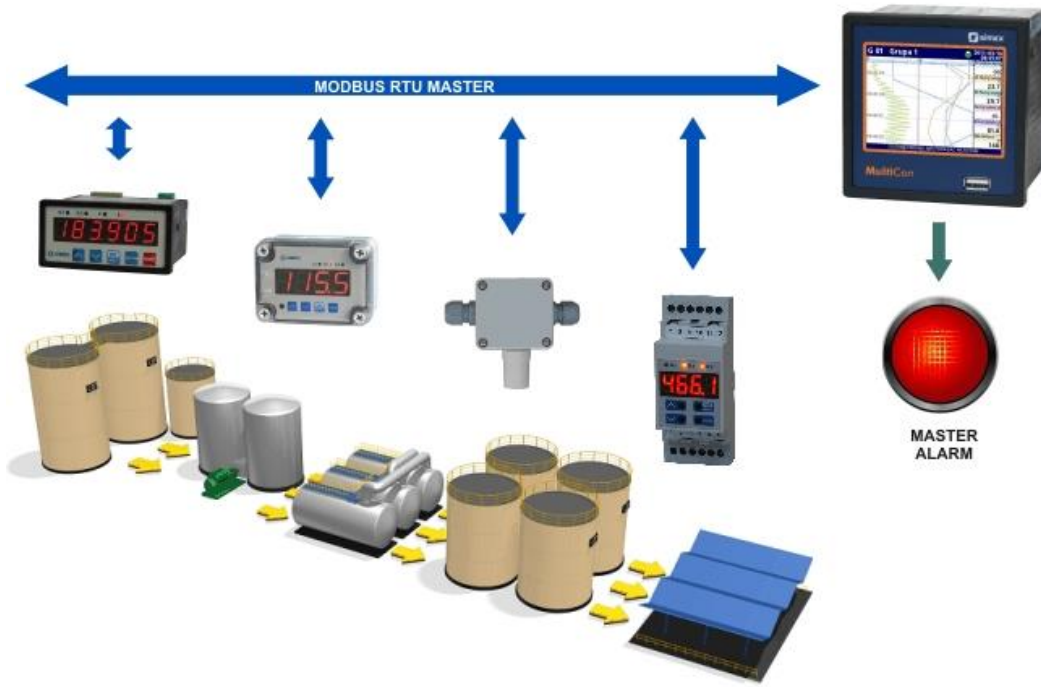
- Kodlama Sistemi: on altılı, ASCII karakterleri 0-9, A-F
- Mesajın her ASCII karakteri bir on altılık karakter içerir.
- Bayt başına bit: 1 başlangıç biti
- 7 veri biti
- Eğer varsa 1 parite biti
- Parite kullanılırsa 1 durdurma biti; parite yoksa 2 bit
- Hata Kontrol Alanı: LRC (Longitudinal Redundancy Check)

RTU Modunda; bir Modbus ağında denetleyiciler, iletişim için RTU (Remote Terminal Unit) modunu kullanarak kurulduğunda, bir mesajdaki her 8 bitlik bayt iki adet 4 bitlik onaltılık karakter içerir. Bu modun ana avantajı yüksek karakter yoğunluğunun aynı baud hızı için ASCII' den daha fazla veri iletimine izin vermesidir. Her bir mesajın sürekli bir akış içinde aktarılması gerekir [15].

RTU modundaki her bayt formatı:

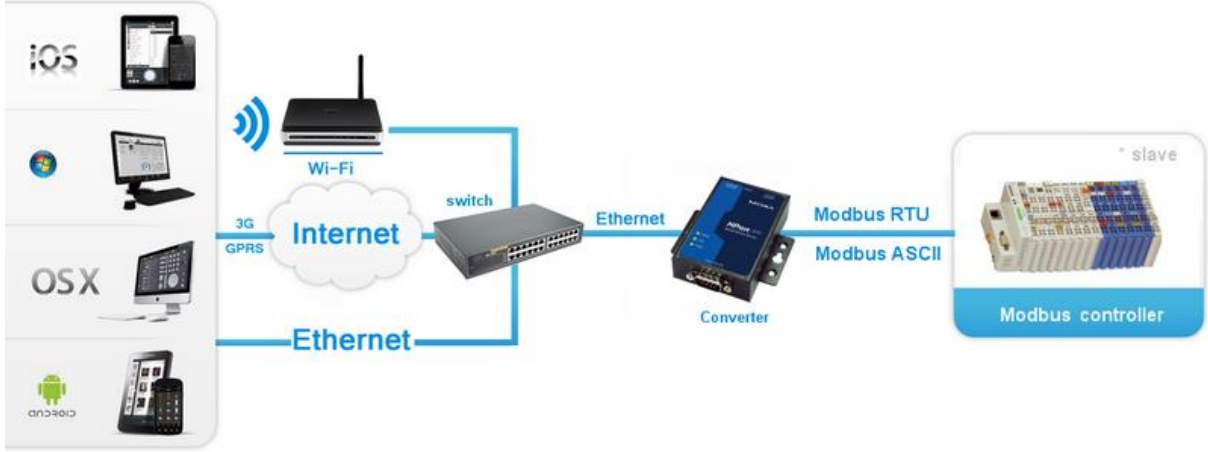
- Kodlama Sistemi: 8-bit ikili, on altılı 0-9, A-F
- Mesajın her 8 bitlik alanı iki on altılık karakter içerir.
- Bayt başına bit: 1 başlangıç biti
- 8 veri biti
- Eğer varsa 1 parite biti
- Parite kullanılırsa 1 durdurma biti; parite yoksa 2 bit
- Hata Kontrol Alanı: CRC (Cyclical Redundancy Check)

Şekil 2.6' da MultiCon firmasının Modbus Master haberleşmesine ait bir uygulama çizimi gösterilmiştir. Farklı özelliklere sahip cihazlar aynı haberleşme protokolünü kullanarak ana merkeze bilgi transfer edebilir.



Şekil 2.6. MODBUS haberleşmesine ait bir uygulama

Ayrıca bazı durumlarda haberleşme protokolleri arasında dönüşüm yapmak gerekebilir. Bu gibi durumlarda Şekil 2.7' de gösterildiği gibi dönüştürücü (converter) kullanılabilir. Böylece daha esnek ve kullanışlı bir veri paylaşımı elde edilmiş olunur.



Şekil 2.7. Modbus – Ethernet dönüştürücü

2.1.5. Data Highway Plus /DH485

Allen Bradley veri iletişimlerinde kullanılan üç ana yapılandırması vardır:

Data Highway: Bu, 64 düğüme kadar noktadan noktaya haberleşmeye izin veren bir yerel alan ağıdır. Yarı çift yönlü bir protokol kullanır. 57.6kbaud’ da çalışır.

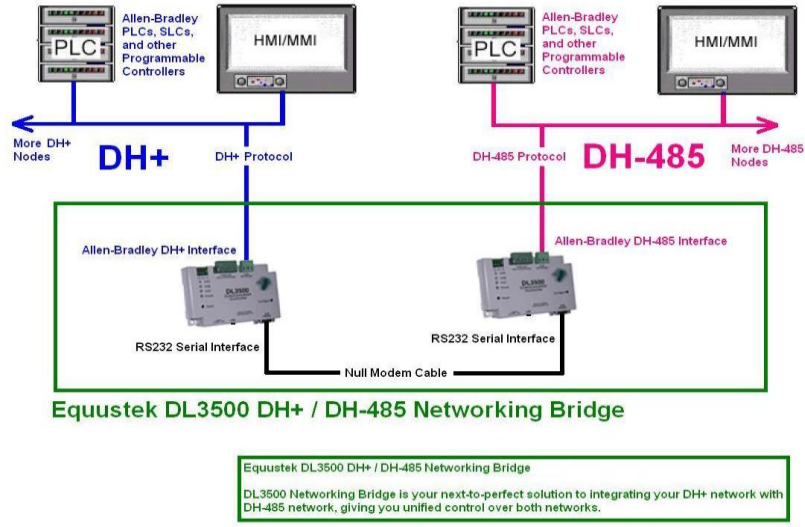
Data Highway Plus: Daha az düğüm için tasarlanmasına rağmen Data Highway ağına benzerdir ve 57.6kbaud veri hızında çalışır. Belirli bir düzende uçlar arasındaki bağlantıyı döndüren bir jetonlu noktadan noktaya iletişimleri vardır.

The Allen Bradley Data Highway plus, OSI katman modelinin üç katmanından oluşur.

- Fiziksel katman donanım
- Veri bağlantı katmanı protokolü
- Uygulama katmanı protokolü

Data Highway-485: Allen Bradley denetçilerinin SLC serisi tarafından kullanılır ve RS-485 tabanlıdır.

Allen Bradley firmasına ait Data Highway haberleşmesi Şekil 2.8’ de gösterilmiştir.

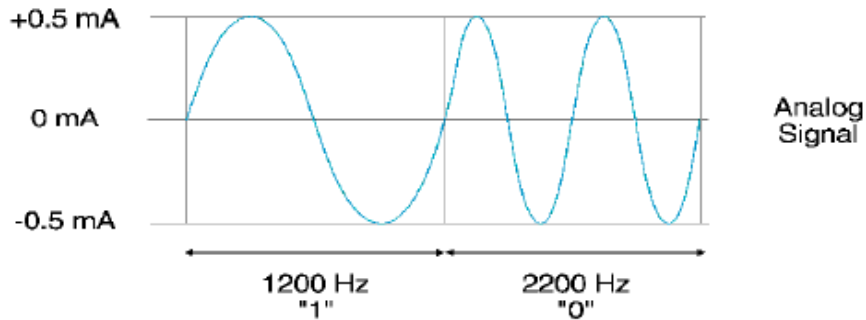


Şekil 2.8. Allen Bradley firmasına ait Data Highway haberleşmesi

2.1.6. HART

İngilizce “Highway Addressable Remote Transducer” ve kısaca HART olarak bilinen bu protokol, analog kablolar kullanarak akıllı sistemler ve izleme sistemleri arasında dijital bilgiler göndermemizi ve bu bilgileri almamızı sağlar. Bu protokol ile ana sistem ve bağlı cihazlar arasındaki veri iletişimi çift yönlü olarak sağlanır. Kullanılan fiziksel katmana bağlı olarak iletim hızı değişkenlik gösterir. 4-20 mA üzerinden kablolama gereçleriyle iletişim kurabiliyor olması en önemli özelliklerindendir.

HART, yalnızca dijital olan birçok sistemin aksine analog ve dijital hibrid bir sistemdir. Bell 202 standardı tabanlı bir FSK (Frequency Shift Keying) tekniği kullanır. Şekil 2.9’deki gibi sayısal 1 ve 0’ ları temsil eden 1200 ve 2200 Hz olmak üzere iki ayrı frekans kullanılır. 4-20mA sinyal üzerine bindirilmiş 1200 / 2400Hz sinüs dalgası ortalama değeri sıfırdır; 4-20mA analog bilgiler akım kontrollü olduğundan dolayı etkilenmez.



Şekil 2.9. HART sinyalizasyon sisteminin frekans dağılımı

HART üç şekilde kullanılabilir:

- Noktadan-noktaya modunda 4-20 mA akım sinyali ile bağlantılı olarak
- Multi-drop modunda diğer alan cihazları ile birlikte
- Yayın yapan sadece bir alan cihazıyla noktadan noktaya modda

2.1.7. AS-i

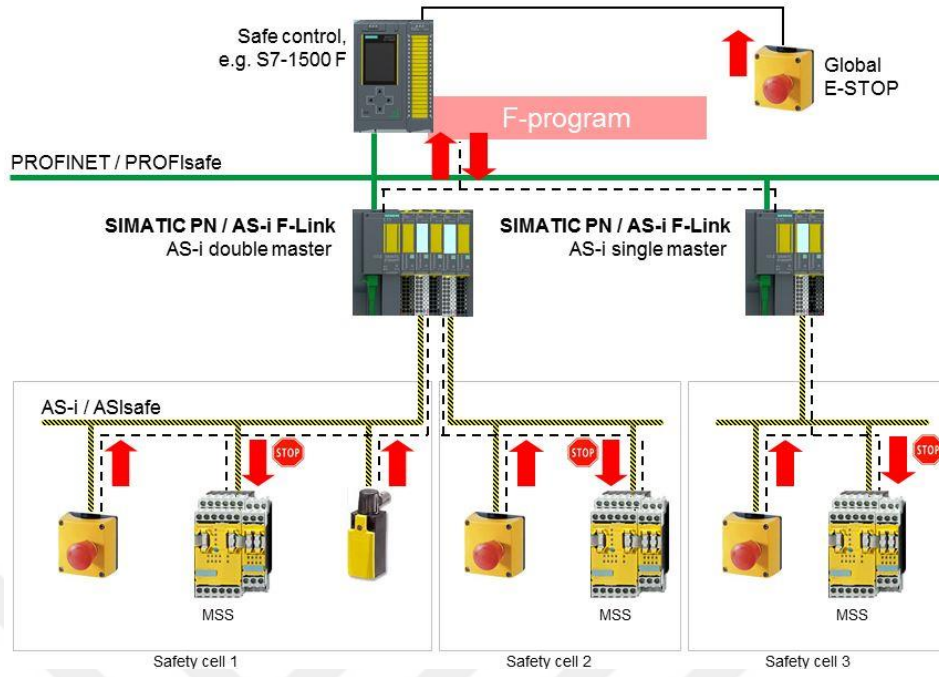
AS-Interface, küresel standartta bir fieldbus/haberleşme sistemidir. Bu sistem sensör ve aktüatörleri, ortak bir veri iletim yolu üzerinden, kontrolöre bağlar. AS-i, "Aktüatör-Sensör Arayüzü" için kullanılan bir kısaltmadır. Bu arayüz otomasyon sistemleri içerisindeki en alt işlem seviyesindeki modülleri birbirine bağlar. Geleneksel kalabalık kablo yapıları, burada tek bir kablo - AS-i Kablosu - ile değiştirilir. Bu kablo hem veriyi hem de gücü taşır.

Dünyada önde gelen üreticiler, bu kolay kablolama sistemini desteklemektedir. Bu sistem dört ana bileşenden oluşur: Kontrolör, Modüller, Güç Kaynağı ve AS-i Kablosu.

Her modülün, kontrolör ile giriş ve çıkış sinyalleri arasındaki veri alışverişi için, kendine ait özel bir adresi vardır. İğneli bağlantı teknolojisi, modüllerin hızlı ve kolay bir şekilde AS-i kablosuna takılabileceği veya kablo üzerinde farklı bölgelere taşınabileceği anlamına gelir. AS-i kablosunun silikon yapısı yalıtım gerekliliklerini de ortadan kaldırmaktadır. Teşhis Cihazları ya da yerleşik teşhis fonksiyonları içeren kontrolörler, sorun gidermeyi kolaylaştırır. Arızalı modüller hızla tespit edilebilir, değiştirilebilir ve kontrolör tarafından otomatik olarak yeniden adreslendirilir.

Kısacası; AS-i, kurulum ve bakım için çok az bir çaba gerektiren; zaman ve para tasarrufu sağlayan bir sistemdir. Ayrıca, uzmanlık gerektirmeden kurulup taşınabilmesi özelliğiyle de en kolay fieldbus sistemlerinin başında gelir [16].

Özellikle montaj ve bakımda sağladığı avantajlardan dolayı sistemdeki çevre birimleri AS-i haberleşme üzerinden birbirine bağlanabilir. Bu durum Şekil 2.10' da gösterilmiştir.



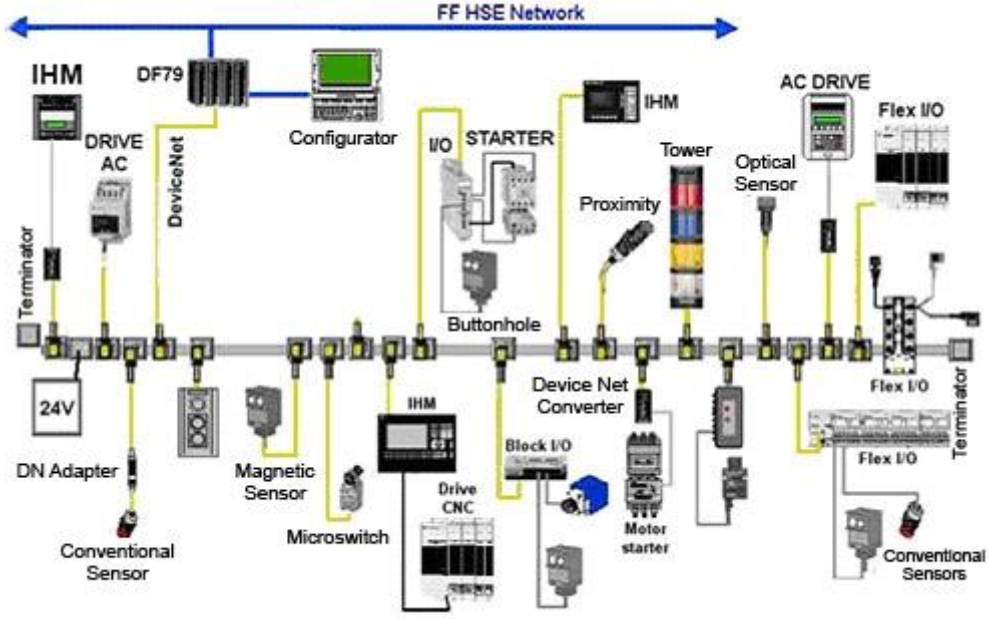
Şekil 2.10. Siemens firmasına ait bir As-i bağlantı şeması

2.1.8. DeviceNet

AllenBradley tarafından geliştirilen ve endüstride kullanılan bir haberleşme protokolüdür. Kurulum maliyetlerini düşürmek, harcanan zamanı azaltmak ve bir tek ağ ile çeşitli cihazları kontrol etmek için CAN mimarisi üzerine inşa edilmiştir. Düşük maliyetli, açık ve güvenilir bir ağ teknolojisidir. Endüstride hem güvenilirlik hem de performans gerektiren uygulamalar için tasarlanmıştır. General Motors, Motorola, Nestle, bu protokolü kullanan ünlü kuruluşlardan bir kaçıdır [17].

DeviceNet ile hem barkod okuyucular, fotoelektrik sensörler ve motor starterleri gibi düşük seviyeli cihazlara bağlanılabilir hem de PC veya PLC gibi daha üst seviyeli cihazlarla haberleşme sağlanabilir [18].

DeviceNet haberleşme protokolünü destekleyen birçok algılayıcı cihaz mevcuttur. Bu cihazlar bir haberleşme ağı üzerinden bağlanarak ana üniteye bilgi gönderebilir. Şekil 2.11' de gösterildiği gibi farklı işlemlerden bilgi transferi yapmak mümkündür.



Şekil 2.11. DeviceNet haberleşme ağı örneği

2.1.9. Profibus

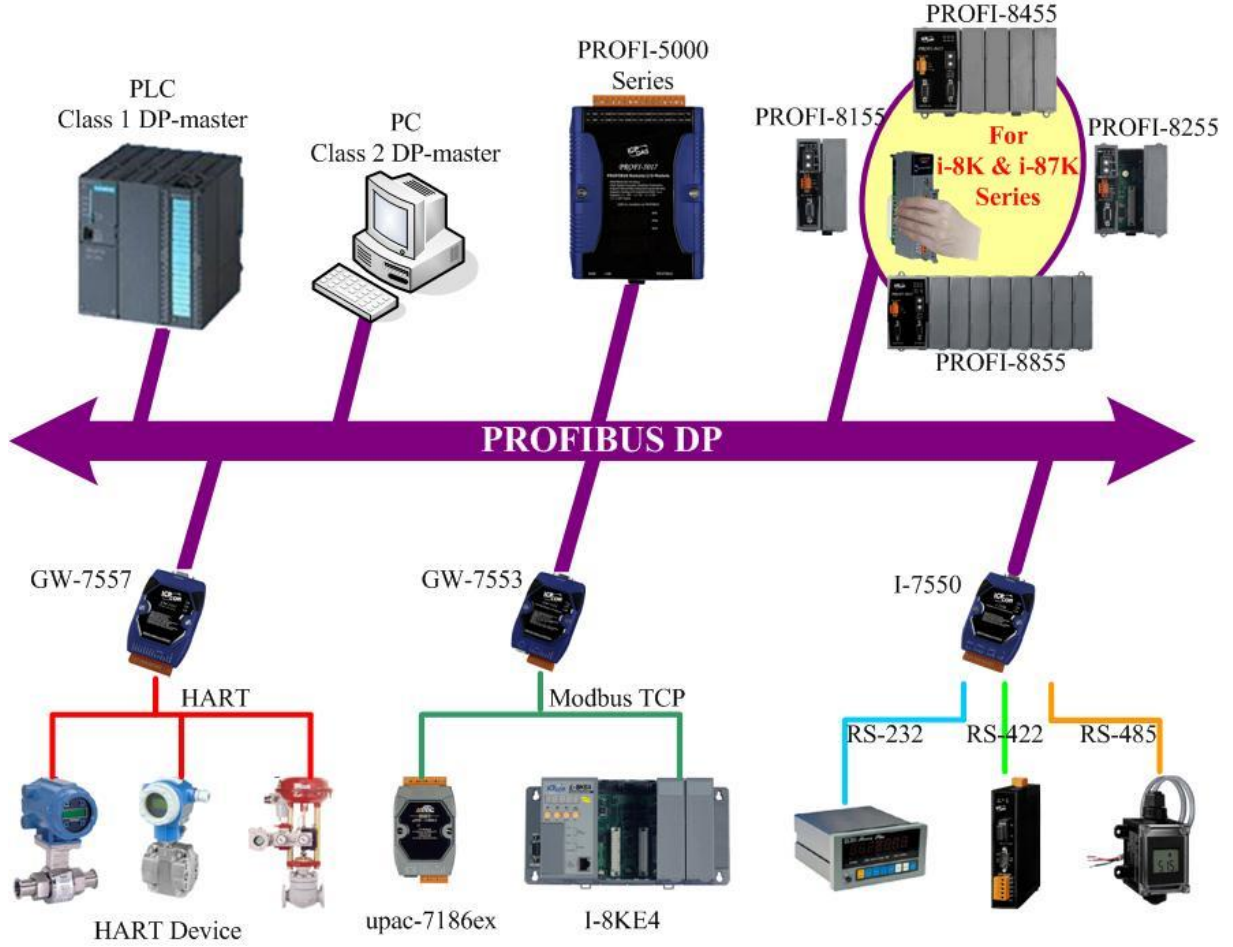
Profibus, üreticiden bağımsız bir açık saha hat protokolüdür. Geniş kapsamlı üretim ve işlem otomasyonları için tasarlanmıştır. Birçok üyesi bulunur ve yine birçok araştırma enstitüsü bu protokolü desteklemektedir. Uluslararası standartlar olarak bilinen EN 50170, EN 50254 ve IEC 61158 üzerine kurulan Profibus, farklı üreticilere ait cihazlar arasındaki haberleşmeyi dahi sağlayabilir ve bu haberleşmeyi sağlarken herhangi bir özel arabirime ihtiyaç duymaz. Hızlı veri alışverişini destekleyen yapısından dolayı yüksek hızlı uygulamalar veya karmaşık haberleşme işlemlerinde yaygın olarak kullanılan bir veri yolu sistemidir [18].

Profibus;

- İşlem otomasyonları ve özel üretim uygulamaları için tasarlanmıştır.
- Siemens gibi büyük PLC üreticileri tarafından desteklenmiş ve standart olarak kabul edilmiştir.
- Çalışma esnasında çevre birimleri bağlanabilir.
- Veri aktarımı kablo veya optik iletkenler aracılığıyla yapılabilir.
- Veri iletimi elektrik kabloları ile 12 km' ye kadar sağlanabilir, optik kablolar ile bu değer yaklaşık 24 km' dir.
- Farklı üreticilerin cihazları arasında haberleşme yapılırken bile herhangi bir özel eklentiye gerek olmaz.

- Özel arabirimlere gerek duyulmayan bir veri yolu kullanılır.
- Karmaşık haberleşme işlemlerinde ve yüksek hız gerektiren uygulamalarda kullanılabilir [17].

Şekil 2.12' de Profibus ile ilgili bir haberleşme örneği gösterilmiştir.



Şekil 2.12. Profibus haberleşme örneği

2.1.10. Foundation Fieldbus

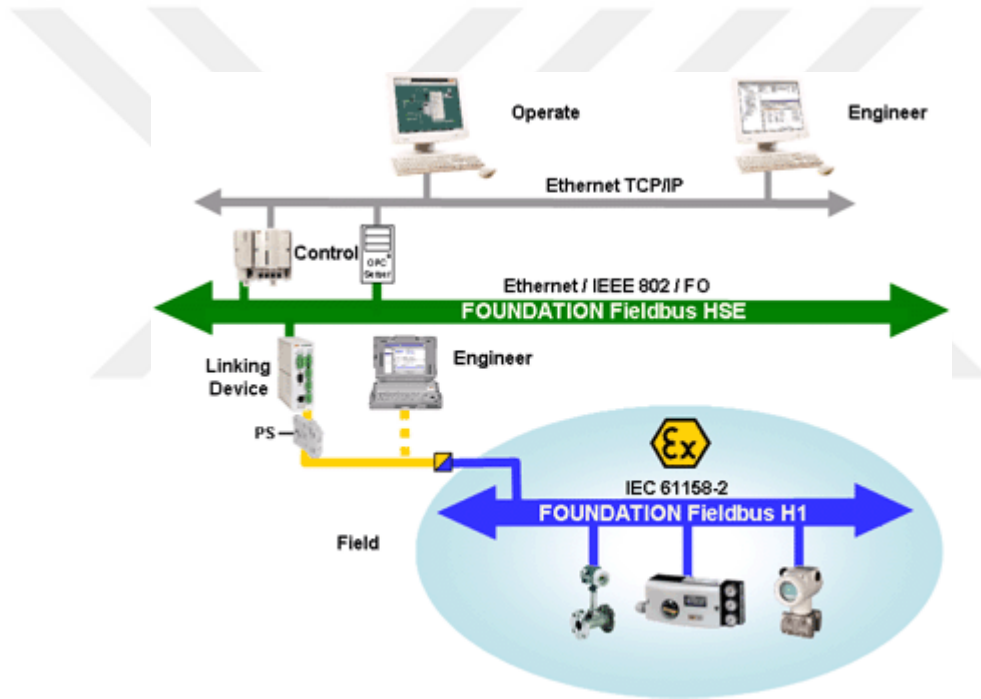
Foundation Fieldbus esas amacı, yeni dijital teknolojilerin avantajlarını kullanırken mevcut 4-20mA standardının beğenilen özelliklerini korumaktır.

Foundation Fieldbus son kullanıcıya şu faydaları sağlar:

- Azaltılmış kablolama
- Tek bir araçtan çoklu işlem değişkenlerinin iletişimi

- Gelişmiş teşhis
- Farklı üreticilerin cihazları arasındaki birlikte çalışabilirlik
- Geliştirilmiş alan seviyesi kontrolü
- Düşük başlangıç zamanı
- Basit entegrasyon.

Foundation Fieldbus dört OSI katmanını kullanır. Bunlardan üçü OSI katmanlarından 1,2 ve 7' ye karşılık gelmektedir. Dördüncü ise yedinci katmanın üzerine yerleşen kullanıcı katmanıdır ve çoğu kez OSI katmanlarının sekizincisi olarak söylenir. Kullanıcı katmanı uygulama yazılımı ve gerçek saha cihazları arasında standart bir arayüz sağlar. Bir Foundation Fieldbus haberleşme örneği Şekil 2.13' te gösterilmiştir.



Şekil 2.13. Foundation Fieldbus haberleşme örneği

2.1.11. Endüstriyel Ethernet

Ethernet önceleri bilgisayarların birbirlerine eş eksenli bir kablo aracılığıyla haberleşmesi fikrine dayalıydı. Bunu yaparken yayın iletimi yöntemi kullanılıyordu ve bu da radyo sistemlerine benzemekteydi. Üstelik yayın iletimi yöntemini kullanan bir sistemde, sistem kablolu ise çakışmaları saptamak radyo yayınına göre çok daha kolaydı.

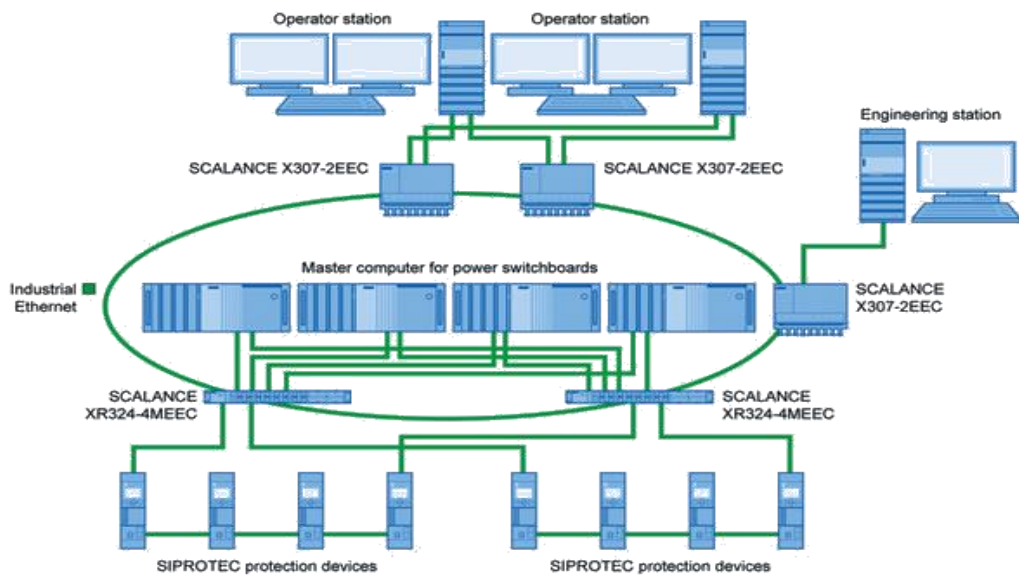
Ancak ethernet bu haliyle çok basit ve geliştirilmesi gereken bir kavram olarak karşımıza çıkmaktadır. Daha sonra karmaşık ağ teknolojilerini destekleyen ve günümüzde

pek çok yerel ağ altyapısını oluşturan yapıya evrimleşmiştir. Sonraları eş merkezli kablolar yerini çeşitli ağ anahtarlarına ve ethernet hubları kullanan noktadan noktaya bağlantılara bırakmıştır. Bu şekilde kurulum masrafları azalmış, güvenilirlik artmış, arıza tespiti ve bir noktadan diğerine ağ yönetimi kolaylaşmıştır. Ayrıca bükülü tel çifti kabloların kullanılmasıyla eski ethernet teknolojileri ve benzer diğer teknolojilere göre kurulum masrafları azalmıştır.

Ethernet istasyonları, birbirleriyle donanım katmanı üzerinden veri paketleri göndererek haberleşir. Bu paketler ayrı ayrı alınıp gönderilir. Diğer IEEE 802 teknolojilerindeki gibi her ethernet istasyonu için 48 bitlik bir MAC adresi vardır. Bu adres veri paketlerinin gönderme ve alma adreslerini belirlemek içindir. Ağ bağdaştırıcı kartları da genelde kendilerine özel bir global adrese sahiptir.

Gelişim sürecine 10 Mbit/s hızındaki eş merkezli kablolarla başlayan ve noktadan noktaya bağlantılarda 1 Gbit/s hızına ulaşabilen tüm ethernet teknolojileri aynı veri çerçeve formatını kullanmalarından dolayı birbirlerine bağlanabilirler.

Günümüzde donanım maliyetlerinin düşmesi, Ethernet arayüzünün fazla yer kaplamaması gibi nedenlerle ethernet oldukça yaygınlaşmıştır ve pek çok bilgisayar üreticisi ana kartlarına ayrıca bir ağ bağdaştırıcısı kartına gerek olmadan tümleşik ethernet arayüzü koymaktadır [19]. Endüstriyel ethernet ağı ile ilgili bir örnek Şekil 2.14’ te gösterilmiştir.



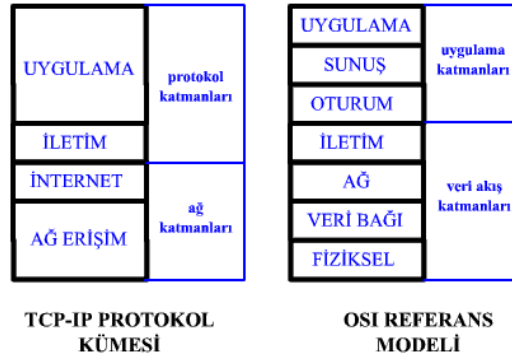
Şekil 2.14. Endüstriyel ethernet ağı ile ilgili bir örnek

2.1.12. TCP/IP

TCP/IP, TCP iletim kontrol protokolü (Transmission Control Protocol) ve IP internet protokolü (Internet Protocol) olmak üzere iki protokol kümesinden oluşmuştur. Bu iki protokol dünya üzerindeki tüm ağları birleştirmek amacıyla oluşturulmuştur. Bu şekilde oluşan uluslararası ağa internetwork veya kısaca internet denilmektedir. İki protokolün de kullanıldıkları ağ içerisinde farklı görevleri vardır. TCP, üst katmandan, yani uygulama katmanından gelen verinin parçalara ayrılması, ağ ortamında kaybolan veya bozulan verinin tekrar gönderilmesi ve karışık gelen verilerin belirli bir düzene sokulması gibi işlevleri yerine getirir. IP ise TCP tarafından segmentlere ayrılmış veriyi uygun bir yol bulup ulaşmak istediği adrese yöneltme görevini yerine getirir. TCP' den gelen veri parçalarına gidilecek adres bilgilerinin bulunduğu bir başlık bilgisi eklenir. Bu şekilde IP başlığı eklenmiş veri paketlerine datagram denir.

TCP/IP aslında OSI referans modeline benzer ama genel olarak biraz daha sadeleştirilmiş halidir. Daha az katmana sahiptir ve daha basit bir yapısı vardır. Ayrıca gerçek hayatta kullanılan yapısıyla OSI referans modelinden ayrılır. OSI daha çok iletişimde standardı belirler. TCP/IP ise uygulamaya yöneliktir.

Bu protokolde uygulama, iletim, internet ve ağ erişimi olmak üzere dört katman vardır. Uygulama katmanı; TELNET, FTP, SMTP, HTTP gibi internet ortamında kullanılan tüm protokolleri kapsar. İletim katmanının; hata düzeltme, iletim kalitesi ve akış kontrolü gibi görevleri vardır. İletim katmanında iki farklı protokol çalışır. Bunlar TCP ve UDP' dir. İnternet katmanı; kaynak ağdan gönderilecek bilgilerin adreslenmesi ve internet ortamına gönderilmesi ile ilgilidir. Bu katmanda IP, ARP ve RARP gibi protokoller çalışır. Ağ erişim katmanı ise üst katmandan gelen verilerin internet ortamına iletilirken gerekli olan fiziksel katman bilgilerini ve yerel alan ve geniş alan ağlarının bilgilerini düzenler. OSI protokolünde bu katman fiziksel ve veri bağı katmanlarına denk gelir.



Şekil 2.15. TCP-IP protokolü ile OSI protokolünün katmanları

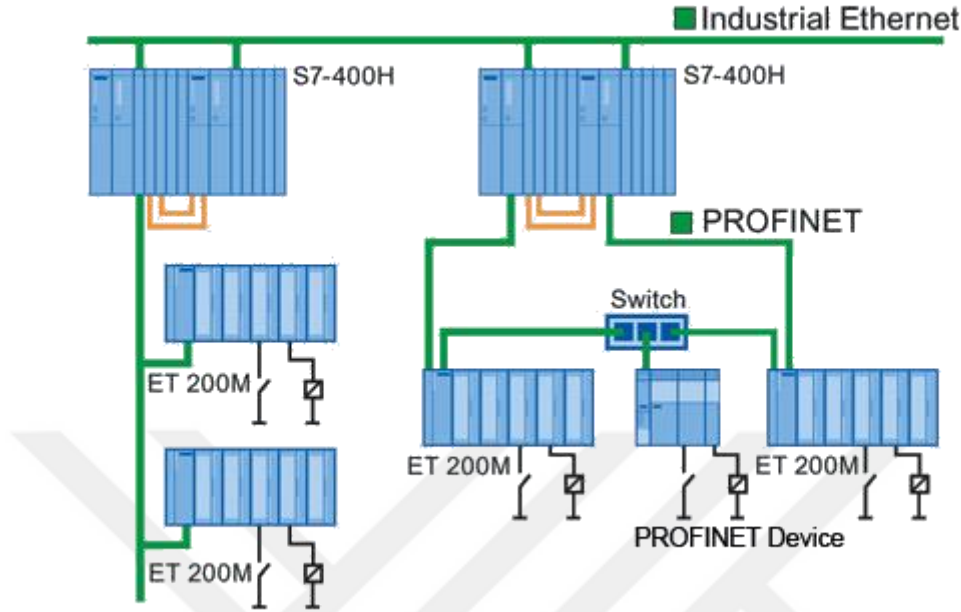
Şekil 2.15’ te TCP/IP ile OSI protokollerine ait katmanlar karşılaştırmalı olarak görülmektedir. TCP/IP protokolüne ait ilk iki katmana protokol katmanları denilmektedir, son iki katman ise ağ katmanları olarak adlandırılmaktadır. OSI referans modelinin ise ilk üç katmanı uygulama, son dört katmanı veri akış katmanları olarak bilinmektedir [20].

2.1.13. Profinet

Profinet, Ethernet donanım ve protokollerini endüstriyel otomasyonun gerçek zamanlı ihtiyaçlarına uygun olarak uyarlayan ethernet tabanlı bir endüstriyel ağ protokolüdür. Saha düzeyinde özellikle endüstriyel ethernet kullanımında ön plana çıkmaktadır. Bilindiği gibi endüstride alınan stratejik kararlar çok önemlidir ve işletme içerisinde bilgi akışının sorunsuz olması gerekir. Profinet, bu bilgi akışının verimli bir şekilde gerçekleşmesi için işletme içerisinde her kademedeki dikey ve yatay entegre haberleşmeyi sağlayan yenilikçi ve açık bir endüstriyel ethernet standardıdır. Bu standartla endüstriyel otomasyonun tüm gereksinimlerinin karşılanması amaçlanmıştır. Profinet, saha içerisinde dağıtılmış cihazların endüstriyel ethernet’e doğrudan bağlanmasına imkan sağlar ve eş zamanlı hareket kontrolü uygulamalarında da kullanılır. Bu durum Şekil 2.16’ da gösterilmiştir. Profinet’ in başka bir faydalı yönü ise standart ürünlerin birlikte çalışmasına olanak tanır ve bunu kablosuz teknoloji ile destekler. Profinet ve Profibus, mevcut cihazların değiştirilmesine ihtiyaç duyulmadan sisteme entegre edilebilir. Profinet standart ethernet üzerinde çalışır ve TCP/IP standardını kullanır.

Profibus ile kullanılmış, standart ve güvenli olması gereken verilerin tek veri kablosu üzerinden iletimi için kullanılan PROFI-safe güvenlik profili, aynı şekilde Profinet’ le

birlikte de kullanılabilir. Ayrıca endüstriyel kablosuz ağlar (WLAN) aracılığıyla güvenli haberleşme sağlanabilir [52].



Şekil 2.16. Profinet ile ilgili bir örnek

2.2. Otomasyon Sistemlerinin Bileşenleri

2.2.1. Programlanabilir Mantıksal Denetleyici (PLC)

Adını Programmable Logic Controller' den alan PLC, özel olarak hazırlanmış bir programa göre algılayıcılardan aldığı bilgileri işleyen ve elde ettiği sonuçları uç elemanlara aktaran bir mikrobilgisayar sistemidir. Röleli kontrol sistemlerinin gerçekleştirdiği fonksiyonları, kolaylaştırmak ve bazı olumsuz yönlerini gidermek adına bu fonksiyonların mikroişlemcili kontrol sistemleri ile yerine getirilebilmesidir. Zaman içerisinde bazı özellikleri geliştirilmiş ve ardışık kontrol, hareket denetimi, süreç denetimi (sıcaklık, hız, basınç, nem), veri yönetimi ve raporlama gibi amaçlarla endüstriyel alanda kullanılabilir hale getirilmiştir. Günümüzde birçok firma çeşitli özelliklerde PLC' ler üretmekte ve ürettikleri bu PLC' ler için uygun yazılımlar geliştirmektedir [8,11].

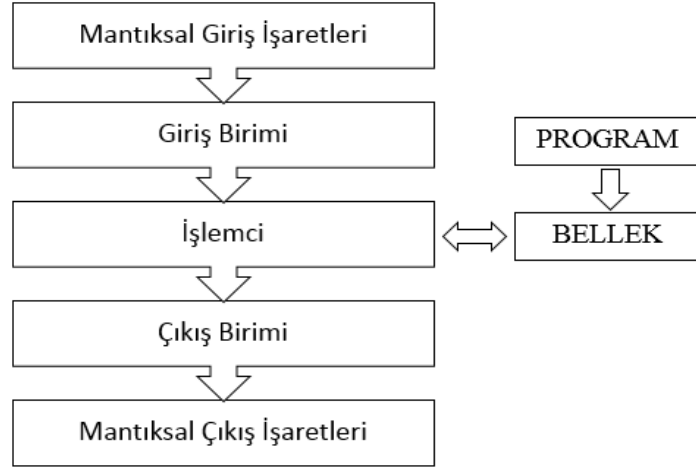


Şekil 2.17. Çeşitli PLC Görüntüleri

Şekil 2.17’ de örnek görüntüleri verilen bu cihazlar, zamanlama, sayma, sıralama gibi işlemleri ve her türlü lojik işlemleri yazılımla gerçekleştirdiklerinden karmaşık otomasyon sistemlerini çözmek daha kolay bir hal almıştır. PLC’ nin mevcut eski sistemlere göre avantajlarını sıralayacak olursak [8];

- Güvenilirdir ve uygulaması kolaydır.
- Eski sistemlere göre az yer kaplar ve az arıza yaparlar.
- Yeni uygulamalara çabuk adapte olurlar.
- Kötü çevre şartlarına karşı dayanıklıdırlar.
- Daha az kablo bağlantısı vardır.
- Hazır fonksiyonlar kullanılabilir.
- Giriş ve çıkış durumları izlenebilir.

Şekil 2.18’ de görüleceği üzere PLC, içerisinde işlemcinin (CPU) , giriş-çıkış ara yüzlerinin ve hafızanın (memory) bulunduğu bir endüstriyel mikrobilgisayardır. Kontrol işlemlerinde PLC’ ye çeşitli giriş ve kontrol içerikli sinyaller gönderilir. Bu sinyallerin işlenmesi sürecinden sonra çıkış sinyalleri üretilir ve çıkış elemanlarına iletilir. Giriş ve çıkış elemanları ve PLC arasındaki sinyal değişimleri için kullanılmak üzere giriş-çıkış arabirimi olarak adlandırılan tekrarlayıcılar gereklidir [10].



Şekil 2.18. PLC' nin genel yapısı

2.2.2. HMI

Human Machine Interface ifadesinin baş harflerinden oluşan HMI, insan makine ara yüzü anlamına gelir ve endüstriyel otomasyon sistemlerinin önemli bileşenlerinden birisidir. Kullanıcıların makine veya üretim tesisleri ile iletişim kurmasına olanak sağlayan ve bu iletişimi kurarken büyük miktarda karmaşık veriyi basite indirgeyerek kullanıcıya sunan aygıt ya da yazılımlardır. Endüstriyel sistemlerde, temelde cihazları görselleştiren ve uygulama kontrolünü sağlayan bir birimdir.

2.2.3. SCADA

2.2.3.1. Giriş ve SCADA' nın Tarihçesi

SCADA' nın geçmişi kontrol sistemlerinin varoluş günlerine dayanır. İlk SCADA sistemleri metre, ışıklar ve şeritli kaydediciler tarafından kullanılmıştır. Denetleyici kontrolü çeşitli kontrol düğmelerinin operatör tarafından manuel olarak işletilmesi amacıyla kullanılmıştır. Bu cihazlar geçmişte olduğu gibi hala işletmeler, fabrikalar ve güç üreten tesislerde denetleyici kontrolü ve veri toplama amacıyla kullanılmaktadır [12].

Endüstriyel otomasyon sistemlerinde kullanılan programlanabilir kontrolörler, dağıtılmış kontrol sistemleri, akıllı sensörler, döngü kontrolörleri ve giriş çıkış sistemleri gibi çeşitli cihazlardan gerçek zamanlı ve daimi veri toplayabilen, bu verileri belirtilen kısıtlara göre değerlendiren, topladığı verilere göre sonuçlar üreten ve gerekirse

kullanıcıya uyarı mesajları gönderen sistemlerdir. Denetleyici gözetim ve veri toplama sistemi olarak bilinen SCADA, endüstriyel sistemlerde sahadaki kontrol noktalarının uzaktan denetlenebilmesine olanak sağlamak için kullanılır.

SCADA, kullanılan sistemlerde veri toplama, toplanan verilerin analizini yapma, veri gönderme ve bu verilerin daha sonra bir kullanıcı ekranında gösterilmesi işlevlerini kapsar. SCADA sistemleri saha içerisindeki donanımlarını görüntüler ve denetler.

SCADA sistemleriyle kullanıcılar, merkezi bir kontrol noktasından geniş bir alandaki termik, hidrolik enerji üretim sistemleri, boru hatları, su şebekeleri ve benzeri iletim, dağıtım tesislerinde vanaları, elektrik makinelerini, ayırıcıları, motorları, valf anahtarları gibi birimleri uzaktan açıp kapayabilme, alarmları görüntüleme, ısı, nem, frekans gibi bilgileri toplama işlevlerini ekonomik ve emniyetli bir şekilde yerine getirebilirler.

SCADA sistemleri günümüzde birçok alanda yaygın olarak kullanılmaktadır. Bu kullanım alanlarına örnek verecek olursak [11];

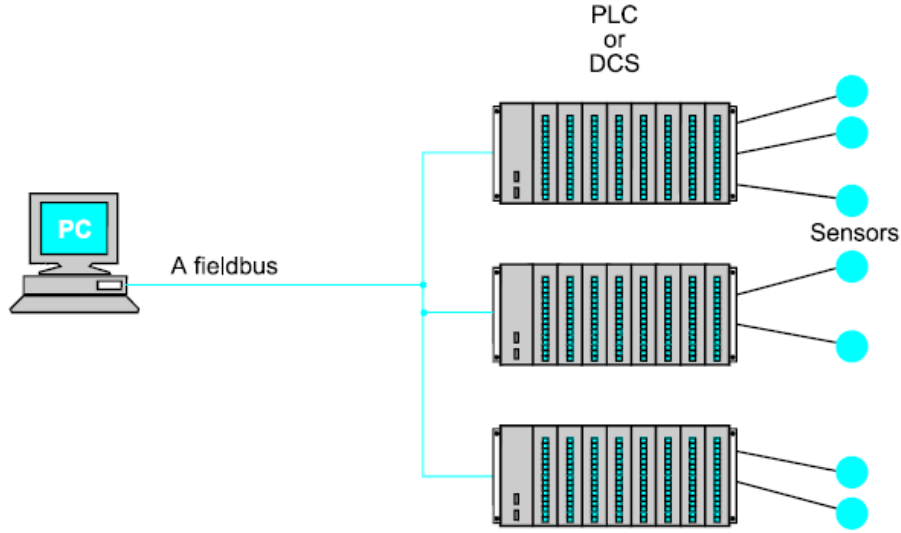
- Kimya endüstrisi
- Petrol ve doğalgaz boru hatları
- Petrokimya endüstrisi
- Elektrik üretim ve iletim sistemleri
- Elektrik dağıtım tesisleri
- Su Toplama, arıtma ve Dağıtım Tesisleri
- Hava kirliliği kontrolü
- Çimento endüstrisi
- Otomotiv endüstrisi
- Bina otomasyonu
- Proses tesisleri

Modern SCADA Sistemlerinin Temel İlkeleri

SCADA telemetri ve veri toplama birleşimine değinmektedir. SCADA, bilgi toplama, onu merkeze geri iletme, gerekli analizleri gerçekleştirme, kontrol ve belirli sayıda santral ekranında bilgi görüntülemeyi kapsar. Gerekli kontrol eylemleri daha sonra tekrar işleme iletilir [12].

PLC, endüstride hala çok yaygın olarak kullanılan kontrol sistemlerinden birisidir. İşletmelerdeki daha çok cihazı izlemek ve kontrol etmek artık büyük bir ihtiyaç haline

geldiğinden, sistemler daha akıllı ve küçük boyutlu hale gelmişlerdir. PLC' ler ve DCS Şekil 2.19' da gösterilmiştir.



Şekil 2.19. Bilgisayarın PLC veya DCS' ye bağlantısı

PLC / DCS SCADA sisteminin avantajları şunlardır:

- Bilgisayar, çok büyük miktardaki verileri saklayabilir.
- Veriler, kullanıcıların ihtiyaçlarına uygun şekilde görüntülenebilir.
- Geniş bir alandaki binlerce sensör sisteme bağlanabilir.
- Operatör sisteme gerçek veri simülasyonları dahil edebilir.
- RTU' lar tarafından birçok farklı türdeki veri toplanabilir.
- Veriler sadece belirli bir konumdan değil, her yerden görüntülenebilir.

Dezavantajları şunlardır:

- Panel tipi sensöre göre sistem daha karmaşıktır.
- Sistem analistleri ve programcılarda olduğu gibi farklı işletim becerilerine ihtiyaç duyulur.
- Binlerce sensörle birlikte hala bir çok kabloyla uğraşmak gerekir.
- Operatör, sadece PLC' nin aktardığı kadar görebilir.

SCADA Donanımı

Bir SCADA sistemi, bir iletişim sistemi üzerinden alan verisi toplayan ve veriyi ana istasyona geri gönderen bir grup RTU' dan oluşur. Ana istasyonu elde edilen verileri görüntüler ve operatöre uzaktan kumanda görevlerini gerçekleştirmesi için izin verir [12].

Daha karmaşık bir SCADA sisteminde esasen beş seviye veya hiyerarşi vardır:

- Alan düzeyi enstrümantasyon ve kontrol cihazları
- Terminal ve RTU' ların sıralanması
- Haberleşme sistemi
- Ana istasyon (lar)
- Ticari veri işleme bölümü bilgisayar sistemi

SCADA Yazılımı

SCADA yazılımı tescilli veya açık olarak iki tipe ayrılabilir. Şirketler donanımlarıyla haberleşmesi için tescilli yazılım geliştirirler. Bu sistemler anahtar teslim çözümler olarak satılmaktadır. Sisteme kazandırdıkları iş yeteneği sayesinde açık yazılım sistemleri popülerlik kazanmıştır [12].

Citect ve Wonderware, SCADA sistemleri için piyasada bulunan açık yazılım paketlerine örnek olarak verilebilir.

SCADA ve Yerel Alan Ağları

Bilgi paylaşımı amacıyla SCADA ağındaki tüm düğümleri etkinleştirmek için, bazı iletim ortamları ile bağlanmış olmaları gerekir. Bağlantı metodu ağ topolojisi olarak da bilinir.

Düğüm, bu iletim ortamlarını paylaşma ihtiyacı duyarlar. Belirli bir göndereni engellemeden tüm düğümlere ortama erişim izni verilmelidir.

Ucuz ve kullanımı kolay olduğu için ethernet, bugün en yaygın kullanılan yerel alan ağıdır. SCADA ağının yerel alan ağına bağlantısı, doğru yazılım ve izinlerle, sisteme erişmek için işletmedeki herkesi tanır. Veriler bir veri tabanında tutulduğundan beri kullanıcıların bilgileri okuması sınırlı olabilir [12].

SCADA Sistemlerinde Modem Kullanımı

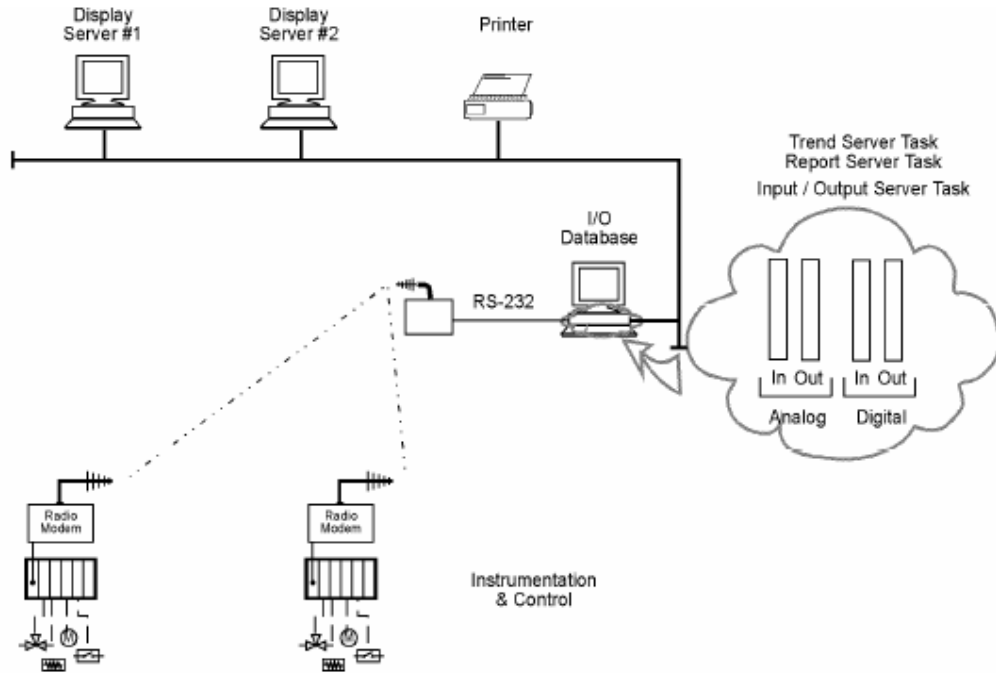
Genellikle SCADA sistemlerinde RTU uzak bir konumda yer almaktadır. Bu mesafe onlarca metreden binlerce kilometreye kadar değişebilir. Uzun mesafelerde RTU ile haberleşmek için var olan en ekonomik yollardan biri çevirmeli telefon bağlantısı ile olabilir [12]. Bu cihazlar için gereken cihazlar; Şekil 2.20’ deki gibi bir bilgisayar, iki modem ve RTU’ dur.



Şekil 2.20. Modem kullanılarak bilgisayardan RTU’ ya bağlantı

SCADA Sistemleri Yazılımı

Bir SCADA sisteminin tipik bileşenleri, SCADA yazılımı üzerinde durularak Şekil 2.21’ de belirtilmiştir.



Şekil 2.21. Bir SCADA sisteminin bileşenleri

SCADA yazılımının beklenen tipik temel özellikleri aşağıda listelenmiştir. Bu özellikler, uygulanacak donanımına bağlıdır.

SCADA Temel Özellikleri

SCADA sisteminin temel özellikleri maddeler halinde sıralanacak olursa [12];

Kullanıcı arayüzü

- Klavye
- Fare
- Trackball
- Dokunmatik ekran

Grafik görüntüler

- Müşteri tarafından yapılandırılabilir, nesne odaklı ve bit eşlenmiş
- Sınırsız sayıda sayfa
- Çözünürlük; kullanıcı ekran çözünürlüğü ile aynıdır.

Alarmlar

- İstemci sunucu mimarisi
- 1 milisaniye hassasiyetli zaman ayarlı alarmlar
- Tek bir ağ alındısı ve alarm kontrolü
- Tüm istemcilerle paylaşılan alarmlar
- Kronolojik sırayla görüntülenen alarmlar
- Alarm sayfalarının dinamik paylaşımı
- Kullanıcı tanımlı biçim ve renkler
- Her bir analog alarm için dört adede kadar ayarlanabilir noktalar
- Analog alarmlar için sapma ve değişim oranları görüntüleme
- Kategoriye göre seçilebilir alarm ekranları
- Tarihsel alarm ve olay günlüğü
- Bağlam-duyarlı yardım
- Çevrimiçi olarak alarmı devre dışı bırakma ve eşik modifikasyonu
- Olay tetiklemeli alarmlar

- Alarm tetiklemeli raporlar
- Alarlara eklenebilir operatör yorumları

Trendler

- İstemci sunucu mimarisi
- Gerçek trend çıktıları
- Lastik bant trend yakınlaştırma
- DBF, CSV dosyalarına veri çıktıları
- X / Y düzlem yeteneği
- Olay tabanlı trendler
- Açılır trend gösterge
- Trend kılavuz çizgiler veya profiller
- Arka plan trend grafikler
- Gerçek zamanlı çoklu kalem trendi
- Kısa ve uzun vadeli trend gösterge
- Veri depolama uzunluğu ve izleme sıklığı
- Tarihsel trend verilerin arşivlenmesi
- Arşivlenen tarihsel trend verilerin On-line alımı
- Görüntülenebilir tam değer ve zaman
- Grafiksel olarak temsil edilebilen gerçek zamanlı trend veri

RTU (ve PLC) Arayüzü

- Tüm uyumlu protokoller standart olarak dahil edilmiştir.
- DDE sürücülerini destekler.
- Ara yüz, RTU' lar, döngü denetleyicileri, barkod okuyucular ve diğer ekipmanlar için uygundur.
- Sürücü araç kiti mevcuttur.
- Geleneksel önceden tanımlı tarama yöntemi yerine ihtiyaç temelli çalışır.
- PLC' ye blok veri isteklerinin optimizasyonu
- Ağ kullanıcı veri isteklerinin rasyonalizasyonu
- PLC veri yolu bant genişliği maksimizasyonu

Ölçeklenebilirlik

Mevcut donanım yer değiştirilmeden ilave donanım eklenebilir. Bu, sadece PLC mimarisi ile sınırlıdır.

Veri Erişimi

- Herhangi bir ağ kullanıcısı tarafından verilere doğrudan, gerçek zamanlı erişim
- Gerçek zamanlı verilere üçüncü taraf erişimi, Ör: Lotus 123 ve EXCEL
- DDE ağı
- DDE uyumluluğu: okuma, yazma ve çalıştırma
- Tüm giriş çıkış cihaz noktalarına DDE
- Pano

Veri tabanı

- ODBC sürücü desteği
- Doğrudan SQL komutları veya üst düzey raporlama

Ağ

- NetWare, LAN Manager, çalışma grupları için Windows, Windows NT gibi tüm NETBIOS uyumlu ağları destekler.
- NetBEUI, IPX/SPX, TCP/IP ve daha fazla protokolü destekler.
- Merkezi alarm, trend ve rapor işleme – verilere ağdaki her yerden erişim
- Bir ağ yapılandırmasına gerek olmaması
- Tek onay kutusu üzerinden etkin olabilir, yapılandırmaya gerek yok
- Ağda oturum açmış kullanıcı sayısı kadar LAN izni, ağdaki düğümlerin sayısı kadar değil
- Dosya sunucusuna ihtiyaç yok
- Çok kullanıcıli sistem, operatörler arasında tam bir iletişim
- Yüksek performanslı RAS ve WAN desteği
- PSTN dial up desteği

İstemci / Sunucu Dağıtılmış İşlemler

- Açık mimari tasarım
- Gerçek zamanlı çoklu görev
- Kullanıcı yapılandırması olmadan İstemci / Sunucu desteği

- Dağıtılmış proje güncellemeleri (ağ üzerinden yansıyan değişiklikler)
- Çoklu ekran düğümlerinin eşzamanlı desteği
- Herhangi bir düğümden herhangi bir veriye (trend, alarm, rapor) erişim

SCADA Yazılım Paketi

SCADA paketi, gelecekteki ihtiyaçlarla başa çıkabilmek için kolayca güncellenebilmelidir. Değişen ve genişleyen ihtiyaçlara cevap verebilmek için sistem kolayca düzenlenebilir olmalıdır.

SCADA sistemini tasarlarken izlenmesi gereken iki ana yaklaşım bulunmaktadır:

- Merkezi, tek bir ana bilgisayar tüm sistemi izler ve tüm veri bu bilgisayarda bulunan tek bir veri tabanında depolanır.
- Dağınık, SCADA sistemi birkaç küçük bilgisayarla paylaşılır. (Genellikle kişisel bilgisayarlar)

Her görev için gerekli veri türünü incelemek ve ardından sistemi uygun şekilde yapılandırmak etkili bir çözüm olacaktır. Bir istemci sunucu yaklaşımı ile daha etkili bir sistem elde edilir.

SCADA sistemlerde genellikle beş görev vardır. Bu görevlerden her biri kendilerine ait işlemleri gerçekleştirir [12].

- Giriş / Çıkış görevi. Bu program sistemin izlenmesi ve kontrol edilmesini sağlayan ara yüzüdür.
- Alarm görevi. Bu, dijital alarm noktalarını tespit ederek ve analog alarm noktalarının değerlerini alarm eşik değerleri ile karşılaştırarak tüm alarmları yönetir.
- Trend görevi. Bu görev zamanla izlenecek verileri toplar.
- Rapor görevi. Raporlar sistem verilerinden üretilir. Bu raporlar periyodik olabilir, bir olayla tetiklenebilir veya operatör tarafından aktive olabilir.
- Görüntü görevi. Bu görev, operatör tarafından izlenecek tüm veriyi ve ihtiyaç duyulan tüm kontrol eylemlerini yönetir.

SCADA Sistemlerde Yeni Teknolojiler

İletişim teknolojisindeki hızlı gelişmeler, yeni SCADA sisteminde önemli bir itici güçtür. SCADA teknolojisi ile ilgili ortaya çıkan yeni teknolojilerden birkaçı kısaca aşağıda listelenmiştir [12].

- Ana istasyonlar için LAN teknolojilerindeki hızlı gelişim
- İnsan makine arayüzü
- Uzak terminal birimleri
- Haberleşme

2.3. ARM Tabanlı İşlemciler

2.3.1. ARM Mimarisi

ARM (Advanced RISC Machine - eskiden Acorn RISC Machine) mimarisi, MIPS mimarisi gibi çip üreticilerine fikri mülkiyet olarak satılır. ARM Ltd. sadece CPU çekirdeği tasarlar ve Samsung, Atmel gibi birçok üreticiye bunları lisanslar.

ARM, hemen hemen tüm uygulama pazarları için performans, güç ve maliyet gereksinimlerini karşılayacak en yaygın mikroişlemci çekirdeklerini sunan, mikroişlemci teknolojisi sektöründe önde gelen tedarikçi konumundadır [21].

MIPS mimarisinden farklı olarak, tüm ARM işlemcileri aynı komut kümesine sahiptir ve böylece bütün işlemciler tamamen yazılım (assembly ve binary kod) uyumludur. Hemen hemen tüm güncel ARM işlemciler ARMv5TE ve ARMV4 mimarilerine dayanmaktadır. Onların da sadece saat hızı, önbellek ve ek özellikleri farklıdır [51].

Enerji tasarrufu özellikleri nedeniyle ARM işlemciler, düşük güç tüketiminin kritik tasarım hedefi olduğu mobil elektronik marketlere hakimdirler. Rakiplerini zorlayacak performanslara erişebilen, onlara nispeten daha az enerji tükettiğinden ve zengin çevresel donanım desteğinden dolayı cep telefonları, tablet bilgisayarlar, geliştirme kartları ve daha birçok elektronik cihazda ARM mimarisi ile üretilmiş işlemciler kullanılmaktadır.

ARM mimarisi beş kategoride gruplandırılmıştır. Bunlar Klasik ARM İşlemciler, ARM Cortex uygulama işlemcileri, ARM Cortex gerçek zamanlı gömülü işlemciler, ARM Cortex gömülü işlemciler, ARM uzman işlemciler [21].

Klasik ARM işlemciler, ARM11, ARM9 ve ARM7 işlemci ailelerini içerir. Dünya çapında yaygın olarak lisanslı olan bu işlemciler, günümüzde kullanılan birçok uygulama için uygun maliyetli çözümler sunuyor. Piyasada 20 milyardan fazla cihazda kullanılan bu işlemciler, geliştiricilere işlemcileri uygulamalarının titiz doğasıyla tanışırken güvence verir ve bu ekosistemi 20 yıldan daha fazla zamandır genişletmektedir.

ARM Cortex uygulama işlemcilerinden Cortex-A serisi, zengin işletim Sistemleri için yüksek performanslı işlemcilerdir. Cortex-A işlemcilerinin tamamı, en son bilgisayar teknolojileri için olağanüstü 32 bit performans sunar. Cortex A-72, Cortex-A57 ve Cortex A53 işlemcileri, yeni nesil mobil, ağ ve sunucu teknolojilerinde 32 bit ve 64 bit performans sunar.

Uygulamalar şunlardır:

- Akıllı Telefonlar
- Netbook'lar
- E-okuyucular
- Dijital TV
- Ev Ağ Geçitleri
- Sunucular ve Ağ

ARM Cortex gerçek zamanlı gömülü işlemcilerden Cortex-R serisi, gerçek zamanlı uygulamalar için olağanüstü performans sunar. Cortex gerçek zamanlı gömülü işlemciler, düşük güç ihtiyacı olan, kesme davranışlarının olağanüstü performansla dengelendiği ve mevcut platformlarda güçlü uyumluluk sağlayan derin gömülü gerçek zamanlı uygulamalar için geliştirilmiştir.

Uygulamalar şunlardır:

- Otomotiv fren sistemleri
- Güç aktarma sistemleri
- Yığın depolama denetleyicisi
- Ağ ve Baskı

ARM Cortex gömülü işlemcilerden Cortex-M serisi, deterministik mikro denetleyici uygulamaları için maliyet duyarlı çözümler sunar. Cortex-M serisi işlemciler, öncelikle, hızlı, son derece deterministik kesme yönetimi, son derece düşük kapı sayısı ve mümkün olan en düşük güç tüketimi arzusu ile birleşen ihtiyaçların olduğu mikro denetleyici alanı için geliştirilmiştir.

Uygulamalar şunlardır:

- Mikrodenetleyiciler
- Karışık sinyal cihazları
- Akıllı sensörler
- Otomotiv gövde elektroniği ve hava yastıkları

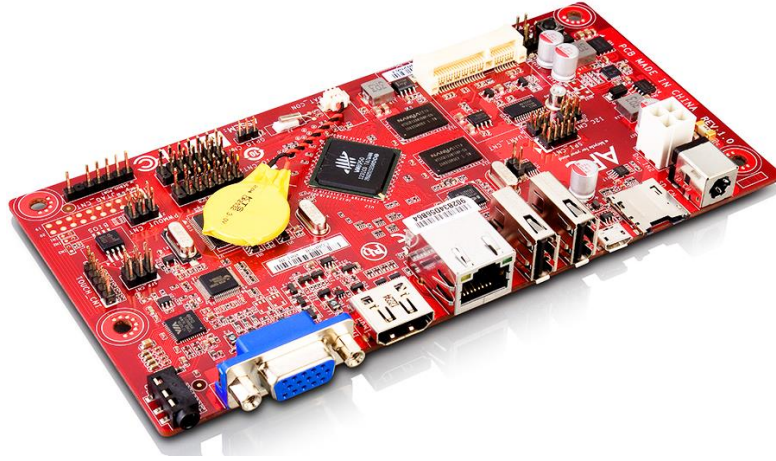
ARM uzman işlemcilerden SecurCore, yüksek güvenlik uygulamaları için geliştirilmiştir. ARM Uzman İşlemciler belirli pazarlarda zorlu ihtiyaçları karşılamak üzere tasarlanmıştır. SecurCore işlemciler, güvenlik piyasalarında mobil SIM' ler ve kimlik uygulamaları için kullanılır ve üstün performanslarının yanında bu işlemcilere güvenlik saldırılarını algılayıp önlemeleri için çok sayıda teknoloji entegre edilmiştir [21].

2.3.2. ARM İşlemcili Geliştirme Kartları

Piyasada ARM mimarisi üzerine kurulu işlemcilerle donatılmış birçok geliştirme kartı bulunmaktadır. Bunlardan bazılarını aşağıda değinilmiştir.

Şekil 2.22' de gösterilen APC Rock, daha hızlı işlemci ve daha fazla kapasitede belleğiyle orijinal APC 8750' nin yeniden tasarımıdır. Rock, prototip projelerini oluşturmaları için geliştiriciler için mükemmel bir platformdur. Özelliklerine bakacak olursak [44]:

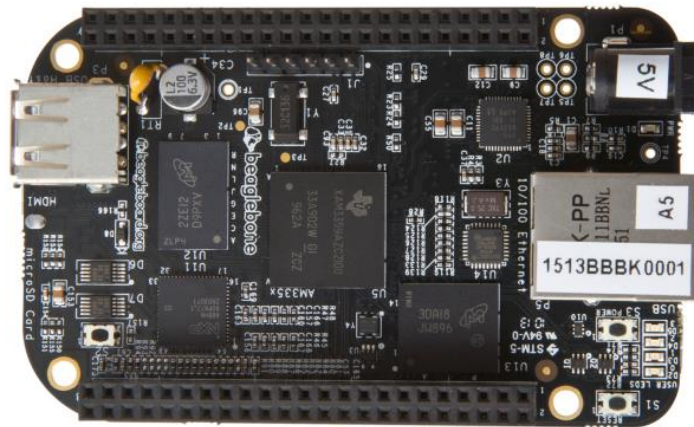
- Firefox OS işletim sistemi
- VIA ARM Cortex-A9 @800Mhz işlemci
- DDR3 512MB hafıza
- 4GB NAND Flash depolama
- 2D/3D Graphic 720p' ye kadar çözünürlük
- HDMI, VGA, USB 2.0 (x2), microUSB (OTG), Ses çıkışı / Mik girişi, microSD Slot
- 10/100 Ethernet
- 20-pin ARM-JTAG header
- Ekstra GPIO, SPI and I2C veri yolları



Şekil 2.22. APC Rock genel görünüm

Şekil 2.23’ te görüntüsü verilen BeagleBone Black, geliştiricileri ve meraklıları için düşük maliyetli, toplum destekli geliştirme platformudur. Linux, 10 saniyenin altında başlatılabilir ve sadece tek bir USB kablosu ile 5 dakikadan daha kısa sürede geliştirmeye başlanabilir. Özellikleri [45]:

- Debian, Android, Ubuntu işletim sistemleri
- AM335x 1GHz ARM® Cortex-A8 işlemci
- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash depolama
- 3D grafik hızlandırma
- 2x PRU 32-bit mikrodenetleyici
- USB, Ethernet, HDMI
- 2x 46 pin headers



Şekil 2.23. BeagleBone Black genel görünüm

Cubieboard2, tek çekirdekten çift çekirdeğe yükseltilebilir ve iki kat gpu performansına sahip ARM Cortex A7 Dual Core işlemcisini kullanan bir geliştirme platformudur ve Şekil 2.24' te bir örneği gösterilmiştir. Özellikleri [46]:

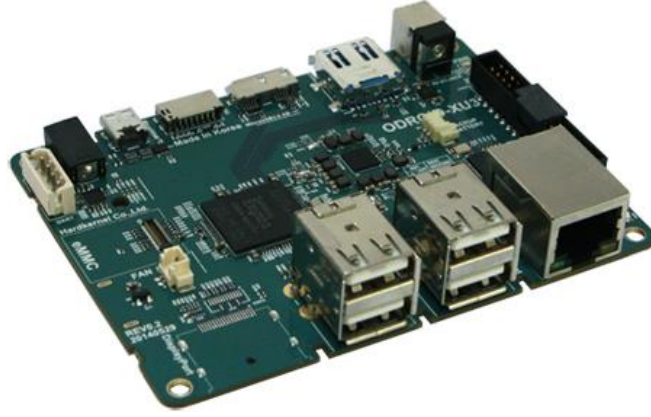
- Android 4.2 Jelly Bean, Ubuntu 12.04 ve diğer Linux dağıtımları
- ARM® Cortex™-A7 Dual-Core işlemci
- ARM® Mali400MP2, Complies with OpenGL ES 2.0/1.1 GPU
- 1GB DDR3 @960M hafıza
- 4GB internal NAND flash, up to 64GB on uSD slot, up to 2T on 2.5 SATA disk depolama
- 5VDC giriş 2A veya USB otg giriş
- 10/100 ethernet, opsiyonel wifi
- 2xUSB 2.0



Şekil 2.24. Cubieboard2 genel görünüm

Şekil 2.25' te bir örneği gösterilen ODROID-XU, Enerji bakımından daha tasarruflu, donanım bakımından daha güçlü ve daha küçük boyutlarıyla yeni nesil bir cihazdır. Açık kaynak desteği sunar ve Ubuntu, Android 4.2 ve çeşitli linux dağıtımlarını destekler. Özellikleri [47]:

- Exynos5 Octa Cortex™-A15 1.6Ghz quad core and Cortex™-A7 quad core işlemciler
- PowerVR SGX544MP3 GPU
- 2Gb LPDDR3 RAM
- USB 3.0 Host x 1, USB 3.0 OTG x 1, USB 2.0 Host x 4
- HDMI
- eMMC 4.5 Flash Storage



Şekil 2.25. ODROID-XU genel görünüm

Şekil 2.26’ da örnek bir görüntüsü verilen Wandboard, Cortex-A9 işlemci tabanlı, yüksek performanslı multimedya özelliklerine sahip, ultra düşük güç ile çalışan bir bilgisayardır. Özellikleri [48]:

- Freescale i.MX6 Quad işlemci, Cortex-A9 Quad çekirdek
- Vivante GC 2000 GPU
- 2GB DDR3
- Ses, HDMI, Micro-SD slot, Serial port, USB, SATA, Gigabit LAN, WiFi (802.11n), Bluetooth



Şekil 2.26. Wandboard genel görünüm

Şekil 2.27’ de örnek bir görüntüsü verilen PandaBoard ES, daha yüksek saat hızlarında çalışan CPU ve GPU ile OMAP 4460 SoC dayalı yeni bir geliştirme kartıdır. Özellikleri [49]:

- 1.2 GHZ Dual-core ARM® Cortex-A9 MPCore işlemci
- Full HD (1080p) multi-standard video encode/decode
- HDMI, DVI-D
- 1 GB DDR2 RAM

- 3.5" Ses giriş/çıkış
- Onboard 10/100 Ethernet, 802.11 b/g/n wifi, Bluetooth
- 2xUSB 2.0



Şekil 2.27. Pandaboard genel görünüm

Özellikleri hakkında bilgi verilen ve piyasada bulunan popüler olarak kullanılan Linux tabanlı geliştirme kartları Tablo 2.1’ de gösterilmiştir.

Tablo 2.1. Linux tabanlı geliştirme kartları

APC Rock -- \$79, Via Technologies, 800MHz Cortex-A9 Wondermedia
BeagleBone Black -- \$45, BeagleBoard.org, 1GHz Cortex-A8 TI Sitara AM3359
Cosmic+ -- \$65, Phytex, 500MHz Cortex-A5 Freescale Vybrid
Cubieboard 2 -- \$59, Wang And Tom Development Ltd. (Allwinner), Cortex-A7 dual-core Allwinner A20
MicroZed -- \$199, Avnet, Xilinx Cortex-A9/FPGA Zynq-7010
MinnowBoard -- \$199, MinnowBoard.org (Intel), 1GHz Intel Atom E640
Odroid-XU -- \$149/\$169, Odroid Project (Hardkernel), Samsung Exynos 5410 Octa (8-core Big, Little Cortex-A15 and –A7)
PandaBoard -- \$182, PandaBoard.org (TI), TI 1.2GHz Cortex-A9 OMAP4460
Raspberry Pi Model B – \$35, Raspberry Pi Foundation, Broadcom 700MHz ARM11 BCM2835
Wandboard Quad -- Wandboard.org (Freescale), Freescale 1GHz Cortex-A9 i.MX 6Quad

2.3.3. G m l  Sistemler

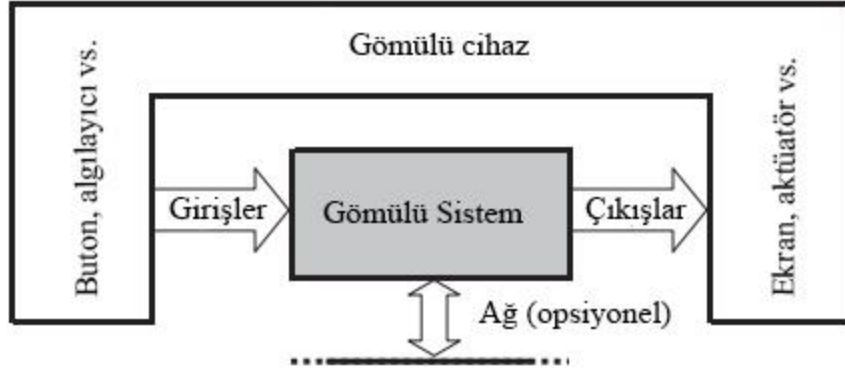
Bir g m l  sistem daha b y k bir cihazda bulunan bilgisayar donanım ve yazılımının birleřimidir. Cihaz tarafından tamamen kapsandığı i in bir ok durumda cihazda bir g m l  sistem olduėunun farkına varılmaz.  oėunlukla cihazlar bu sistemler tarafından kontrol edilir. Diėer g revler birden fazla donanım par ası arasındaki iletiřimi saėlar veya makineler i in insan-makine ara y z n  şekillendirir. G m l  sistemler, bir masa st  bilgisayarın aksine  nceden tanımlanmış sadece bir veya birden fazla g revi ger ekleřtirir. Genel ama lı bilgisayar (masa st )  reticisi, m řterinin onunla ne yapacaėını bilmez. Bir sunucu veya iř istasyonu olarak veya sadece oyun oynamak i in kullanılabilir. Bunun aksine, g m l  bir sistemin hem donanımı hem de yazılımı genellikle sadece belirli bir uygulama i in tasarlanmıştır. řaşırtıcı bir şekilde bir masa st  sistemi birkaç g m l  sistemden oluřmaktadır. Hemen hemen her bileřende (klavye, fare, s r c ler ve hatta monit rde bile) k  k bir bilgisayar  alıřır.

Kontrol ama lı kullanılan g m l  sistemlerin genellikle sıkı zamanlama kısıtlamaları vardır.  nemli bir hesaplamanın belli bir s re i erisinde tamamlanmaması halinde, kontrol edilen cihaz (makine) veya  zerinde  alıřan bir par ası hasar g rebilir ve hatta insanlar bile zarar g rebilir.  rneėin bir u ak u uř kontrol sistemine ait ger ek zamanlı sistemin, zaman sınırlı bir g revi ka ırması felaketle sonu lanabilir.

Yani ger ek zamanlı sistemler, rastgele asenkron olaylar ( rneėin  evresel sens rlerin  ng r lemeyen deėerleri) i inde zamanlama kısıtlamalarına uygun olmalıdır. Sonu ları daha ciddi olan bu kısıtlamalara baėlı cihaz daha sert ger ek zamanlı sistemlere ihtiya  duyar. Tam tersi olarak, zayıf zamanlama kısıtlamaları yumuřak ger ek zamanlı sistemlere izin verir.

Tanım olarak, g m l  sistemler bir iřlemci ve yazılım i erir. Ancak zorunlu olarak daha  ok bileřen vardır. Yazılım bir yerde (ROM veya flash  ip gibi) kalıcı olarak kaydedilmek zorundadır ve  alıřma zamanı verilerini depolamak i in RAM kullanılmalıdır. Sistemin depolama boyutu gereksinimlerine baėlı olarak, bellek  ipte (mikro denetleyici) veya ek devrelerde veya harici cihazlarda (hard-disk) olabilir. K  k cihazların RAM ve ROM' u 1 KB' tan daha az olabilir iken daha kapsamlı sistemlerde bu deėer daha fazladır.

Belleğin yanı sıra bir gömülü sistem, sensör verilerini almak, kullanıcı ile etkileşim kurmak veya aktüatörleri tetiklemek için Şekil 2.28’ de görüldüğü gibi giriş ve çıkışlara ihtiyaç duyar [50].



Şekil 2.28. Bir gömülü sistemin yapısı

Bu ortak özelliklere ek olarak, gömülü donanımın kalanı genellikle cihazın belirli gereksinimlerini karşılamak için benzersizdir. Bazı gömülü sistemler, diğer gömülü sistemlerle veya bir kontrol merkeziyle iletişim kurmak zorundadır. Bu genelde bazı ağ türleri kullanılarak yapılır.

3. WEB TEKNOLOJİLERİ

3.1. HTML

HTML (Hypertext Markup Language), etiket veya tag denilen kod parçalarından oluşur ve web sayfalarındaki metin, resim gibi öğelerin sayfa içerisinde nasıl yerleşeceklerini ve sayfalar arası bağlantıların nasıl olacağını belirler. Web sisteminde arka planda çalışır, çoklu ortamı destekler ve ek yazılımlarla istemci/sunucu gibi etkileşimli uygulamaları oluşturmak için de kullanılır.

Web, metin, ses, görüntü ve video gibi öğelerden, kısaca yaşayan sayfalardan oluşan bir yapıdır. Bu sayfalar web alanlarını oluşturur. Her web alanı da ana sayfa ve bu ana sayfaya bağlantılı diğer sayfalardan oluşur. Bu bağlantılar web alanındaki bir iç bağlantı olabileceği gibi dış kaynaklı bir bağlantı da olabilir.

Aslında HTML tam olarak bir programlama dili değildir. Çünkü bir seri prosedürden oluşan programlama dilleri genelde dış bir veriye ulaşmayı amaçlar. HTML belgeler ise verinin kendisini barındırır. HTML veriler arasına yerleştirilen etiketler aracılığıyla metin, resim gibi öğelerin, yani tüm sayfanın, tarayıcı programları tarafından nasıl görüntüleneceğini belirler.

Yayıncılıkta yazarın hazırlamış olduğu içerik editörler tarafından düzenlenerek basıma gönderilir ve en son adımda belge basılır. Bu işlem Web’ de HTML aracılığıyla gerçekleşir ve içerikler bir web sunucu üzerinden yayınlanır. Başka bir fark ise yayıncılıkta içerikler okuyuculara dağıtılır; ama HTML ile hazırlanan içerikler kullanıcılar tarafından okunmak üzere ziyaret edilir [22].

Bütün HTML etiketleri "<" ve ">" simgeleri arasına yazılır. Çok az bir kısmı tek olarak kullanılır,
 gibi, genelde ise açılıp kapatılarak kullanılırlar, gibi. Tarayıcılar HTML taglarını ve scriptleri görüntüleyemezler; ama onları sayfanın içeriğini yorumlamak için kullanırlar [40].

HTML, web sayfalarının davranışını etkileyen javascript gibi dillerde yazılmış scriptleri barındırabilir. Web tarayıcıları ayrıca görüntüyü, yazı ve diğer materyallerin yerleşimini tanımlamak için basamaklı stil şablonlarına (Cascading Style Sheets - CSS) başvurabilir.

HTML’ nin geçmişine bakılırsa; 1980 yılında CERN bünyesinde bir fizikçi olan Tim Berners-Lee, ENQUIRE isimli bir sistemi araştırmacıların kullanması ve doküman

paylaşımı için önermiş ve prototip olarak üretmiştir. Berners-Lee, HTML' yi SGML (Standart Genelleştirilmiş İşaretleme Dili)' nin bir uygulaması olarak düşünmüştür. SGML, 1960' ların sonlarında IBM tarafından belge taşıma sorununa çözüm olarak geliştirilmiştir. Öncelikle GML (General Markup Language) ismini almış ve daha sonra ISO tarafından standart hale getirilmiştir.

1989 yılında, Berners-Lee, Internet tabanlı bir zengin metin sistemi öneren bir not yazdı. 1990 yılında Berners-Lee HTML' yi duyurdu ve tarayıcı ve sunucu yazılımını yazdı. O yıl, Berners-Lee ve CERN veri sistemleri mühendisi Robert Cailliau finansman için ortak bir istek üzerine işbirliği sağladı; ancak proje CERN tarafından resmen kabul edilmedi. 1990 yılındaki kendi kişisel notlarında zengin metnin kullanıldığı bazı alanları listeledi ve öncelikle bir ansiklopediye koydu.

HTML' nin halka açık ilk açıklaması 1991 yılında Tim Berners-Lee tarafından internet üzerinde belirtilen "HTML Etiketleri" adlı bir belge oldu. Burada HTML' nin ilk basit tasarımını içeren 18 öge açıklanmıştır. Bu öğelerden 11' i HTML 4' te halen mevcuttur.

HTML web tarayıcılarının, görsel ya da işitsel web sayfalarına metin, resim ve diğer malzemeleri yorumlaması ve oluşturması için kullanılan bir işaretleme dilidir. HTML işaretlemenin her ögesi için varsayılan özellikler tarayıcıda tanımlanır ve bu özellikler CSS kullanan bir web sayfası tasarımcısı tarafından değiştirilebilir veya geliştirilebilir. Metin öğelerinin birçoğu 1988 ISO teknik raporunda bulunur [40].

HTML, etiketler ve özellikleri, karakter tabanlı veri türleri, karakter ve varlık referansları gibi bir çok önemli bileşenden oluşur. HTML etiketleri çoğunlukla <h1> ve </h1> şeklinde ikili şekilde kullanılır ve bazen de gibi tek olarak da kullanılabilirler. Bu şekilde çift olarak kullanılan ilk etiket başlama etiketi ve ikincisi de bitiş etiketidir. Aşağıdaki örnekte basit bir HTML kod bloğu görülmektedir:

```
<!DOCTYPE html>
<html>
<head>
<title>Bu bir başlık</title>
</head>
<body>
<p>Merhaba Dünya!</p>
</body>
</html>
```

3.2. JavaScript

JavaScript, web sayfalarında yaygın olarak kullanılan dinamik bir programlama dilidir. JavaScript dili ile yazılan ve istemci tarafında çalışan kod bloklarıyla tarayıcı kullanıcı etkileşimi, tarayıcının istenilen şekilde kontrol edilmesi ve sunucu ile asenkron olarak iletişim kurulup web sayfalarının içeriğinin değiştirilmesi gibi işlevler sağlanır. JavaScript, Node.js gibi platformlar aracılığıyla sunucu tarafında da kullanılabilir [23].

JavaScript tartışmasız dünyanın en önemli programlama dilidir. Önce bir oyuncak gibi düşünülse de JavaScript, şu anda tarihin en yaygın programlama dilidir. Neredeyse bir bilgisayar veya bir akıllı telefonu olan herkes JavaScript programları çalıştırmak ve kendilerine ait programlar oluşturmak için gereken tüm araçlara sahiptir. Tek ihtiyacınız olan bir tarayıcı ve bir metin editörüdür.

JavaScript, HTML, CSS yaygın hale gelmiştir. Öyle ki açık web standartları sunum katmanı olarak Windows, Firefox OS, Gnome ve Google'ın Chrome OS dahil olmak üzere yerli uygulamalar için bir çok işletim sistemi tarafından benimsenmiştir. Ayrıca iPhone ve Android mobil cihazlar, yerli uygulamalarda HTML5 ve JavaScript işlevselliğine izin veren web görünümelerini destekliyor.

JavaScript ayrıca donanım dünyasına taşınıyor. Arduino, tessel, Espruino ve NodeBots gibi projeler yakın gelecekte JavaScript' in gömülü sistemler ve robotik için ortak bir dil olabileceğinin sinyalini veriyor.

JavaScript programı oluşturma, bir metin dosyasını düzenleyerek ve tarayıcıda açmak kadar basittir. İndirmek ve yüklemek için hiçbir kompleks geliştirme ortamı yoktur ve öğrenmek için hiçbir karmaşık IDE bulunmamaktadır. JavaScript, öğrenmesi de kolay bir dildir. Temel sözdizimi, C ailesi sözdizimiyle uğraşmış herhangi bir programcı için hemen tanıdık gelecektir.

Uzun bir süre, JavaScript ile verileri kaydetmek için hiçbir yol yoktu. Eğer verilerin devamını istiyorsanız, bir web sunucusuna bir form göndermek ve sayfa yenilemesi için beklemek zorundaydınız. Bu duyarlı ve dinamik web uygulamaları oluşturma işlemini engelliyordu. Ancak, 2000 yılında Microsoft, Ajax teknolojisini Internet Explorer' a taşımaya başladı. Kısa bir süre sonra, diğer tarayıcılar da XMLHttpRequest nesnesi için destek eklediler. 2004 yılında, Google Gmail' i başlattı. Hemen beğenildi; çünkü kullanıcılara e-postaları için neredeyse sınırsız depolama alanı vaat ediyordu. Gmail ayrıca büyük bir yenilik de getirmiş oldu. Artık sayfa yenilemelerine ihtiyaç duymuyordu.

Aslında Gmail, tek bir sayfada hızlı ve duyarlı web uygulamaları oluştururken yeni Ajax teknolojisinin avantajlarından yararlanıyordu. O zamandan bu yana, web geliştiricileri tam gelişmiş, bulut tabanlı ofis paketleri, Facebook'un JavaScript SDK' sı gibi sosyal API'leri ve hatta yoğun grafikli video oyunları da dahil olmak üzere uygulamaların neredeyse her türünü üretti. Tüm bunlar Atwood Yasası' nı kanıtlar niteliktedir: "JavaScript ile yazılabilecek herhangi bir uygulama, sonunda JavaScript ile yazılmış olacaktır." JavaScript Web' de baskın istemci tarafı dili olarak konumuna şans ile ulaşmadı. Gerçekten de dünyada ön plana çıkan bir dil olması şaşılabilecek bir şey değildir. Bugüne kadar geliştirilen en gelişmiş ve etkileyici programlama dillerinden biridir. JavaScript' in avantajlı yönlerini açıklayan aşağıdaki bölümlerde aşına olabileceğiniz bazı özellikleri özetlenmektedir [41].

Modern tarayıcılarda pek çok JavaScript, derlenmiş, son derece optimize edilmiştir ve yerel kod gibi çalıştırılır. Bu nedenle çalışma zamanı performansı C veya C ++ ile yazılmış bir yazılım olduğu hale yakındır. Tabi ki, çöp toplama ve dinamik bağlayıcı hala vardır ve bu şekilde bazı şeyleri daha hızlı yapmak mümkündür. Node.js (Google'ın son derece optimize V8 JavaScript motoru üzerine inşa edilmiş yüksek performanslı, olay tabanlı, sunucu tarafı JavaScript ortamı) ile JavaScript uygulamaları olay güdümlüdür ve genellikle JavaScript ile düşük dinamik diller arasındaki kod yürütülmesi için yapılan iyileştirmeden daha fazlasıdır.

JavaScript, çok zengin nesne yönelimli (OO) özelliklere sahiptir. Hem iletişim hem de veri kalıcılığı için neredeyse tüm modern web uygulamalarında kullanılan JSON (JavaScript Object Notation) standardı JavaScript'in mükemmel nesne-literal notasyonunun bir alt kümesidir.

JavaScript örnek miras modelini kullanır. Sınıfların yerine nesne örnekleri vardır. Yeni nesneler otomatik olarak örnek zinciri yoluyla yöntemleri ve üst nesnelerinin özelliklerini devralır. Herhangi bir zamanda bir nesnenin örneğini değiştirmek ve JavaScript' i çok esnek ve dinamik bir dil yapmak mümkündür.

JavaScript sözdizimi, C ++, Java, C # ve PHP gibi C ailesi dilleri ile deneyimleri olan hemen herkes için tanıdık olmalıdır. JavaScript' in popülaritesi bu yakınlığındandır. Ancak JavaScript' in bütün bu dillerden çok farklı çalıştığını anlamak gerekir.

JavaScript'in nesne-literal sözdizimi çok basittir, kısa, öz ve esnektir. Yerine geçtiği Xml' den daha kompakt ve esnek olan JSON biçiminde istemci / sunucu iletişimi için baskın standart olmak üzere uyarlandı.

JavaScript' te nesneler sonradan eklenilmemiştir. Fonksiyonlar da dahil neredeyse her şey birer nesnedir. Bu özelliğinden dolayı fonksiyonlar, fonksiyon çağırımlarındaki parametreler dahil bir değişken kullanabileceğiniz her yerde kullanılabilirler. Bu özellik genellikle asenkron işlemler için anonim geri çağırım işlevlerini tanımlamak için ya da daha yüksek mertebeden fonksiyonları (diğer fonksiyonları parametre olarak alan fonksiyonlar, bir fonksiyona geri dönüş veya her ikisi) oluşturmak için kullanılır. Çoğunlukla tekrar eden kodlama kalıplarını, örneğin tekrarlanan döngüler veya diğer yönerge dizileri, soyutlamak için fonksiyonel programlama stilinde yüksek mertebe fonksiyonlar kullanılır.

Tarayıcı içerisinde, her şey bir olay döngüsünde çalışır. JavaScript kodlayıcıları hızla olay işleyicilerinin sınırlarını düşünmeyi öğrenirler ve bunun sonucunda deneyimli JavaScript geliştiricilerinin kodları organize ve verimli olurlar. Diğer dillerde işlemleri engelleyebilen operasyonlar, JavaScript' te eş zamanlıdır.

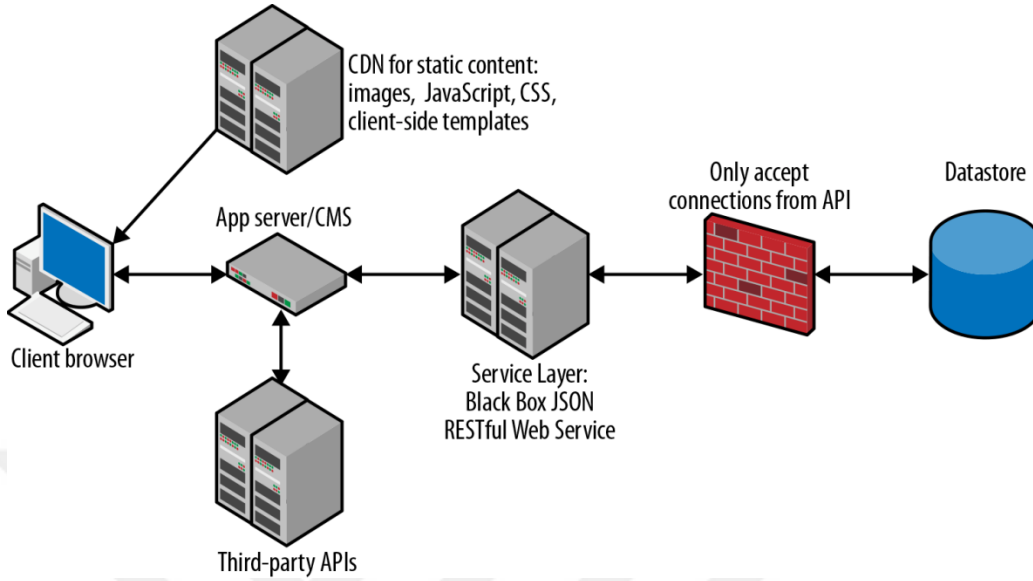
JavaScript kodu, her alanda çalışma özelliği sayesinde, en taşınabilir ve yeniden kullanılabilir koddur. Bu dil size, hem sunucu hem de istemcide çalışan aynı kodu yazmanıza olanak sağlar. JavaScript modüler ve kapsamlı olabilir ve aralarında iletişim olmayan, aynı sayfada uyum içerisinde çalışan farklı geliştirici ekipleri tarafından yazılan kodları görebilirsiniz [41].

Modern bir JavaScript uygulamasının anatomisine bakılırsa; her uygulama benzersiz olduğu halde sunucu altyapı, kaynak yönetimi, sunum ve kullanıcı ara yüzü davranışları gibi bazı ortak endişeler mevcuttur.

Altyapı çok farklı şekillerde, birçok saklama mekanizmasına sahip olabilir. Genellikle şunlardan oluşur (geriden ileriye) [41]:

- Bir veri deposu
- Sanal özel ağ (VPN) veya güvenlik duvarı (yetkisiz erişime karşı veri deposunu korumak için)
- Bir kara kutu JSON RESTful Web Servis Katmanı
- Çeşitli üçüncü taraf API' leri
- İstekleri yönlendirmek ve sayfaları istemciye iletmek için bir app sunucu / içerik yönetim sistemi (CMS)
- Saklanan dosyalar (görüntüler, JavaScript, CSS ve istemci tarafı şablonları gibi) için bir statik içerik iletim ağı (CDN)
- İstemci (tarayıcı)

Şekil 3.1’ de bunların bir aradaki uyumu gösterilmiştir.



Şekil 3.1. JavaScript altyapısı

Bu bileşenlerin çoğu kendini açıklayıcı olmakla birlikte, bazı önemli noktalar da vardır. Uygulama verisinin depolanması ve iletişimi konusunda dikkatli olunmalıdır. Veri deposu adından da anlaşıldığı üzere uygulama verilerini saklamak için bir yerdir. Bu genellikle bir planlı sorgu dili (Structured Query Language - SQL) arayüzü ile ilişkisel veri tabanı yönetim sistemidir (RDBMS), fakat NoSQL çözümleri giderek popülerlik kazanıyor. Gelecekte birçok uygulama her ikisinin birleşimini kullanacağı benziyor.

JavaScript nesne notasyonu (JSON) veri gösterimi, iletişim ve depolama için kullanılmak üzere yerli JavaScript nesne sözdiziminin bir alt kümesini belirten Douglas Crockford tarafından geliştirilen açık bir standarttır.

JSON’ dan önce, çoğu istemci-sunucu iletişimi çok gereksiz XML parçacıkları ile sağlanıyordu. JavaScript geliştiricileri sık sık JSON web hizmetleri ile çalışırlar ve dahili verileri tanımlarken çoğunlukla JSON sözdizimini kullanırlar.

Kitap koleksiyonunu anlatan aşağıdaki örnek mesaja bakılırsa:

```
[
  {
    "başlık" : "JavaScript: İyi Bölümler",
    "yazar" : "Douglas Crockford",
    "ISBN" : "0596517742"
```

```
},  
{  
  "başlık" : "JavaScript Kalıpları",  
  "yazar" : "Stoyan Stefanov",  
  "ISBN" : "0596806752"  
}  
]
```

Görünen üzere bu biçim, bir çift önemli farklılıkla birlikte, JavaScript’ in nesne sözdizimi ile neredeyse aynıdır:

- Tüm özellik isimleri ve string değerleri çift tırnak içine alınmalıdır. Diğer değerler asıl biçimlerinde görünebilirler.
- JSON kayıtları döngüsel başvurular içeremez.
- JSON fonksiyonlar içeremez.

Genişletilebilir İşaretleme Dili (XML) ve JSON veri depolarının gelişinden evvel, neredeyse tüm web hizmetleri RDBMS tarafından desteklenirdi. Bir RDBMS soyut veri noktalarını bir tabloda tutar ve SQL sorgu cevaplarında tablo aramalarını kullanarak veri çıkışını gruplandırır.

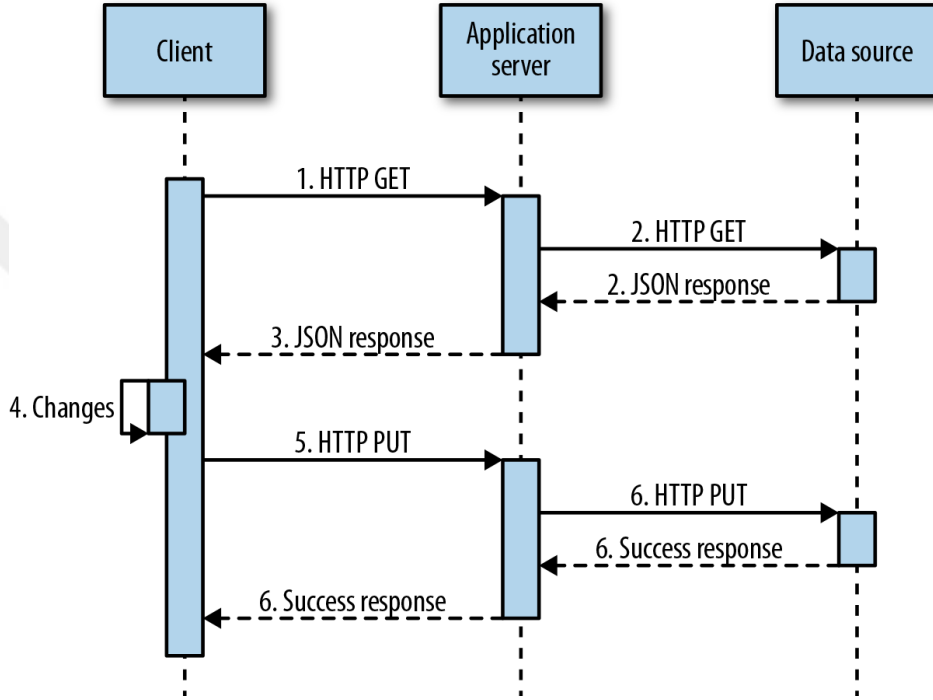
NoSQL veri depoları, aksine, tüm kayıtları tablo şeklinde yapılandırılmış depolamaya başvurmadan döküman ve doküman parçaları içerisinde saklar. Nesne yönelimli veri depoları genelde JSON formatını kullanmalarına rağmen verileri çoğunlukla XML formatında saklarlar. Web uygulamaları geliştirmek için JSON daha uygundur; çünkü JavaScript’ te yerel olarak iletişim kurmak için kullanılan format JSON’ dur.

Popüler JSON tabanlı NoSQL veri depoları örnekleri içerisinde MongoDB ve CouchDB bulunur. NoSQL’ nin yeni popüleritesine rağmen, hala MySQL ve benzeri RDBMS tarafından desteklenen modern JavaScript uygulamaları yaygın olarak bulunur. REST (Representational State Transfer), veri kaynakları ve kullanıcı ara yüzü arasında bir görev ayrılığı yaratan istemci-sunucu iletişim mimarisidir. REST uygulayan hizmetler, RESTful olarak adlandırılır. Sunucu veri kaynaklarını (kullanıcı kayıtları gibi) yönetir. Kullanıcı ara yüzü ile ilgilenmez. İstemciler bu konuda serbesttir. REST mimarisi kullanıcı ara yüzünün nasıl uygulandığıyla ilgilenmezler, sadece istemci ve sunucu arasındaki uygulama durumunu korumaya uğraşırlar.

RESTful web servisleri, istemcinin isteklerini sunucuya anlatan http eylemlerini kullanırlar. Desteklenen eylemler şunlardır [41]:

- HTTP POST: Kaynak koleksiyonunda yeni bir giriş oluşturur.
- HTTP GET: Bir kaynak temsilini alır.
- HTTP PUT: Kaynağı günceller.
- HTTP DELETE: Bir kaynak siler.

Şekil 3.2 tipik bir durum transfer akışını gösterir.



Şekil 3.2. REST düzeni

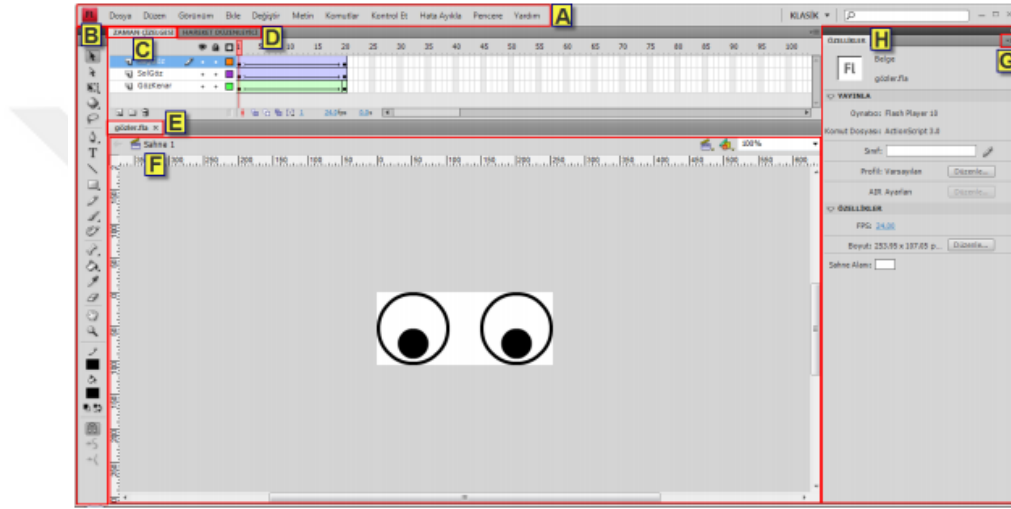
3.3. FLASH ve Actionscript

Adobe Flash programı, Adobe şirketine ait internet kullanıcılarına daha görsel içerikler sunabilmek amacıyla hazırlanmış site yöneticileri ve grafikçiler tarafından yaygın olarak kullanılan ve vektörel tabanlı olan bir grafik yazılımıdır. Flash vektör grafikleri kullanır, yani Flash' ta hazırlanan efektler, animasyonlar veya çizgi filmler aslında temelde matematiksel hesaplamalarla ortaya çıkar. Flash' ta hazırlanan içerikler ve çizimler tamamen matematiksel denklemler ve hesaplamalar üzerinde kuruludur, yani piksellerden oluşmaz. Bunun bir getirisi olarak Flash ile hazırlanmış bir nesne için görüntü yakınlştırılsa dahi bir bozulma meydana gelmez [24].

Adobe Flash ile grafik, video, dijital animasyon ve efektlerle desteklenmiş ilgi çekici uygulamalar ve etkileşimli Web siteleri oluşturmak mümkündür. İçerikler yazılımcının

tercihine bağılı olarak doğrudan Flash içerisinde oluşturulabilir veya diğer Adobe uygulamalarından içe aktarılabilir. Flash programı ile birlikte grafiksel animasyon ve ara yüzler oluşturulurken kullanıcı ile daha etkin bir yapıda olmaları için Actionscript adı verilen bir dil de kullanıma sunulmuştur.

Flash Çalışma Alanı; Şekil 3.3’ teki gibi, menü çubuğu (A), araçlar paneli (B), zaman çizelgesi (C), hareket düzenleyici (D), sekmeler (E), sahne (F), paneller (G) ve özellikler (H) bölümlerinden oluşur [42].



Şekil 3.3. Flash çalışma alanı bölümleri

Görsel ara yüzün arka planında çalışan kodlama kısmı Actionscript programlama dili ile hazırlanır. Actionscript, Javascript’ in temelini oluşturan ECMAscript üzerine inşa edilen nesne yönelimli bir programlama dilidir. Bu sayede birçok yazılımcı için anlaması kolay bir dildir. Genelde geliştiriciler tarafından sunum seviyesi mantığını tasarlamak için kullanılır [25].

Flash Player ve AIR uygulamalarının bir parçası olan ActionScript Virtual Machine (AVM) tarafından çalıştırılan ActionScript kodu, genelde bir derleyici ile byte kodu formatında dönüştürülür. Bu byte kodu, Flash Player veya AIR gibi uygulamalar tarafından çalıştırılabilen SWF dosyalarında gömülüdür.

Actionscript, günümüze dek olan gelişim sürecinde önceki sürümlere göre çok daha gelişmiş olan ve önemli bazı özellikler sunan Actionscript 3.0 halini almıştır. Bu özellikler arasında [43]:

- Farklı bir bayt kodu kullanan yeni bir ActionScript Virtual Machine (AVM2) ile büyük ölçüde performans iyileştirmesi sağlanmıştır.
- Önceki derleyici sürümleri ile karşılaştırıldığında daha derin iyileştirmeler yapabilen yeni bir derleyici kodu vardır.
- Daha gelişmiş ve kapsamlı bir uygulama programlama arabirimi (API) vardır.
- XML için ECMAScript' i esas alan özel bir XML arabirimi vardır.



4. RASPBERRY PI

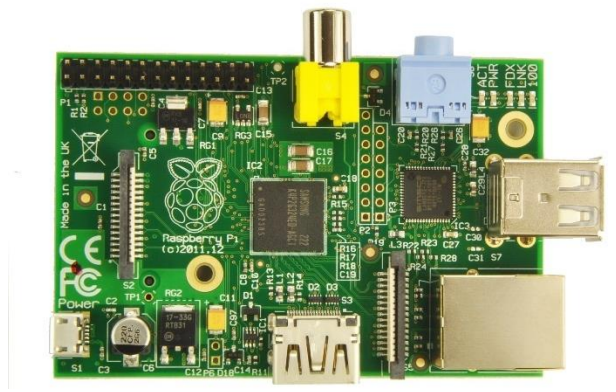
Kredi kartı büyüklüğünde, tek kartlı mini bir bilgisayar olan Raspberry Pi, Birleşik Krallık' ta Raspberry Pi Vakfı tarafından bilgisayar bilimini öğretmek üzere okullar için geliştirilmiştir.

Raspberry Pi bilgisayarı RS Components, Element 14/Premier Farnell ve Egoman gibi farklı firmalar tarafından üretilmektedir. Ayrıca bu firmalar aracılığıyla internet üzerinden satışa sunulmaktadır.

Raspberry Pi' nin ilk modellerinde Broadcom BCM2835 mikroçipi kullanılmıştır ve bu çip ARM1176JZF-S 700 MHz merkezi işlem birimini barındırır. Ardından üretimine geçilen Raspberry Pi 2 modelinde ise Broadcom BCM2836 çipi kullanılmıştır. Bu modelde VideoCore IV GPU grafik işlem birimi vardır. Sistem yüklemesi ve verilerin depolanması amacıyla SD kart kullanır. Üzerinde HDMI video çıkışı, ses çıkışı, USB 2.0 portları, GPIO arayüzü, 5V MicroUSB güç girişi ve MIPI kamera girişi bulunmaktadır.

Vakfın web sitesinden Pidora (Fedora tabanlı), Raspbian (Debian Wheezy tabanlı), Snappy Ubuntu Core veya desteklenen diğer işletim sistemleri indirilebilir. Sitesi dışındaki Windows 10 IoT Core, Arch Linux ARM, Pardus ARM gibi işletim sistemlerini de destekler. Python, BBC Basic, Perl ve C gibi programlama dilleri ile programlanabilmektedir.

Raspberry Pi Vakfı, Velocix ve IndieCity firmaları, kullanıcıların faydalanabilmeleri amacıyla 17 Aralık 2012 tarihinde Pi Store' yi açmışlardır. Böylelikle kullanıcıların farklı kategorilerdeki uygulamaları bulup indirmelerine imkan sağlanmıştır. Ayrıca kullanıcılar tarafından geliştirilen uygulamalar da buraya yüklenebilir ve onay işleminin ardından diğer kullanıcılarla paylaşma sunulabilir [26].



Şekil 4.1. Raspberry Pi üst görünüm

Raspberry Pi, farklı versiyonlarıyla piyasada bulunmaktadır. Her versiyonun birbirinden farklı amaçlar için tasarlanmış olabileceği gibi, yeni donanım özellikleriyle yükseltilmiş bir versiyon olarak da karşımıza çıkmaktadır. Bu tezde Şekil 4.1’ de gösterildiği gibi Raspberry Pi B rev2 kullanılmıştır ve bu versiyon üzerinde donanım bileşenlerinden ve teknik özelliklerden bahsedilecektir.

Öncelikle Raspberry Pi için önerilen güç kaynakları aşağıdaki gibi olmalıdır.

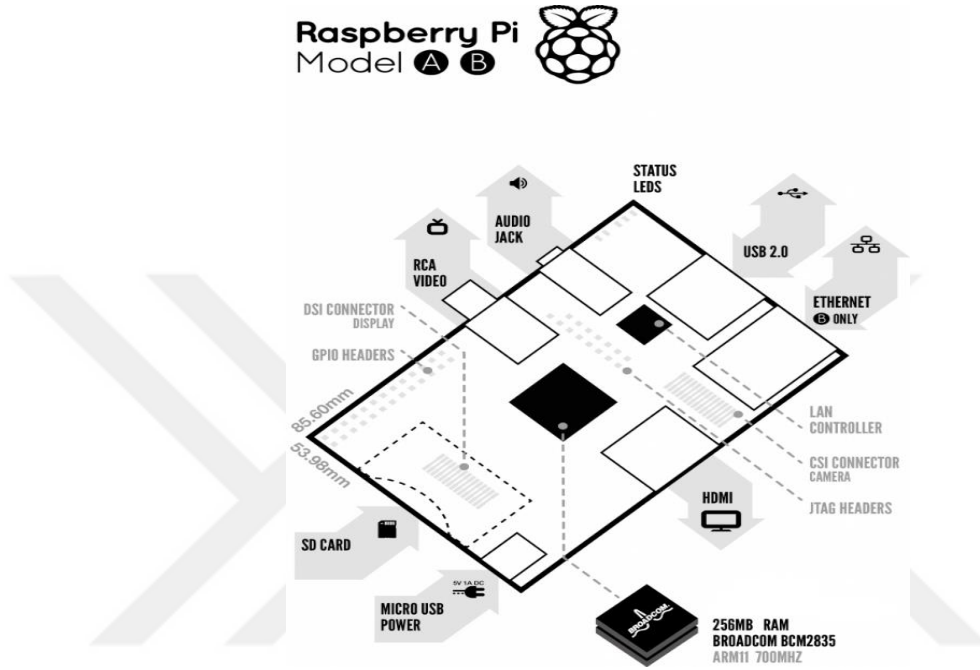
- Model A: 5V dc, 500-700mA
- Model B: 5V dc, 700-1500mA

4.1. Donanım

Raspberry Pi donanımı bellek kapasitesi ve çevresel aygıt desteği özelliklerine göre farklı versiyonlar halinde geliştirilmiştir. Kullanmış olduğumuz modele ait teknik özellikler aşağıda listelenmiştir ve modele ait bileşenler Şekil 4.2 üzerinde gösterilmiştir [27].

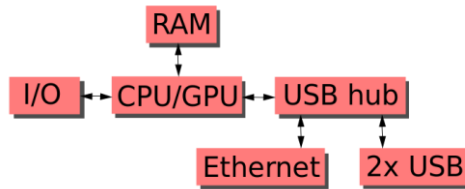
- Çip: Broadcom BCM2835 (CPU, GPU and SDRAM)
- İşlemci: 700 MHz ARM1176JZF-S çekirdeği (ARM11 ailesi)
- Grafik işlemci: Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264/MPEG-4 AVC High-Profile decoder
- Hafıza (SDRAM): 512 Megabayt
- Video çıkış: Composite RCA, HDMI
- Audio çıkış: 3.5 mm, HDMI
- Mass storage: SD/MMC slot

- USB: 2 USB2.0 port
- Network: 10/100 Ethernet RJ45
- Güç: 5V/700 mA (3.5 W) via microUSB
- Boyutlar: 85.6 mm x 53.98 mm



Şekil 4.2. RPI bileşenlerinin yerlerini belirten bir diyagram

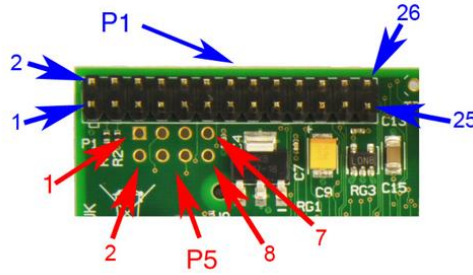
A, B, A + ve B + modellerine uygun blok şeması Şekil 4.3’ te gösterilmiştir. Ethernet ve usb hub gibi bileşenler bazı modellerde bulunmayabilir.



Şekil 4.3. Raspberry Pi B Rev2 blok şeması

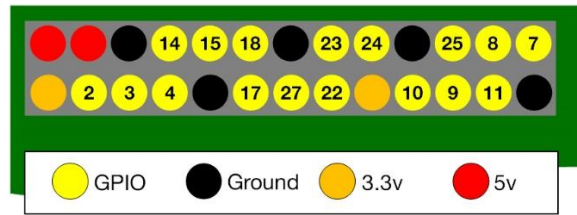
Şekil 4.4’ te görüldüğü üzere Raspberry Pi Model B Revizyon 2.0, toplamda 21 GPIO (genel amaçlı giriş/çıkış) olmak üzere, 17 GPIO pinine erişim sağlayan ve P1 olarak

isimlendirilen 26 pinlik bir genişleme tablası ve ek olarak 4 GPIO pini sağlayan, P5 isimli 8 pinlik bir genişleme tablası bulunmaktadır [28].



Şekil 4.4. Raspberry Pi' nin pinlerini gösteren üstten görünümü

Raspberry Pi' nin güçlü özelliklerinden birisi de giriş çıkış pinlerinin, kartın kenar kısmında bulunmasıdır. Bu pinler dış dünya ve Pi arasındaki fiziksel arayüzdür. Bu pinleri açıp kapatabildiğiniz veya Pi' nin açıp kapatabildiği anahtarlar gibi düşünebilirsiniz. Bu pinlerden GPIO haricindekiler, Şekil 4.5' te görüldüğü gibi güç ve topraklama pinleridir.

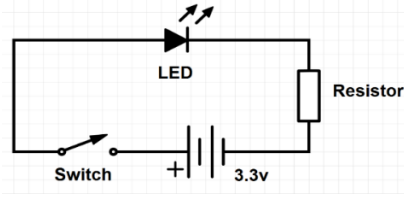


Şekil 4.5. GPIO, güç ve topraklama pinleri

Pinler, gerçek dünya ile birçok farklı şekilde etkileşim kurmak için kullanılabilir. Girişler sadece fiziksel bir anahtardan gelmek zorunda değildir; başka bir bilgisayardan, cihazdan veya bir sensörden de veri girişi yapılabilir. Çıkışlar ile de bir LED' i yakmaktan, başka bir cihaza sinyal veya veri göndermeye kadar pek çok şey yapılabilir. Eğer Raspberry Pi bir ağa dahilse, her yerden kendisine bağlı cihazları kontrol edebilirsiniz ve bu cihazlardan geriye veri dönütleri alabilirsiniz. Bağlantı ve internet üzerinden fiziksel cihazların kontrolü güçlü ve heyecan verici bir şeydir ve Raspberry Pi bunun için idealdir [29].

GPIO pinlerinin çalışma mantığına bakılacak olursa; öncelikle pi üzerindeki GPIO' ları kullanırken dikkatli olunmalıdır. Rastgele bağlanan kablolar ve güç kaynakları, Pi' ye zarar verebilir. Motorlar gibi çok güç kullanan parçalar Pi' ye bağlanırken kötü şeylerin olabileceği akılda tutulmalıdır. Bir an için Pi' yi göz ardı edecek olursak; yapabileceğimiz

basit elektrik devrelerinden biri Şekil 4.6’ daki gibi güç kaynağı, direnç, ışık kaynağı ve anahtardan oluşan yapıdır.



Şekil 4.6. Basit bir elektrik devresi

Bir GPIO’yu çıkış olarak ayarladığımız zaman, Raspberry Pi şeklindeki anahtarı kapalı konuma getirmiş gibi olur. Her pin açık veya kapalı konuma getirilebilir. Pinlerin açık konumdaki halleri için literatürde “High”, kapalı konumları için ise “Low” ifadeleri kullanılır. Pin High konumunda 3.3V çıkış verir, Low konumunda ise kapalıdır.

GPIO çıkışları kolaydır, çünkü ya açıklardır ya da kapalı, yani 3.3V veya 0V. Girişler ise dijital cihazların çalışma mantıklarından dolayı biraz daha dikkat isterler. Bir butonu, bir giriş pini ve topraklama pini arasına bağlamak mantıklı gelse de, Pi’ye butonun açık veya kapalı olup olmadığını kodlama kısmında anlatmak gerekir.

GPIO pinlerini programlarken iki farklı yol kullanılır: GPIO numaralandırma ve fiziksel numaralandırma. GPIO numaralandırma, bilgisayarın gördüğü GPIO pinleridir. Sayılar insanlar için bir anlam ifade etmez ve hatırlanmaları çok da kolay değildir. GPIO pinlerini kullanırken, hepsini gösteren bir materyale ihtiyaç duyarsınız. Pinler için tercih edilen diğer bir yol da fiziksel numaralandırma dediğimiz basitçe sıralanmalarıdır. Kullanımları yönünden sadece kodlama kısmında küçük farkları vardır.

Yeni başlayanlar genelde kolay olduğu için fiziksel numaralandırmayı kullanırlar. Yine de hangi pinlerin GPIO, topraklama veya güç olduğunun anlaşılması için pinleri gösteren bir şekle ihtiyaç vardır. Genellikle bir çok sistemde GPIO numaralandırma kullanılır. Aslında bir programda aynı sistemşi kullandığınız sürece hiçbir sorun olmayacaktır. Bazen de pin numaralandırması kullandığınız programa bağlı olabilir. Örneğin Scratch GPIO, fiziksel numaralandırmayı kullanırken, Python’da istediğinizi kullanabilirsiniz. Şekil 4.7’de hem fiziksel numaralandırma hem de GPIO numaralandırma açıkça gösterilmiştir.

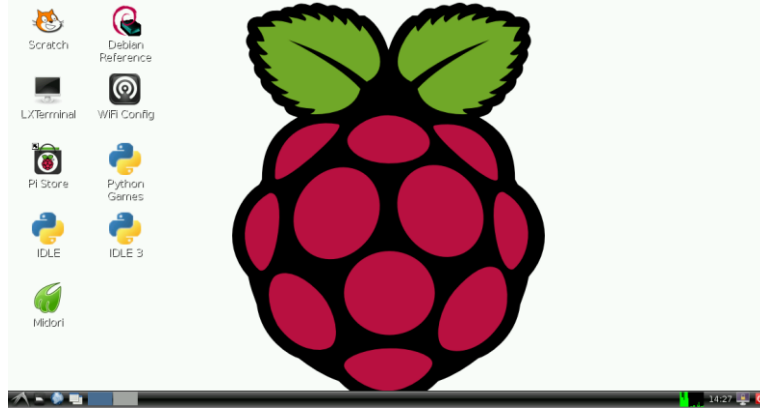
GPIO#	2nd func	pin#	pin#	2nd func	GPIO#
N/A	+3V3	1	2	+5V	N/A
GPIO2	SDA1 (I2C)	3	4	+5V	N/A
GPIO3	SCL1 (I2C)	5	6	GND	N/A
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
N/A	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18
GPIO27	GEN2	13	14	GND	N/A
GPIO22	GEN3	15	16	GEN4	GPIO23
N/A	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	N/A
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
N/A	GND	25	26	CE1_N (SPI)	GPIO7

Function	2nd func	pin#	pin#	2nd func	Function
N/A	+5V	1	2	+3V3	N/A
GPIO28	GPIO_GEN7	3	4	GPIO_GEN8	GPIO29
GPIO30	GPIO_GEN9	5	6	GPIO_GEN10	GPIO31
N/A	GND	7	8	GND	N/A

Şekil 4.7. Raspberry Pi' nin pin yapısı

4.2. Yazılım

Raspberry Pi öncelikle Linux kernel tabanlı işletim sistemlerini kullanıyor. Pi' nin kalbindeki ARM11 çipi (birinci kuşak modeller), ARM' nin 6. Sürümünü temel almıştır. Ubuntu' nun geçerli sürümü Raspberry 2' yi destekler. Ubuntu ve bir çok popüler Linux versiyonu, ARM11 üzerinde çalışan eski sürümü Raspberry Pi 1' i desteklemez. Raspberry Pi 2 ayrıca Windows 10 IOT çekirdek işletim sistemini çalıştırabilir; ama Pi' nin hiçbir versiyonu geleneksel Windows' u çalıştıramaz. Bununla birlikte Raspberry Pi 2 şu anda da Raspbian, OpenELEC ve RISC OS gibi işletim sistemlerini destekliyor [26].



Şekil 4.8. Raspbian ekran görüntüsü

Raspberry Pi için var olan kurulum yöneticisi NOOBS' tur. Noobs ile birlikte yüklenebilen işletim sistemleri şunlardır:

- Arch Linux ARM
- OpenELEC
- Pidora (Fedora Remix)
- Puppy Linux
- Raspbmc ve XBMC açık kaynak dijital medya merkezi
- RISC OS - İlk ARM tabanlı bilgisayarın işletim sistemidir.
- Raspbian (Raspberry Pi 1 için önerilir) –Şekil 4.8' de ekran görüntüsü verilen Raspbian, Raspberry Pi üreticileri tarafından da desteklenen bir işletim sistemidir. Öncelikle ARMv7 ve sonraki işlemciler için tasarlanmış Debian hard-float (armhf) mimarisine dayanmaktadır. RPi üreticileri tarafından sağlanan Raspbian imajlarını yüklemek için en az 4 gb' lık bir sd karta ihtiyaç vardır [26].

4.3. I2C ve SPI Haberleşme

Günümüzde var olan elektronik ürünlerden birçoğu bir veya daha fazla seri veri yolu içerir. Bir sistemde seri veri yolunu kullanmak, paralel veri yolu bağlantılarına göre birçok avantaj sağlar. Bu avantajları şu şekilde sıralayabiliriz [31]:

- Düşük bileşen maliyeti
- Küçük baskılı devre kartı
- Basitleştirilmiş tasarım
- Düşük güç tüketimi

Bu avantajlar, elektronik ürünlerinde neden seri veri yolu bağlantılarının tercih edildiğinin en büyük göstergelerinden bir kaçıdır. Tabi ki avantajlarının yanı sıra paralel bağlantılara göre eksileri de bulunmaktadır. Bu bağlamda paralel veri yoluna göre akla gelen ilk eksi, daha düşük veri aktarım hızlarına sahip olmalarıdır. Seri veri yollarında, cihaza doğru veya aksi yönde veri sadece bir bağlantı üzerinden taşınır. Bazı durumlarda cihaz içerisine doğru veri taşınırken bir sinyal ve cihazdan dışarı doğru veri taşınırken başka bir sinyal kullanılır. Öte yandan paralel veri yollarında, verileri cihaz içerisine ve aksi yönde aynı anda birçok sinyal taşır. Aşağıdaki tabloda bu veri yollarının karşılaştırmasını görebilirsiniz [31].

Tablo 4.1. I2C ve SPI seri veri yollarına genel bakış

Veri Yolu	Yapı	Özellik	Multi- Master	Veri Hızı	Flyby Veri Transferi	Tam Çift Yönlü
SPI (Serial)	Ortak tek yönlü 2 veri sinyali ve ortak saat	Ortak veri yolunda çift yönlü iletişim	Mümkün, ama standart değil	1 Mbps	Hayır	Evet
I2C (Inter IC)	Ortak veri sinyali ve ortak bir saat sinyali	Ortak veri yolunda çift yönlü iletişim	Evet	100 kbps, 400 kbps, 3.2 Mbps	Evet	Hayır

I2C ve SPI, iletişim protokolleri arasında düşük maliyetleri nedeniyle öne çıkmaktadır. Her iki protokol de on-board çevre birimleri ile yavaş iletişim için ve tümleşik devreler arasındaki iletişim için iyi uygundur. Bu iki popüler protokolün geçmişine bakılırsa iki büyük şirket – I2C için Philips ve SPI için Motorola – ve protokollerin neden, ne zaman ve nasıl ortaya çıktığıyla ilgili iki farklı hikaye ile karşılaşılır.

I2C, 1982 yılında geliştirildi. Asıl amacı bir televizyon setindeki çevresel yongalara bir işlemci bağlamak için kolay bir yol sağlamak oldu. Gömülü sistemlerde çevresel aygıtlar genellikle bellek eşlemeli G / Ç aygıtları gibi mikro denetleyicilere bağlanır. Bunu yapmanın bilinen bir yolu, çevre birimlerini mikro denetleyicinin paralel adres ve veri yoluna bağlamaktır. Bu, bağlı tüm çevre birimlerinin adres yollarını çözümlemek için baskılı devre kartında çok sayıda kablo bağlantısına neden olur. Mikro denetleyici pinlerini iradeli kullanmak ve baskılı devre kartını daha basit hale getirmek için, başka bir deyişle

daha düşük maliyetler için, Eindhoven' daki Philips laboratuvarlarında, bütün çevre birimlerini mikro denetleyiciye bağlamak için sadece iki bağlantıya ihtiyaç duyan IIC veya I2C (Inter-Integrated Circuit) protokolü geliştirildi. Orijinal halinde 100 kbps veri yolu hızı (saniyede 100 kilo bit) tanımlanmıştır. Bu yapısı defalarca gözden geçirildi, önce 1995 yılında 400 kbps hız tanımlandı ve daha hızlı çevre birimleri için 1998' den bu yana 3.4 Mbps hızına kavuştu.

SPI (Serial Peripheral Interface), ilk olarak 1979 yılında, popüler Motorola 68000 mikro işlemcisi ile aynı mimariye sahip ilk mikro denetleyici ile tanıtıldı. SPI tanımlı harici mikro denetleyici veri yolu, mikro denetleyici çevre birimlerini dört bağlantı ile bağlamak için kullanıldı. I2C' nin aksine SPI' nin resmi bir spesifikasyonunu bulmak zordur. Detaylı bilgiler için mikro denetleyicilerinin veri sayfaları ve ilgili uygulama notları okunmalıdır [32].

SPI ve I2C, çok geniş bir yelpazede cihazı kontrol etmek için kullanılabilir. Yaygın örnekler arasında [33]:

- Dijital-analog dönüştürücüler (DAC)
- Analog-dijital dönüştürücüler (ADC)
- LCD ekranlar
- OLED ekranlar
- Klavyeler
- Motor sürücüler
- LED sürücüler
- Bellek yongaları ve kartları (EEPROM, RAM, FERAM, Flaş)
- Veri yolu genişleticiler / uzatıcılar
- diğer mikrodeneleyiciler

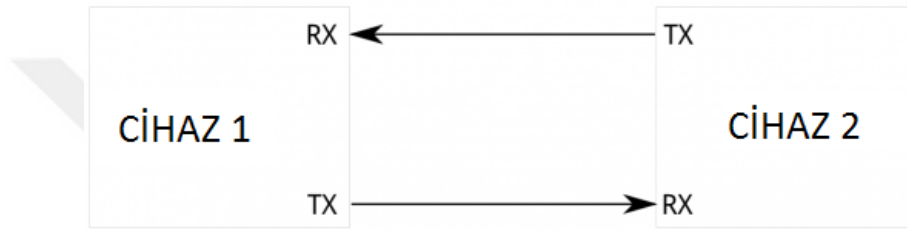
4.3.1. I2C

I2C başlangıçta çeşitli Philips yongaları için Philips tarafından 1982 yılında geliştirilmiştir. Kısa mesafeler için geliştirilmiş, düşük bant genişliğine sahip, basit bir protokoldür. Mevcut I2C ile haberleşen cihazların çoğu 400kbps' ye kadar hızlarda çalışmaktadır.

Inter-entegre devre (I2C) protokolü, birden fazla slave dijital entegre devrenin (çip) bir veya daha fazla master çiple iletişim kurması için tasarlanmış bir protokoldür. Seri çevresel

ara yüz (SPI) gibi, tek bir cihaz içinde, kısa mesafe haberleşmeler için tasarlanmıştır. Asenkron seri ara yüzler (örneğin RS-232 veya UART) gibi, bilgi alışverişi için sadece iki sinyal bağlantısı gereklidir [30].

Seri portlarla bir kıyaslama yapılacak olunursa; seri portlar asenkrondur, yani iletilen saat verisi yoktur ve bunları kullanan cihazların bir veri hızında önceden anlaşmış olmaları gerekir. İki cihazın da aynı hızlara yakın saatleri olmalıdır. Her ikisinin saat hızlarında farklılıklar olacaktır ve bu da bozuk verilere sebep olacaktır. Seri port kullanan iki cihazın iletişimi Şekil 4.9’ da gösterilmiştir.

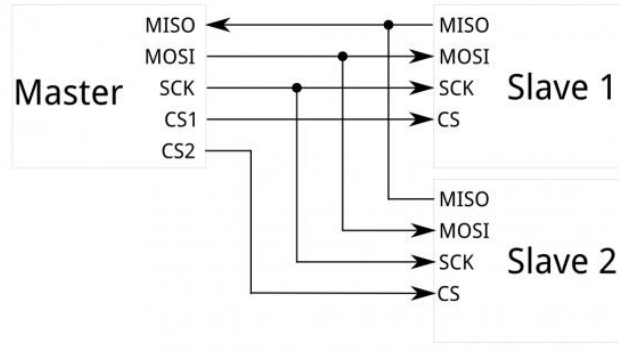


Şekil 4.9. Seri port ile iki cihazın iletişimi

Asenkron seri portlar, her iki uçta üst düzey donanımlara ihtiyaç duyarlar ve yazılımları doğru uygulamak oldukça karmaşık ve zordur. En az bir başlangıç ve bitiş biti her veri çerçevesinin bir parçasıdır. Bu da 8 bitlik bir veri için 10 bitlik bir aktarım zamanı gerekiyor demektir.

Asenkron seri portlarının başka bir eksi yönü ise, sadece iki cihaz arasında iletişime uygun olmalarıdır. Tek bir seri bağlantı noktasına birden fazla cihaz bağlamak mümkün olsa da veri yolu üzerindeki çekişme hep sorun olmuştur ve genellikle söz konusu cihazların zarar görmesini önlemek için dikkatli bir şekilde ele alınmalıdır.

Sonuç olarak veri hızı istenmeyen bir durumdur. Asenkron seri iletişim için hiçbir teorik sınır bulunmamakla birlikte, çoğu UART cihazlar sadece belirli bir baud hızını destekler ve bunlardan en yükseği saniyede yaklaşık 230400 bittir [30].



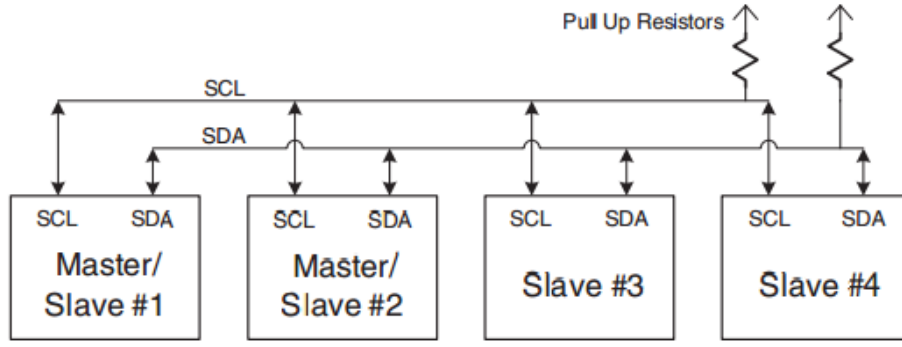
Şekil 4.10. SPI haberleşme protokolü master - slave bağlantısı

I2C ve SPI arasında bir kıyaslama yapılırsa; öncelikle SPI' nin en belirgin dezavantajı gerekli pin sayısıdır. Şekil 4.10' da görünen üzere tek bir master' ı tek bir slave' e bağlamak için dört hat gereklidir; her ek slave için, masterda bir çip seçim pini gereklidir. Birçok cihazın bir mastera slave olarak bağlanmak zorunda olduğu durumlarda pin bağlantılarının hızla çoğalması, kötü sonuçlar doğurabilir. Ayrıca her cihaz için çok sayıda bağlantının olması, yakın PCB düzeni durumlarında sinyalleri yönlendirmeyi zorlaştırabilir. SPI veri yolu üzerinde sadece tek mastera izin verir; ama istenilen sayıda slave' i destekler. Sadece veri yoluna bağlanan cihazların sürücü kapasitesi ve mevcut yonga seçme pin sayısına bağlıdır.

SPI, yüksek veri hızlı tam çift yönlü (eş zamanlı veri gönderme ve alma) bağlantılarda iyidir, bazı cihazlar için 10 Mhz' den fazla saat hızlarını (saniyede 10 milyon bit) destekler. I2C, asenkron seride olduğu gibi sadece iki bağlantıya ihtiyaç duyar; ama bu iki bağlantı 1008' e kadar slave cihazı destekler. Ayrıca SPI' nin aksine çoklu master sistemini destekler. Veri yolundaki bütün cihazlarla iletişim kurmak için birden fazla mastera izin verir. Yalnız master cihazlar veri yolu üzerinden kendi aralarında iletişim kuramazlar ve veri yolunu sırayla kullanırlar.

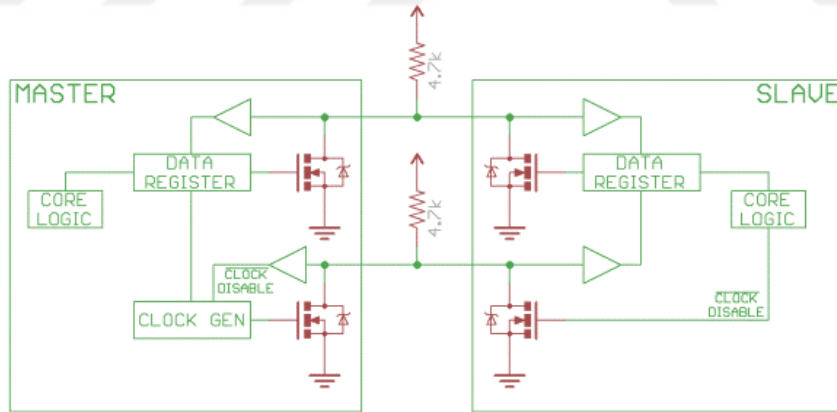
Veri hızları asenkron seri ve SPI arasında düşer. Bütün I2C cihazlar 100kHz veya 400kHz hızlarında haberleşebilirler. Gönderilen her sekiz bitlik veriyle bir extra meta veri bitinin de iletilmesi gerekir. I2C' nin uygulanması için gerekli donanım SPI' den daha karmaşık, ancak asenkron seriye göre daha az karmaşıktır [30].

Her I2C veriyolunda iki sinyal vardır: SCL ve SDA. SCL saat sinyali, SDA ise veri sinyalidir. Saat sinyali her zaman veri yolunda geçerli master cihazlar tarafından oluşturulur. Bazı slave cihazlar master' ın daha fazla veri göndermesini geciktirmek için saat sinyalini düşürmek üzere zorlayabilir. Bu "saat germe" olarak adlandırılır [31].



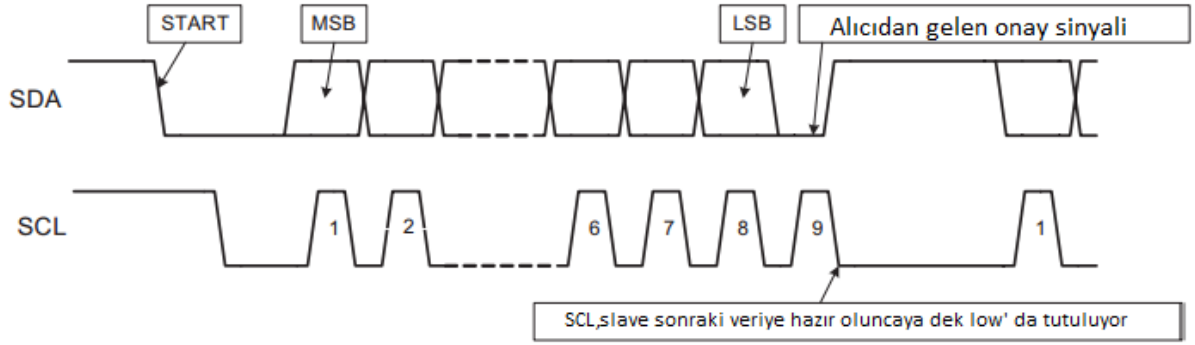
Şekil 4.11. Tipik bir I2C bağlantısı

UART ve SPI bağlantılarının aksine, I2C sürücülerini sinyal hattını lojik low' a çekebilir ama tekrar high değerine yükseltmez. Böylece, bir cihaz hattı high, bir diğeri low yapmak isterken yaşanabilecek karışıklıklar önlenmiş olur ve sistemdeki aşırı güç dağılımı ve sürücülere verilebilecek hasar potansiyeli ortadan kalkar. Her sinyal hattında cihazlardan hiçbiri hattı low yapmak istemiyorken sinyali tekrar high haline getirmek için Şekil 4.11 ve Şekil 4.12' deki gibi bir pull-up direnci vardır [30].



Şekil 4.12. İki iletim hattındaki pull-up dirençleri

I2C saat sinyali (SCL) geçerli master tarafından oluşturulur ve bütün cihazlar hazır olana kadar SCL sinyalini low' da tutar. Böylece SCL sinyali, Şekil 4.13' teki gibi en yavaş I2C cihazı hazır olana dek geciktirilmiş olur [31].



Şekil 4.13. I2C sinyalleri

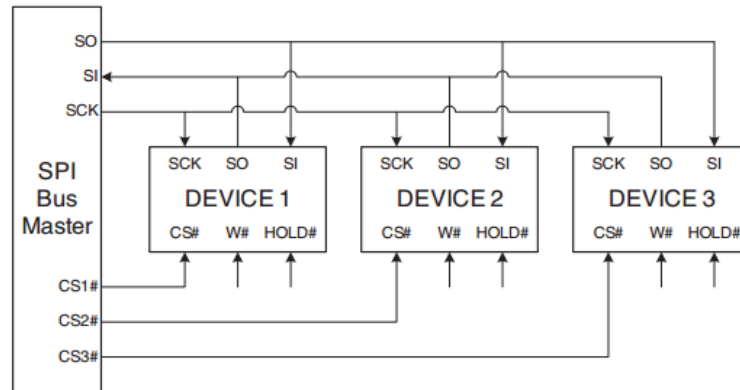
4.3.2. SPI

SPI Serial Peripheral Interface bus için kullanılan bir kısaltmadır. SPI tam çift yönlü veri aktarımı sağlar. Tipik olarak veri yolunda sadece bir master vardır ve bütün diğer SPI cihazları slave' dir.

SPI paylaşılan bir veri yoluna sahiptir. Master SO sinyali üzerinden veriyi iletir ve SI sinyal hattından da veri okur. Bu master' a eş zamanlı olarak veri okuma ve iletme şansı verir. Tüm veri aktarımları veri yolu master ve slaveler arasında yer almalıdır. Doğrudan iki slave cihaz (Flyby transferler) arasındaki veri aktarımlarına izin verilmez. SPI veri yolu Master sinyalleri aşağıda gösterilmiştir:

- SCK Master tarafından sunulan SPI saat sinyali
- SI Master için veri girişi
- SO Master' dan veri çıkışı

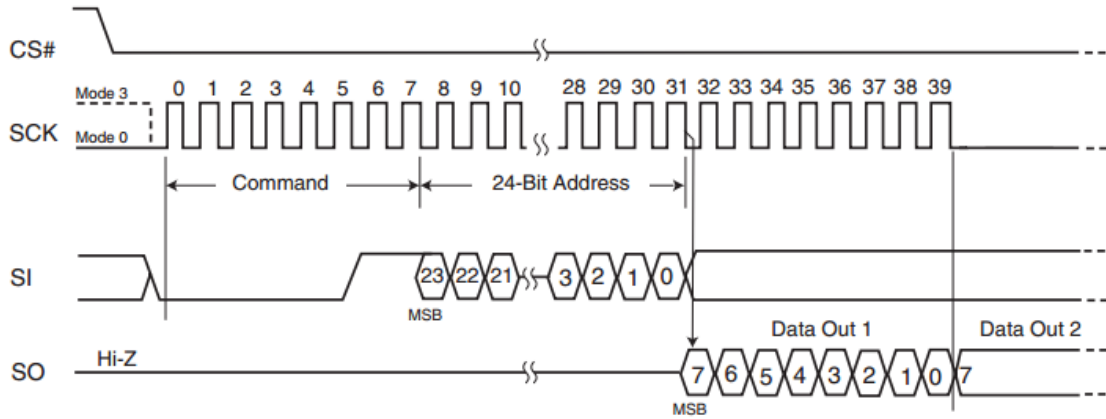
Şekil 4.14' te bir master ve üç slave cihazdan oluşan SPI sistem örneği verilmiştir. SPI Master her SPI slave cihaz için ayrı bir cihaz seçim sinyali verir [31].



Şekil 4.14. Tipik bir SPI bağlantısı

SPI veri döngüsünde olaylar aşağıdaki gibidir:

1. CS # flashı seçmek için low olur.
2. SPI okuma komutu gönderilir.
3. İç 24-bit flash bellek adresi gönderilir.
4. İlk flash veri baytı okunur.



Şekil 4.15. SPI veri döngüsü

Şekil 4.15’ te SCK sinyalinin solunda mode0 ve mode3 olmak üzere iki seçenek vardır. SCK sistemine bağlı olarak boştayken low veya high olabilir. Her iki modda da veri saatin yükselen kenarında başlar ve düşen kenarında biter [31].

5. İŞLETİM SİSTEMİ VE PROGRAMLAMA DİLİ

Bu tezde Linux tabanlı bir ARM işlemci kullanılmıştır. Bundan dolayı işletim sistemi ile uyum içerisinde çalışabilen ve sistemde mevcut olan Phyton programlama dili kullanılmıştır. Böylece yazılım doğrudan donanım içinde geliştirilmeye müsait bir yapı oluşturmaktadır.

5.1 Linux

Linux, 1991 yılında Linus Torvalds tarafından geliştirilmeye başlanmış ve geliştirilirken daha eski bir işletim sistemi olan Unix' ten esinlenilmiştir. Unix 70' li yıllarda büyük bilgisayarlar için geliştirilmiş çok kullanıcıli bir işletim sistemidir. Zaman içerisinde birçok türevi ortaya çıkmıştır. Bu türevlerin çoğu yüksek fiyatlara satılan ticari yazılımlardır. Birçoğunun da gelişimi belli bir noktaya kadar devam etmiş sonrasında durmuştur.

Açık kaynak kodlu olması ve serbestçe dağıtılabilmesiyle internet üzerinde ilgili birçok kişi tarafından gelişimine katkı yapılmaktadır. İşletim sistemi geliştirilirken oluşturulan sürümler internet üzerinde yayınlanmakta, meraklı kullanıcılar tarafından eksik ve hataları tespit edilmek üzere test edilmektedir. Bu şekilde dünyanın dört bir yanından insanların destekleriyle geliştirilmektedir. Bu süreçte zaman zaman geliştirilen parçalar toparlanıp kararlı bir sürüm oluşturulur ve yeni bir seri ile gelişime devam edilmektedir. Bu şekilde bazı sürümleri çok kısa süreler içerisinde güncellenebilmektedir. Zaten gelişiminin, eksik ve hata tespit sürecinin bu şekilde olması Linux' un en büyük avantajlarından biridir.

Linux, aslında Unix' in çalışma sistemini öğretmeyi amaçlayan, deneysel bir işletim sistemi olan Minix işletim sistemi üzerinde geliştirilmeye başlanmıştır. Kendi işletim sistemini yapmak ve Minix' den daha iyi bir işletim sistemi oluşturmak isteyen Linus Torvalds 1991 yılı ağustos ayında ilk çalışan Linux çekirdeğini geliştirmiştir. Akabinde Linux 0.02 olan ilk resmi sürüm 5 Ekim 1991 tarihinde yayınlanmıştır.

Linux, yayınlandığı tarihten günümüze hızlı bir gelişme göstermiş, farklı birçok ülkeden kullanıcıların beğenisini ve desteğini kazanmıştır. Günümüzde Linux, sadece bilgisayarlar üzerinde değil, mobil telefonlar, geliştirme kartları, televizyonlar, sunucular, otomobiller gibi hayatın farklı alanlarında kullanılmaktadır [52].

5.2. Python Programlama Dili

Python programlama dili, insanın düşünme tarzından yola çıkılarak geliştirilmiştir. Çok geniş bir yelpazede programcılığa ışık tutmaktadır. 1990 yılında Amsterdam’ da Guido van Rossum tarafından geliştirilmeye başlanan Python (piton), adını bir yilandan değil de geliştiricisinin çok sevdiği bir İngiliz komedi grubunun bir gösterisinden almıştır. Nesne yönelimli, modüler, etkileşimli, girinti ve çıkıntılara dayalı söz dizimi sayesinde öğrenilmesi ve akılda kalması kolay üst seviyeli bir programlama dilidir. Söz diziminin bu şekilde olması, dilin anlaşılmasını kolaylaştırır ve ayrıntılarla harcanan vakit kaybını en aza indirir. Modüler yapısı sayesinde her türde veri alanı girişini destekler ve hemen hemen tüm platformlarda çalışabilir. (Unix, Linux, Mac, Windows, Amiga, Symbian). Python ile sistemler programlanabilir, kullanıcı ara birimleri oluşturulabilir, ağ programlanabilir, veritabanı ve uygulama yazılımları geliştirilebilir.

Pardus Linux dağıtımı, YouTube, Django ve Zope uygulama sunucuları, orijinal BitTorrent istemcisi Python kullanılarak geliştirilen önemli projelerdendir. Ayrıca NASA, Google ve CERN gibi Python kullanan büyük kuruluşlar da vardır. Bunlardan başka Python, GIMP, Scribus, OpenOffice, Inkscape, Blender ve Paint Shop Pro gibi bazı programlarda betik dili olarak kullanılır. Üstelik Linux dağıtımlarının pek çoğunda Python ön tanımlı bir bileşen olarak gelir [34].

Python, diğer programlama dillerinden pek çok yönüyle ayrılır. Öncelikle nesneye yönelik bir dildir ve Python’ da her şey bir nesnedir. Nesne özellikli programlama dillerinden bahsederken aklımıza ilk gelmesi gereken dillerden biridir. Bu konuda Java’ nın bile ötesinde olduğunu söylemek yanlış olmaz. Tim fonksiyonlar, sabit ve değişkenler, modüller, sınıflar birer nesnedir. Aslında sınıf özelliği olan programlama dilleri de nesne tabanlı diller olarak bilinmektedir. Python bu dillerin özelliklerini barındırmakla birlikte nesnellik biraz daha ağır basmaktadır. Örneğin aşağıdaki örnekte gösterildiği gibi bir cümle sabiti bile bir nesnedir [35]:

```
>>> 'python'.upper()
'PYTHON'
```

Görünen üzere daha sabit tanımlanır tanımlanmaz bir nesne haline geliyor ve buna çeşitli özellikler bağlanıyor. Burada ‘python’ cümle sabiti bir nesnedir, bu nesneye upper() fonksiyonu uygulanıyor ve sabit içerisindeki harfler büyük harfe çevriliyor. Aşağıda bir değişkenle oluşturulmuş benzer bir örneğe de bakılırsa:

```
>>>a=199
>>>a._hex_()
'0xc7'
```

Bu örnekte ise a isimli bir tamsayı değişkeni oluşturuluyor ve 199 değerini bu değişkene atanıyor. Değişken tanımlanır tanımlanmaz bir nesne haline geliyor ve var olan özelliklerinden biri bu değişkene uygulanıyor. Sonuç olarak tamsayının on altılık (hexadecimal) gösterimi elde ediliyor.

Python ayrıca özgür bir dildir ve C ile yazılmış açık kaynak kodlarıyla herkesin kendi amaçları doğrultusunda kullanabileceği ve değişiklik yapabileceği açık bir programlama dilidir. Hatta yazardan izin almak ve yapılan değişiklikleri belirtmek kaydıyla farklı Python dağıtımları geliştirilebilir.

Python yorumlamalı diller sınıfına girse de hem derlemeli hem de yorumlamalı dillerin özelliklerini taşır. Yazılan kod önce Python yorumlayıcısı tarafından istek dışı “byte-code” denilen bir biçimde derlenir. Derleme çok hızlıdır ve ardından yorumlanır. Son zamanlarda popüler bir ortam haline gelen .net’ e özgü çeşitli Python derleyicileri yazılmıştır. Bu şekilde Python yorumlayıcısına gerek kalmaz ve derlenen programlar .net ortamında çalışabilir.

Python farklı platformlar arasında taşınabilir bir dildir. Hemen her işlemci ve işletim sisteminde çalışabilir. Yazılan kaynak kodlar ile byte-code’ ların işlemci-mimari veya işletim sistemi bağımlılığı yoktur. İşletim sistemine özgü herhangi bir modül veya fonksiyon kullanılmadığı sürece tüm kodlar farklı işletim sistemleri arasında transfer edilebilir.

Python ile birçok farklı uygulama geliştirmek mümkündür. Örneğin Unix üzerinde tüm C dinamik kütüphaneleri çağrılıp Python’ un rahatlığı altında kullanılabilir. Güçlü COM desteği sayesinde tüm Windows uygulamalarına erişilebilir ve üzerinde pek çok işlem yapılabilir. Visual C++ ile de yapılabilecek hemen her şey Python ile yapılabilir. Birçok programcı Python’ un kendi amaçları için geliştirildiğini iddia eder ve bu da Python’ un ne denli güçlü bir dil olduğunun göstergesidir [35].

Benzerlerine nazaran Python oldukça hızlı bir programlama dilidir. Örneğin bir alt seviyedeki Java ile gözle görülebilecek bir hız farkı vardır. Aslında Python programlarının Java’ ya göre daha yavaş çalışması beklenir ama Python ile program geliştirmek daha az zaman alır. Python programları Java programlarına göre üç beş kat daha kısadır. Bu fark

Python’ un gömülü yüksek seviye veri türlerine ve dinamik yapısına bağlıdır. Örneğin bir Python programcısı bir argümanı veya değişkeni tanımlarken hiç zaman harcamaz [36].

Ayrıca Python’ un hızlı olmasının nedenlerinden biri de modüllerinin ana sitemden bağımsız olmasıdır. Bu başlarda biraz yorucu olabilir; ancak istek dışı hiçbir modül hafızaya yüklenmeyecek ve sistemi yormayacaktır. Bu da tüm programın kontrol altında tutulmasını kolaylaştırırken, sistemin de hızını artıracaktır. Güçlü “Garbage-Collector”, yani çöp toplayıcısı sayesinde, artık işe yaramayacağı saptanan nesnelere olan bağlantılar bellekten silinecek, hafıza gereksiz yere kirlenmeyecek ve program ne kadar büyük olursa olsun sistem performansı etkilenmeyecektir.

Kullanımı ve yazılımı en kolay ve en açık dillerden biri olan Python sayesinde, yazılan koda yıllar sonra bakıldığında bile yapılanlar anlaşılabilir ve gerekli değişiklikler kolayca uygulanabilir. Oysaki bir çok programlama dilinde, yazılan kodlar karmaşık olduğundan kendi yazarı için dahi uzun bir aradan sonra kodlar anlaşılmaz hale gelmektedir. Yazımı kolay olarak bilinen Pascal ve Perl ile aynı ifadeleri yazıp Python ile karşılaştırılırsa [35]:

Pascal Kodu:

```
If a=3
```

```
Begin
```

```
    Writeln(‘Üç’);
```

```
End;
```

Perl Kodu:

```
İf (a==3){
```

```
    Print “Üç\n”;
```

```
}
```

Python Kodu:

```
İf a==3:
```

```
    Print “Üç”
```

Görülen üzere Python daha kolay anlaşılır bir dildir ve kodları da oldukça kısadır. Tabi ki bu kadar kısa bir kodda pek fark anlaşılmayacaktır; ama sayfalar dolusu kod yazılacaksa kodların kısalığının farkına varılacaktır.

Bu anlamda bu dili kullanan pek çok uzman eğitmen ve programcılar da bu dilin öğrenmesi kolay bir dil olduğunu belirtmişlerdir. Aşağıdaki örnekte göreceğimiz üzere, bir kelimedeki harfleri ekrana yazdırmak istediğimizde, bu kelime üzerinde bir döngü oluşturmak yeterli olacaktır.

For i in “elma”

Print i

Bu kod örneğinde görüldüğü gibi değişkenler için tip bildirimleriyle, işaretçilerle ya da ayrımlarla uğraşılmasına gerek yoktur. Kısaca kullanıcının zihnini bulandıracak hiçbir şey Python’ da bulunmaz. Değişkenlerin ne tipte olduğu daha tanımlanırken Python tarafından algılanır ve işlemler yürütülür.

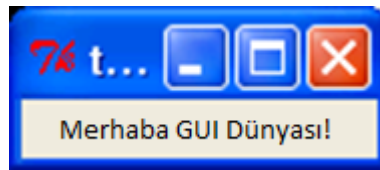
Sonuç olarak Python hemen her alanda kullanılabilir. Her programcı kendi amacına göre hazır pek çok kütüphane ve fonksiyon bulacaktır. Web üzerine uygulama geliştirmek isteyen bir kullanıcı da, dağıtık sistemler, sunucu/istemci şeklinde yazılımlar üzerine çalışan başka bir kullanıcı da Python dilini rahatlıkla kullanabilir.

Örneğin veritabanı programcılığıyla uğraşan biri, programlama dilinin zorluklarıyla uğraşmak istemiyor ve dikkatini sadece veritabanı ve sorgulamalara yoğunlaştırmak istiyorsa Python tam da bu işe uygun bir programlama dilidir. ODBC, SQL, gdbm, bdb ve xbase gibi hemen hemen her veri tabanı sistemi ile barışıktır.

Bütün bunlar dışında Python ile masaüstü yazılımlar oluşturmak da mümkündür. GUI (kullanıcı grafik arabirimi) programcılığı için birçok toolkit mevcuttur ve Python ile son derece uyumlu çalışırlar. Aşağıdaki kod bloğuna bakılırsa sonucu Şekil 5.1’ de gösterildiği üzere iki satırda GUI programları yazmak da Python ile mümkün olmaktadır [35].

```
From Tkinter import *
```

```
Label(None, {'text': “Merhaba GUI Dünyası!”, Pack: {}}).mainloop()
```



Şekil 5.1. Python ile anında GUI

Python dünya genelinde de birçok ticari firma tarafından da kullanılıyor. Bunların başında Linux’ a verdiği desteklerle bilinen Red Hat firması geliyor. Bu firma hemen her uygulamasını Python ile geliştiriyor. Microsoft .net ortamında Python’ un yaygınlaşması için büyük çaba sarfediyor. Dünyanın en hızlı arama motoru olarak görülen Google birçok işlemde Python’ u tercih etmiş ve son kullanıcıya web üzerinden verilen birçok hizmette Python kullanıyor. Yahoo da aynı şekilde bazı servislerinde bu dilin avantajlarından yararlanıyor. NASA, güvenliğin ön plana çıktığı gizli birçok uygulamasını Python ile

geliştiriyor. Bunların dışında ünlü bir web uygulama sunucusu olan Zope, Python ile geliştirilmiştir. Grail gibi Python ile yazılmış web tarayıcıları ise ilgi çeken uygulamalardandır [35].

5.3. Flask

Flask, Jinja2 ve Werkzeug' a dayalı Python için geliştirilmiş bir uygulama geliştirme çatısıdır (framework) ve BSD (Berkeley Software Distribution, Kaliforniya Üniversitesi, Berkeley' in kaynak kodu dağıtımı) lisanslıdır. Flask üç madde altında BSD lisanslıdır. Bu temelde şu anlama gelir: Telif hakkına uyulduğu, feragatname mevcut olduğu ve koşullar değiştirilmediği sürece onunla her istediğinizi yapabilirsiniz. Ayrıca yazılı izin olmadan yazılımın türevlerini düzenlemek için yazarların isimlerini kullanamazsınız. Flask başta Armin Ronacher olmak üzere bazı katkı sağlayanlar tarafından yazılmıştır [37].

2015 yılı itibariyle Flask' ın kararlı çalışan son sürümü 0.10.1' dir. Pinterest, LinkedIn ve yanı sıra Flask' ın kendisi için hazırlanmış topluluk web sayfası, Flask kullanılarak yapılmış örnek uygulamalardandır. Flask microframework olarak adlandırılır; çünkü geliştiriciyi belirli bir araç veya kütüphane kullanmaya zorlamaz. Hiçbir veritabanı soyutlama katmanı, form doğrulama veya ortak fonksiyonları sağlayan üçüncü parti kütüphanelerinin olduğu başka bileşenler yoktur. Ancak, bazı uygulama özellikleri için var olan ekleri destekler. Bu ekler nesne-ilişkili haritalar, form doğrulama, upload kullanımı, çeşitli açık kimlik doğrulama teknolojileri ve çeşitli ortak bir framwrok ile ilgili araçlar için vardır.

Genel olarak Flask' ın özellikleri aşağıdaki gibidir [38]:

- Geliştirme sunucusu ve debugger içerir.
- Birim test için entegre desteği vardır.
- Aktarım için kolay istekler.
- Jinja2 şablonlamayı kullanır.
- Güvenli çerezler için destek (istemci tarafı oturumları) sağlar.
- % 100 WSGI 1.0 uyumludur.
- Unicode tabanlıdır.
- Kapsamlı dokümantasyon vardır.
- Google App Engine uyumluluğu vardır.
- İstenilen özelliklerini geliştirmek için mevcut ekler.

Flask, Werkzeug ve Jinja2 olmak üzere iki harici kütüphaneye bağlıdır. Werkzeug WSGI için bir araç kitidir. WSGI, web uygulamaları ve çeşitli geliştirme ve dağıtım sunucuları arasında standart Python ara yüzüdür. Jinja2 şablonları oluşturur.

Bütün bunları bilgisayarımıza yüklemenin birden fazla yolu bulunmaktadır; fakat en bilineni virtualenv' dir. Bunun için Python 2.6 veya daha üst bir sürümüne ihtiyaç duyulacaktır [39].

Virtualenv, geliştirme sürecinde ve üretim makinelerine kabuk erişimi varsa kullanmak istenilecek bir şeydir. Eğer Python' un farklı sürümlerinde çalışılıyorsa veya en azından farklı kütüphaneleri ile çalışılıyorsa; virtualenv kullanmak gerekecektir. Virtualenv, her proje için Python' un çoklu yan yana yüklemelerini sağlar. Aslında Python' un ayrı kopyalarını yüklemez; ama farklı proje ortamlarını korumak için akıllı bir yol sağlar [39].

Eğer Mac Os veya Linux üzerinde çalışılıyorsa; kurulum için aşağıdaki iki kod da kullanılabilir:

```
$ sudo easy_install virtualenv
```

```
$ sudo pip install virtualenv
```

Bu kodlardan biri sisteme virtualenv' yi yükleyecektir. Eğer Ubuntu kullanılıyorsa aşağıdaki kod denenebilir:

```
$ sudo apt-get install python-virtualenv
```

Eğer Windows kullanılıyorsa ve easy_install komutu yüklü değilse, önce on yüklenmelidir.

Virtualenv bir kez kurulduktan sonra kendi ortamımızı oluşturabiliriz. Öncelikle aşağıdaki gibi bir proje klasörü ve venv klasörü oluşturmakla başlanabilir:

```
$ mkdir projem
```

```
$ cd projem
```

```
$ virtualenv venv
```

```
New python executable in venv/bin/python
```

```
Installing distribute.....done.
```

Bu işlemlerin ardından bir proje üzerinde çalışılmak istersenirse; sadece karşılık gelen ortam etkinleştirilmelidir. OS X ve Linux için:

```
$ . venv/bin/activate
```

Windows için ise:

```
$ venv\scripts\activate
```

Kendi virtualenv' mizde artık Flask' ı etkinleştirmek için aşağıdaki komut kullanılabilir.

```
$ sudo pip install Flask
```

Aşağıda “Merhaba Dünya” yazan basit bir web uygulama kodu bulunmaktadır:

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return "Merhaba Dünya!"
if __name__ == "__main__":
    app.run()
```

Bu kod merhaba.py veya buna benzer bir şekilde kaydedilmeli ve sonra Python yorumlayıcısı ile aşağıdaki gibi çalıştırılmalıdır.

\$ python hello.py

* Running on <http://127.0.0.1:5000/>

Kodlar kısaca açıklanırsa;

1. Öncelikle Flask sınıfı içeri aktarılıyor. Bu sınıfın bir örneği bizim WSGI uygulamamız olacaktır.
2. Sonra bu sınıfın bir örneğini oluşturuyoruz. İlk argüman, uygulama modülünün veya paketin adıdır. Eğer (bu örnekte olduğu gibi) tek bir modül kullanıyorsanız, __name__ kullanmalısınız.
3. Ardından fonksiyonumuzu başlatacak URL’ nin hangisi olduğunu Flask’ a anlatacak olan route() komutunu kullanıyoruz.
4. Fonksiyona bir isim verilmiştir ve bu aynı zamanda özel fonksiyonlar için URL’ ler oluştururken kullanılır ve kullanıcının tarayıcısında görüntülemek istediğimiz mesajı geri gönderir.
5. Sonuç olarak; uygulamamızla yerel sunucuyu çalıştırmak için run() fonksiyonunu kullanırız. “if __name__ == '__main__':” bloğu sunucunun sadece Python yorumlayıcısından direkt gelen kodları çalıştırması için gereklidir. Sunucuyu durdurmak için Control – C kısa yolu kullanılabilir.

Sunucu çalıştırıldığında, sunucuya ağdaki başka bir kaynaktan değil de sadece kendi bilgisayarınızdan erişebileceğiniz ile ilgili bir uyarı görüntülenir. Varsayılan değer budur; çünkü hata ayıklama modunda uygulamayı kullanan birisi bilgisayarda rastgele bir Python kodu çalıştırabilir. Eğer hata ayıklama kapalıysa veya ağdaki kullanıcılara güveniliyorsa; run() metodunda aşağıdaki gibi ufak bir değişiklikle sunucu ağda herkesin erişimine açılabilir:

```
app.run(host='0.0.0.0')
```

Bu kod, işletim sistemine ağdaki tüm IP' leri dinlemesini söyler.

Run() metodu, yerel bir sunucuyu başlatmak için iyidir; ama kodlardaki her değişiklikte yeniden başlatmak gereklidir. Eğer hata ayıklama etkinleştirilirse, sunucu kod değişikliklerinde kendini yeniden yükleyecektir ve işler ters gittiğinde kullanılmak üzere yardımcı bir hata ayıklama devrede olacaktır. Aşağıdaki kodla hata ayıklamayı etkinleştirmek mümkün olmaktadır [39]:

```
app.debug = True
```

```
app.run()
```

veya

```
app.run(debug=True)
```

6. WEB TABANLI SCADA SİSTEMİNİN GELİŞTİRİLMESİ

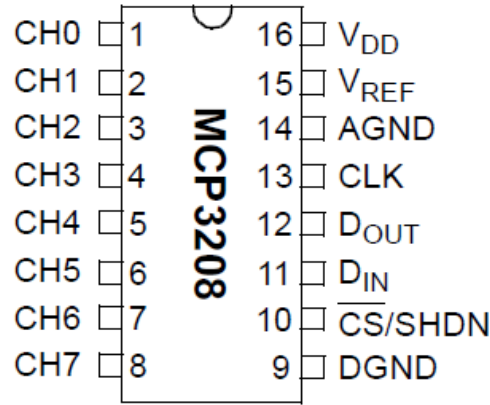
6.1. Sistemin Tanımlanması

Bu tez çalışmasında WEB üzerinde çalışabilen bir SCADA sistemi geliştirilmiştir. Geliştirilen SCADA sisteminin ARM işlemci tabanlı bir elektronik devre ile erişimi ve sistemin yönetilmesi sağlanmıştır. Sistemin kullanıcı tarafından kontrol edilebilmesi için WEB tabanlı bir yazılım hazırlanmış ve sunucu tarafında ise WEB yazılımına bağlı çalışan bir sunucu yazılımı geliştirilmiştir. SCADA sistemi tasarlanırken, bu tür sistemlerde olabilecek bileşenler üzerinde durulmuş ve örneklenmeye çalışılmıştır.

Öncelikle bu çalışmada çeşitli çevresel elemanların çalışmasını sağlamak adına ve bu olayı temsilen ledler kullanılmış, sisteme sunucu tarafından müdahale edilmesi gereken durumlar için analog girişler (buton, potansiyometre), potansiyometre ve benzer analog girişleri kullanabilmek amacıyla analog dijital dönüştürücü, yine sunucu tarafında sistemle ilgili bilgi alımı için lcd ekran, ayrıca bir sıcaklık sensörü ve tüm bu ve bunlara benzer elemanların bağlanması için Raspberry Pi' nin GPIO sayısı yetersiz gelebileceği için kullanılabilecek pin sayısını artırmak üzere bir entegre kullanılmıştır. Tüm sistemi uzaktan gözden geçirmek, verileri almak ve müdahale etmek amacıyla bir WEB yazılımı geliştirilmiştir.

6.2. Sistemin Elektronik Alt Yapısı

Raspberry Pi içerisinde bir analog dijital dönüştürücü bulunmamaktadır. Kullanılan potansiyometreyi bağlamak için SPI veri yolunu kullanan MCP3208 entegresi kullanıldı. Potansiyometrenin bir ucu Raspberry Pi' nin 3.3V, diğer ucu GND pinlerine ve ortanca uç ise Şekil 6.1' de görünen bu entegrenin ilk pinine bağlandı.



Name	Function
V _{DD}	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
D _{IN}	Serial Data In
D _{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input
V _{REF}	Reference Voltage Input

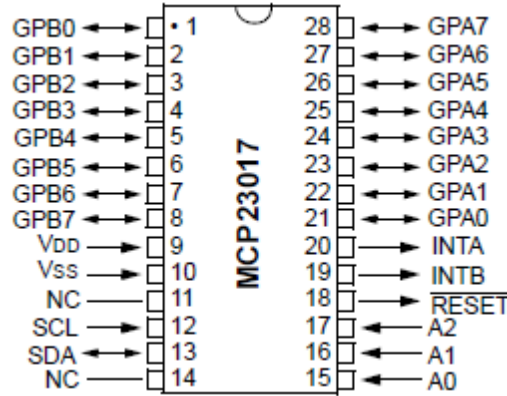
Şekil 6.1. MCP3208 pin yapısı

MCP entegrelerini Raspberry Pi' ye bağlamak ve üzerlerinde daha kolay işlemler yapmak üzere Adafruit Endüstri' nin hazırlamış olduğu kütüphaneler ve fonksiyonlar bulunmaktadır. Bu kısımda Analog dijital dönüştürücü MCP3208 entegresi üzerinden veri okumak için bu fonksiyonlardan birinden yararlanıldı. Entegrenin saat, veri girişi, veri çıkışı ve çip seçimi için kullanılan pinlerine 18,23,24 ve 25 numaralı GPIO' ların bağlantısı sağlandı.

MCP3208 12 bitlik bir dönüştürücüdür. Bu da 0 ile 4096 ($2^{12}=4096$) arasında bir veri okuyacağı anlamına geliyor. Fonksiyon içerisinde okunan değer 0 ile 100 aralığına dönüştürülüyor.

Raspberry Pi' nin sahip olduğu giriş ve çıkış amaçlı kullanılan pin sayısı bazı uygulamalarda yetersiz kalabileceği için pin sayısını genişletici entegreler kullanmak kaçınılmaz olmaktadır. Bu açıdan uygulamamıza çokça led, buton, lcd ekran gibi çevre elemanlar bağlamak istediğimizde bu entegrelerden birini kullanmak durumundayız. Şekil 6.2' de görünen yüksek hızlı I2C arayüzünü kullanan MCP23017 entegresi sayesinde 16 bitlik giriş çıkış pinleri elde etmiş olunur. Bu çalışmada da böyle bir entegreden

yararlanılmış, GPA pinlerinden ilk beşi lcd için kullanılmış, yedincisine bir buton bağlanmıştır. GPB pinlerinin tamamı da ledler için kullanılmıştır.

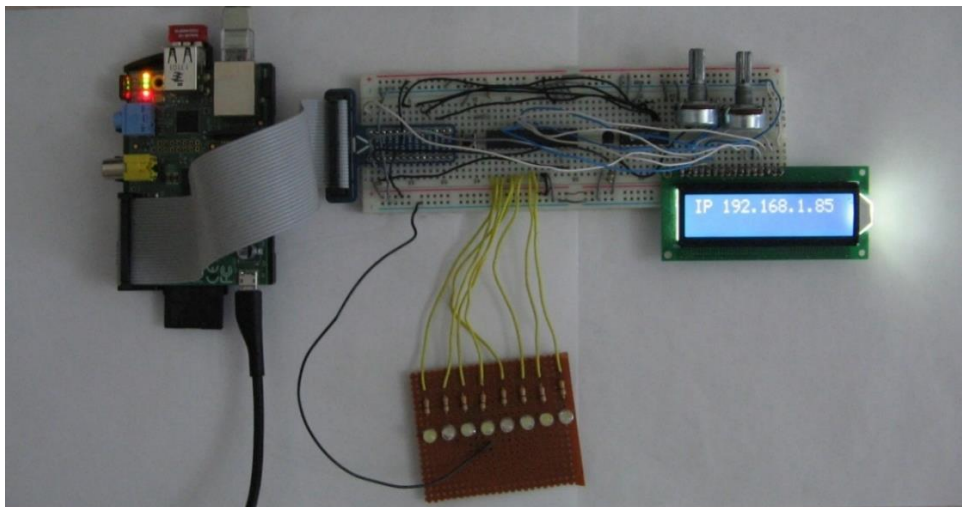


Şekil 6.2. MCP23017 pin yapısı

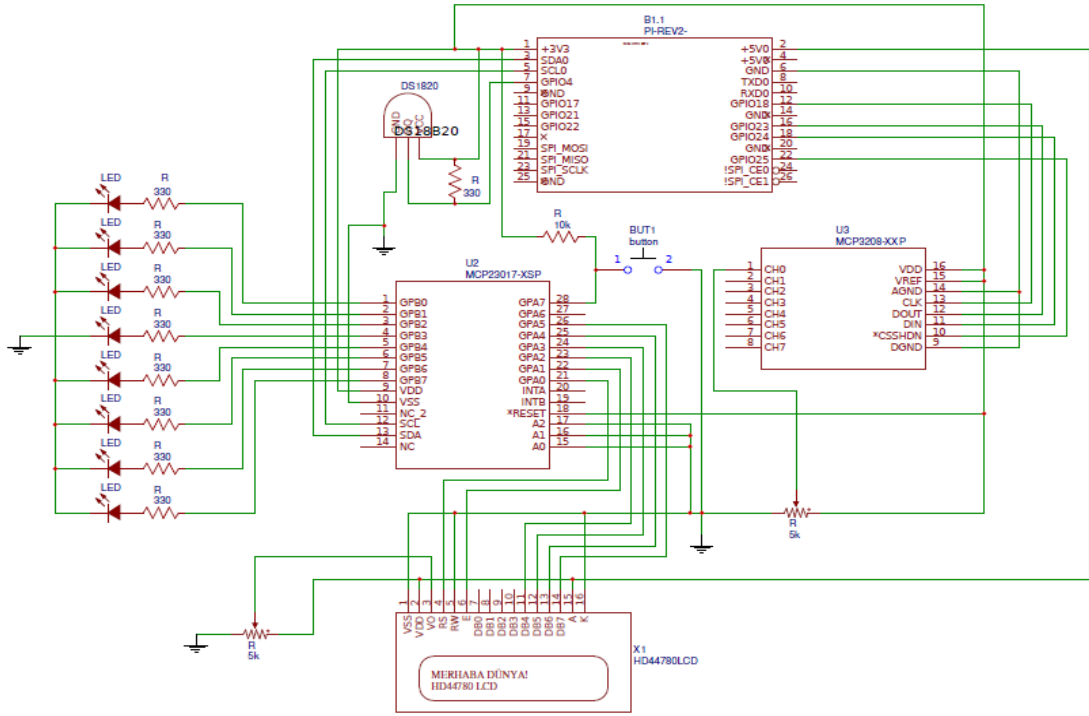
Yine bu kısımda bağlantıları sağladıktan ve gerekli kütüphaneleri yükledikten sonra bazı modüllerin kod içerisinde eklenmesi gerekiyor. Daha sonra lcd ekranın pinleri kod içerisinde tanımlanmış ve bu pin değerleri kodlarda ilgili sınıf içerisinde kullanılmıştır.

Bunlardan başka devremizde bir de sıcaklık sensörü kullanılmıştır. Kullanımı kolay olmasından dolayı dijital bir sensör olan DS18B20 sıcaklık ölçümü için kullanılmıştır. Bu tür sensörlerin transistörlere benzer şekilde üç bacağı bulunmaktadır. Bir bacağı 3.3V, diğeri GROUND ve ilk ve orta bacağı arasında konulan bir direnç sayesinde ölçüm yapılabilir.

Sonuç olarak çalışmamızın elektronik kısmı Şekil 6.3 ve Şekil 6.4’ te görüldüğü gibi düzenlenmiş ve sonrasında WEB arayüzü geliştirilmiştir.



Şekil 6.3. Sistemin elektronik altyapısı



Şekil 6.4. Kullanılan çevresel elemanların Raspberry Pi ile bağlantısı

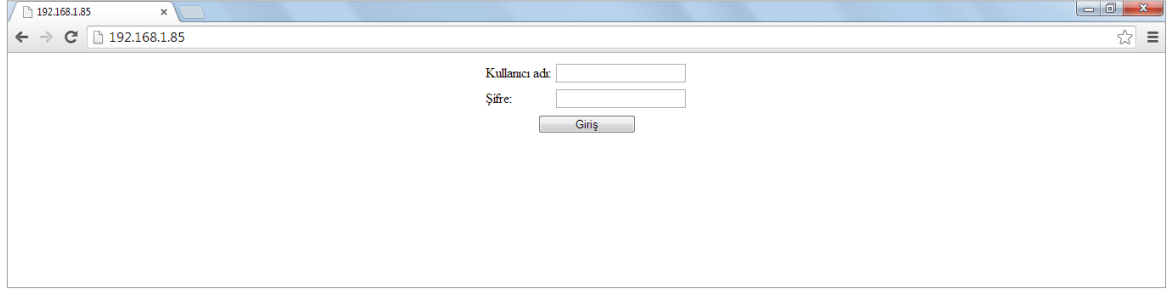
6.3. Yazılım Altyapısı

Sistemde Raspberry Pi ile elektronik devre arasında veri okuma, veri iletimi ve kontrol gibi tüm işlemlerde ve sistemin bir sunucu olarak kullanıcı ile etkileşimini sağlamak üzere Linux üzerinde oldukça uyumlu ve sorunsuz çalışan ve ayrıca tamamen ücretsiz olan Python programlama dili genel altyapı olarak kullanılmıştır. Sunucu yazılımı oluşturulurken Python için geliştirilmiş bir uygulama geliştirme çatısı (framework) olan Flask kullanılmıştır. Web arayüzü oluşturulurken web sayfalarını oluşturmak için kullanılan standart metin işaretleme dili olan HTML kullanılmış, görsel ve etkileşimli öğelerin hazırlanması için actionscript ve javascript dillerinden faydalanılmış ve sunucu yazılımı ile web arayüzündeki işlemlerin sayfa yenilenmesine gerek duymadan daha hızlı bir şekilde gerçekleşmesi için yine javascript dilinden yararlanılmıştır.

6.4. Web Arayüzü Tanıtımı

Sistem ağa bağlandığında ve başlatıldığında, aldığı ip adresi lcd ekran üzerine otomatik olarak yansıtılmaktadır. Daha sonra bu ip adresine herhangi bir web tarayıcısıyla

bağlanıldığında öncelikle sisteme giriş yapılabilmesi için bir sorgulama ekranı gelmektedir. Sistem yöneticisi tarafından oluşturulan ve değiştirilebilen kullanıcı adı ve şifresi ile oturum açılarak sisteme giriş yapılmaktadır.



Şekil 6.5. Web arayüzü giriş ekranı

Sisteme Şekil 6.5’ teki web arayüzü giriş ekranı üzerinden giriş yapıldıktan sonra Şekil 6.6’ daki gibi bir karşılama ekranı gelecektir. Bu kısımda sol taraftaki menüler ile ilgili bağlantıya erişmek mümkündür. Bu bağlantılar aracılığıyla sistemden veriler okunabilmektedir ve dilerse sisteme müdahale edilebilmektedir. Ayrıca sağ kısımda bulunan bağlantı ile istenirse sistemden çıkış yapılabilir.



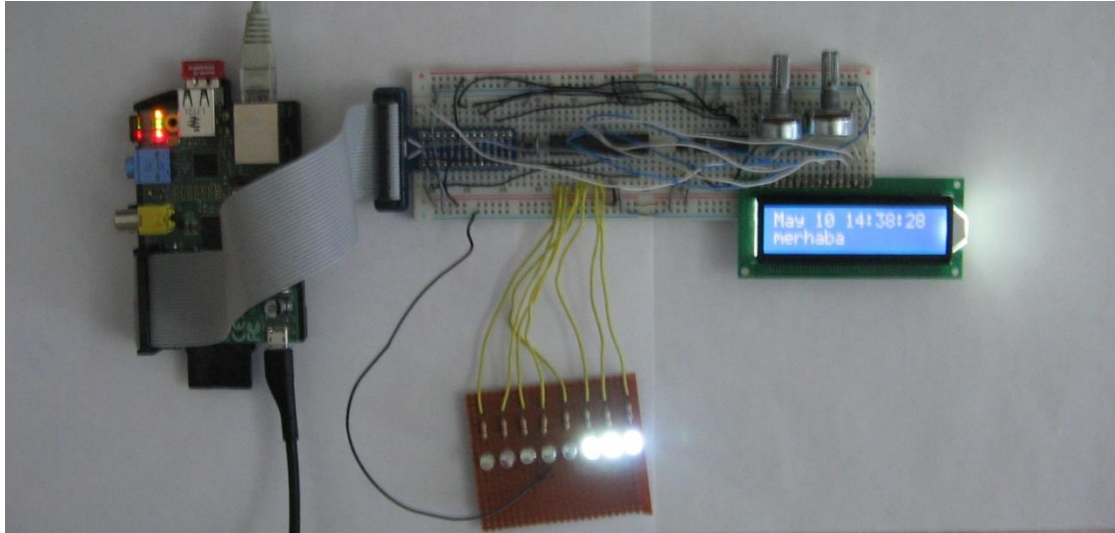
Şekil 6.6. Karşılama ekranı

Bilindiği üzere elektronik devrelerde cihazların açılıp kapatılması gibi işlemler ledler ile temsil edilmektedir. Şekil 6.7 ve Şekil 6.8’ de görüldüğü üzere led kontrolleri başlığıyla

ulařılan baęlantıda 8 adet temsili led ve mevcut durumları ile ilgili bilgi alabileceęimiz bir ekran karřımıza ıkmaktadır. Bu kısımda yeřil renk ledin yanıyor olduęunu, kırmızı renk ise sönük olduęunu göstermektedir. Ledlerin üzerine tıklanması ile istenilen led yakılabilir veya tersi řekilde sönürülebilir.



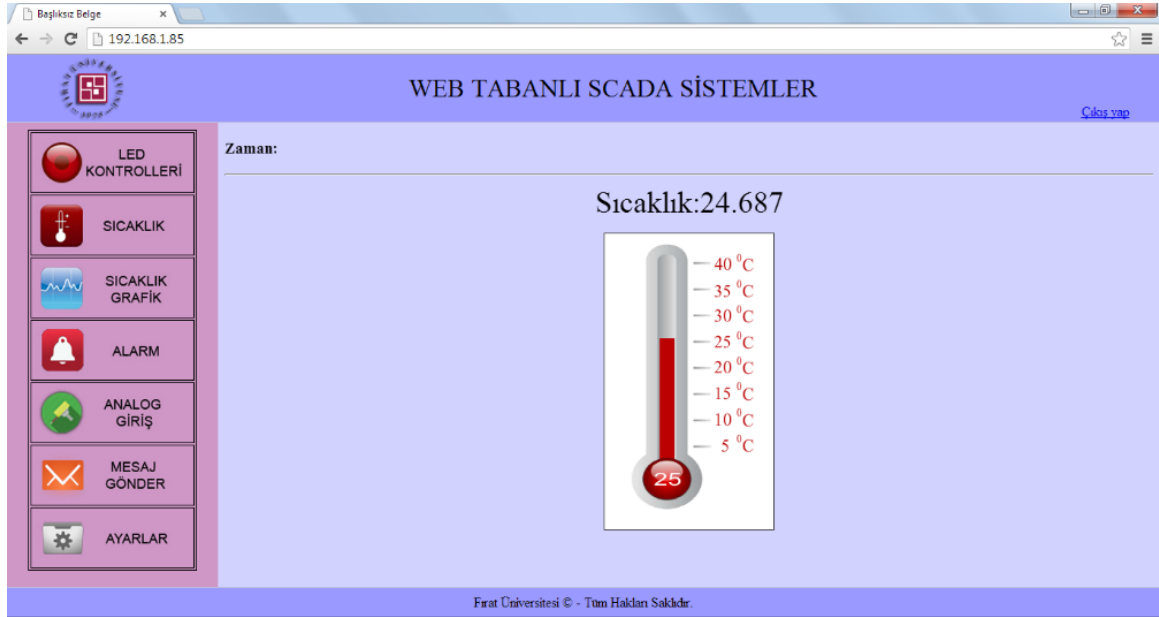
Őekil 6.7. Led kontrolleri ekranı



Őekil 6.8. Sistem üzerinde led kontrolleri

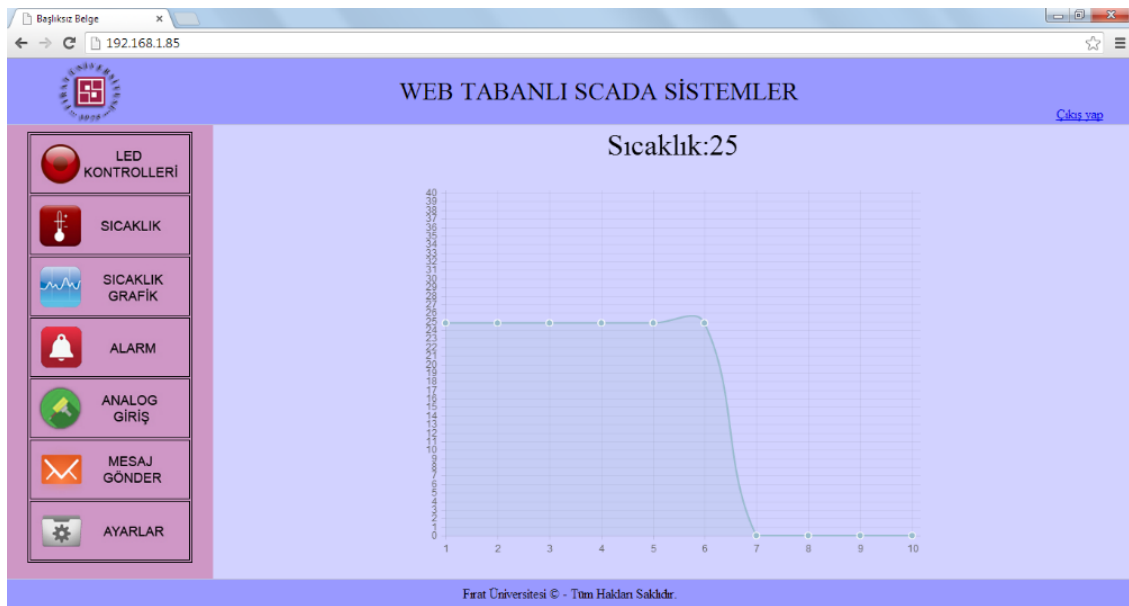
Sol kısımdaki menüde sıcaklık ile ilgili ulaşabilen iki baęlantı bulunmaktadır. Burada bulunan sıcaklık deęerleri, sistemin elektronik alt yapısında bulunan sıcaklık sensöründen okunan deęerlerdir. Sıcaklık deęeri anlık olarak okunur ve web yazılımını açan kullanıcı

ekranında anlık olarak görüntülenir. Bu kısımda ayrıca, Şekil 6.9’ da görüldüğü gibi sıcaklık sensöründen okunan değeri bir termometre üzerinde gösterilmiştir.



Şekil 6.9. Anlık sıcaklık değeri ekranı

Sıcaklık grafiği kısmında ise; bağlantı adından anlaşılacağı üzere sıcaklık ile ilgili bir grafik gösterilmektedir. Bu grafik, istenilen belirli zaman aralıklarında sıcaklık sensöründen okunan verileri göstermektedir. Bu sayede Şekil 6.10’ da görüldüğü üzere istenilen aralıkta sıcaklık değişimleri gözlenebilmektedir.



Şekil 6.10. Sıcaklık grafiği ekranı

Sonraki alarm menüsünden ulaşacağımız kısımda, SCADA sistemlerde çokça kullanılan alarm sistemi örneklenmeye çalışılmıştır. Açılan bağlantıda bulunan düğme aracılığıyla sistemde alarm çalması sağlanıyor ve yine aynı şekilde durdurulabiliyor. Bu durum Şekil 6.11 ve Şekil 6.12’ de gösterilmiştir.



Şekil 6.11. Alarm ekranı pasif



Şekil 6.12. Alarm ekranı aktif

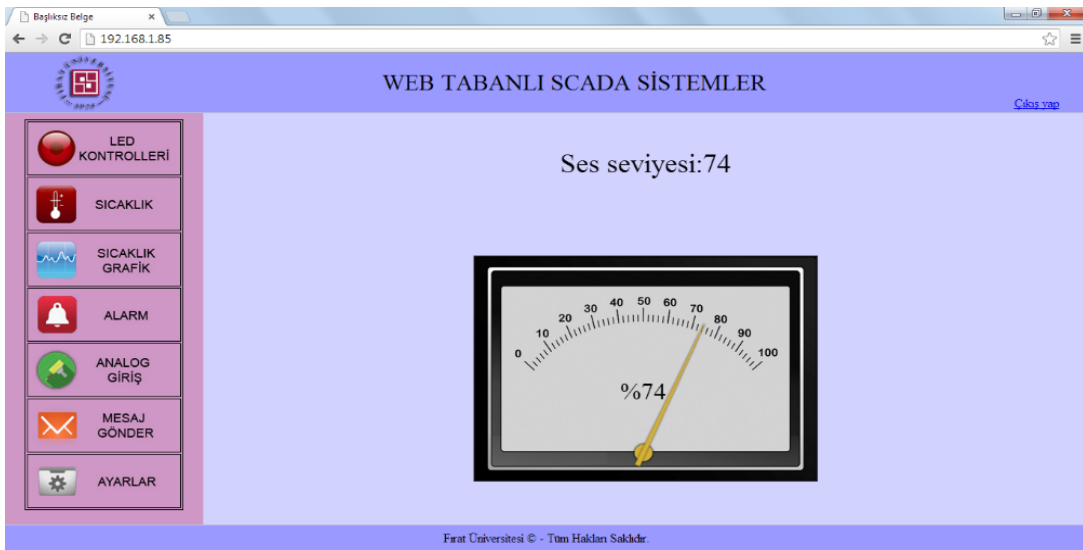
Alarm ekranında ayrıca alt kısımda sistem butonunun basılıp basılmadığını gösteren bir kısım bulunmaktadır. Alarm sisteminin sadece web yazılımı üzerinden değil de elektronik

devre kısmından da aktifleştirilmesi gerekebileceğinden bu buton sistemin elektronik devresi üzerinde bulunmaktadır. Butonun basılı olup olmadığı Şekil 6.13’ deki gibi web yazılımı üzerinden görüntülenebilmektedir.



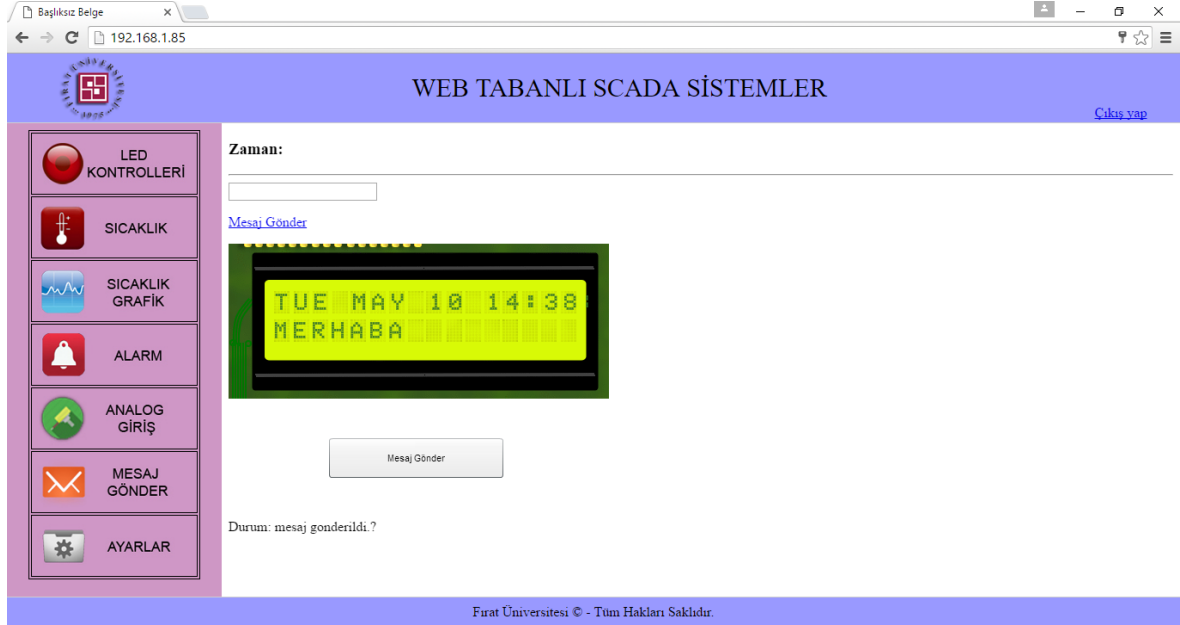
Şekil 6.13. Alarm ekranı – buton basıldı

SCADA sistemlerde analog girişler çokça kullanılmakta ve bu girişlerden alınan verilerle sisteme müdahale edilmesi gibi durumlarla çokça karşılaşılmaktadır. Bu kısımda da elektronik devre kısmında bulunan bir potansiyometreden okunan veriler, Şekil 6.14’ te görüldüğü üzere anlık olarak sıfır ile yüz arasında değişen değerler halinde ibre ve şekil üzerinde gösterilmektedir. Bu değerler sistemde örnek olarak ses seviyesini ayarlamak için kullanılmıştır.

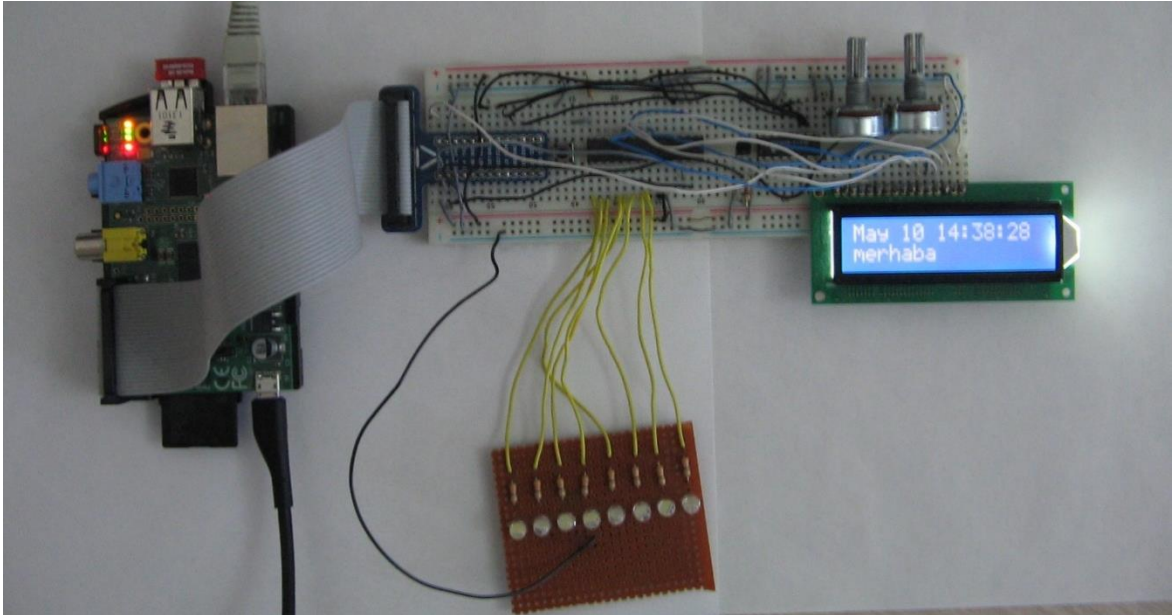


Şekil 6.14. Analog giriş ekranı

Web yazılımında, sisteme mesaj gönderilebilen bir kısım da bulunmaktadır. Gönderilen bu mesaj sunucu kısmında bulunan devre üzerindeki LCD ekrana yansıtılmaktadır. Mesaj girişi daha görsel olması açısından web yazılımında Şekil 6.15 ve Şekil 6.16’ daki gibi bir LCD ekran arayüzü üzerinden yapılmaktadır.



Şekil 6.15. Mesaj iletim ekranı



Şekil 6.16. Sistem üzerinde mesaj iletimi

Son olarak; Şekil 6.17 ve 6.18’ te görüldüğü gibi sisteme giriş yaparken bilinmesi gereken kullanıcı adı ve şifreyi değiştirebileceğimiz ayarlar kısmı bulunmaktadır. Daha önceden girilen veya sonradan değiştirilen bu değerler sistemin sunucu kısmında tutulmaktadır.



Şekil 6.17. Ayarlar ekranı



Şekil 6.18. Yeni kullanıcı adı ve şifre belirleme ekranı

7. SONUÇLAR

Günümüzde endüstriyel uygulamalarda, makinalar ile insanlar arasında haberleşmenin gerçekleştirilmesi için birçok farklı sistem kullanılmaktadır. Temel olarak kumanda mantığıyla çalışan ve özellikle karmaşık kablo hatları ve arıza verme sıklıkları fazla olan sistemler, yerini daha gelişmiş HMI ve SCADA sistemlerine bırakmaya başlamıştır. Ancak bu uygulamalar genel olarak PLC cihazlarına ihtiyaç duymaktadır. Endüstride kullanım alanı bir hayli fazla olan PLC'lerin işlem kapasiteleri ve hafızaları yetersiz kalabilmekte, bu yüzden de karmaşık işlemler karşısında bilgisayarlar ile birlikte kullanılabilirler. Özellikle PLC ve SCADA sistemlerinin birlikte kullanıldıkları uygulamalarda bilgisayar kullanılması, sistem maliyetini ve enerji tüketimini artırmaktadır. Ayrıca kullanılan bilgisayarların işletim sistemi, SCADA yazılımların lisans ücretleri gibi ek maliyetler söz konusudur ve mikroişlemci teknolojisinin gelişim süreciyle birlikte insanlar endüstriyel uygulamalarda daha kompakt tasarımlara yönelmektedir.

Bu çalışmaya başlarken, sistem maliyetini ve enerji sarfiyatını düşürecek ve boyutları daha kompakt hale getirecek çözümler üzerinde durulmuştur. Dolayısıyla bu hususlar göz önünde bulundurularak web tabanlı bir SCADA sistemi prototipi geliştirilmiştir. Bu aşamada web tabanlı sistemde sunucu görevini üstlenecek, üzerinde ethernet modülü barındıran, ucuz, enerji sarfiyatı düşük ve Linux işletim sistemi gibi açık kaynak kodlu ücretsiz yazılımlar kullanan bir geliştirme kartı seçilmiştir. Web tabanlı yazılım gerçekleştirilirken de açık kaynak kodlu programlama dilleri kullanılmıştır. Böylece sistem maliyetinin piyasadaki hali hazırda var olan SCADA sistemleri ile karşılaştırıldığında ciddi oranda düşük olduğu görülmüştür. SCADA sistem geliştirilirken bu tür sistemlerde olabilecek farklı modüllere yer verilmiş ve sistemin geliştirilmeye açık ve modüler bir yapıda olması sağlanmıştır.

Sonuç olarak; hazırlanan yazılımla uzaktan kolayca PC, tablet bilgisayar veya cep telefonları gibi iletişim araçlarıyla sistemin uzaktan izlenmesi, sensörler veya dijital-analog girdiler aracılığıyla sistem ile ilgili verilerin toplanması, bu verilerin raporlanması ve gerektiğinde sisteme müdahale edilmesi sağlanmıştır. Böylelikle özel bireysel amaçlara hizmet edebilecek veya özellikle küçük ve orta ölçekli işletmelerin kullanabileceği, kurulum maliyeti ve enerji sarfiyatı açısından avantaj sağlayan, boyutları küçük bir çözüm önerisi getirilmiştir.

Gelecek alıřmalar iin donanım tasarımı ile endüstriye uygun giriş ıkıřlar elde edilerek daha etkin bir sistem kurulması mümkündür. Ayrıca kompakt bir tasarım sayesinde mevcut PLC-SCADA sistemlerine alternatif bir ürün geliştirilmesi mümkün olacaktır. Özellikle ARM tabanlı donanım tasarımı yapılması gerekmektedir. Buna baėlı olarak standart endüstriyel panolarda kullanılması mümkün olan bir yapı elde edilebilir. Bu tez alıřmasında elde edilen sistem paralelinde ileride ticari ürünlerin piyasada satıřa sunulması öngörülmektedir.



KAYNAKLAR

- [1] web: <http://tr.wikipedia.org/wiki/SCADA>, 17/10/2012.
- [2] **Aydoğmuş, Z., Aydoğmuş, Ö.**, 2009. A Web-Based Remote Access Laboratory Using SCADA. IEEE Transaction on Education, Vol.52/1, pp.126-132.
- [3] **McDonald, J.D.**, ``Developing and Defining Basic SCADA System Concepts", Rural Electric Power Conference: B3/1 - B3/5, 1993 (IEEE).
- [4] **Berçin, N.**, “SCADA Sistemlerinin İncelenmesi ve O.G. Elektrik Dağıtım Tesislerinde Uygulanması”, İ.T.Ü Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Haziran–1997.
- [5] **Topak, H.**, “Bilgisayarlı Veri Toplama ve Veri Kontrol”, Mersin Üniversitesi.
- [6] **Wei-Fu Chang, Yu-Chi Wu and Chui-Wen Chiu**, “Development of a Web-Based Remote Load Supervision and Control System”, International Journal of Electrical Power&Energy Systems, Vol. 28, Issue 6, July 2006. p. 401–407.
- [7] **Gaushell Dennis J. and Block Wayne R.**, ``SCADA Communication Techniques and Standarts", IEEEComputer Application in Power, 6(3):45-50, 1993.
- [8] **Arifoğlu, B., Beşer, E.**, 2014. “Otomasyon Sistemleri Ders Notları”, Kocaeli Üniversitesi.
- [9] **Karaçor M., Keleş, K.**, “Otomasyon Sistemlerinin Bileşenleri”, Kocaeli Üniversitesi.
- [10] Anonim, Endüstriyel Otomasyon Teknolojileri, Plc Programlama, M.E.B. Yayınları, Ankara, 2007.
- [11] Anonim, Endüstriyel Otomasyon Teknolojileri, Scada Sistemlerine Giriş, M.E.B. Yayınları, Ankara, 2014.
- [12] Anonim, Industrial Automation, IDC Engineering Pocket Guide, IDC Technologies, 2008.
- [13] web: https://tr.wikipedia.org/wiki/OSI_modeli, 15/11/2015.
- [14] web: <https://tr.wikipedia.org/wiki/Fiberoptik>, 18/11/2015.
- [15] Anonim, Modbus Protocol Reference Guide, Industrial Automation Systems, Modicon, Inc., 1996.
- [16] web: <http://www.bihl-wiedemann.de/tr/uygulamalar/as-interface>, 02/12/2015.
- [17] web: <http://www.umitbulut.com/plc-haberlesmesi-ve-protokolleri>, 08/12/2015.

- [18] **Bayır, R., Soylu, E.**, 2014. “Endüstriyel İletişim Sistemleri Ders Notları”, Karabük Üniversitesi, 2014.
- [19] web: <https://tr.wikipedia.org/wiki/Ethernet>, 13/12/2015.
- [20] web: <http://mimoza.marmara.edu.tr/~hkaptan/tcp-ip.htm>, 13/12/2015.
- [21] web: <http://www.arm.com/products/processors/index.php>, 14/12/2015.
- [22] web: <https://tr.wikipedia.org/wiki/HTML>, 15/12/2015.
- [23] web: <https://tr.wikipedia.org/wiki/JavaScript>, 21/12/2015.
- [24] web: https://tr.wikipedia.org/wiki/Adobe_Flash, 23/12/2015.
- [25] web: <https://tr.wikipedia.org/wiki/ActionScript>, 23/12/2015.
- [26] web: https://tr.wikipedia.org/wiki/Raspberry_Pi, 23/12/2015.
- [27] web: http://elinux.org/RPi_Hardware, 26/12/2015.
- [28] web: <http://pi4j.com/pins/model-b-rev2.html>, 26/12/2015.
- [29] web: <https://www.raspberrypi.org/documentation/usage/gpio/>, 03/01/2016.
- [30] web: <https://learn.sparkfun.com/tutorials/i2c>, 03/01/2016.
- [31] **Hanabusa, R.**, “Comparing JTAG, SPI, and I2C”, Spansion Inc., April 13, 2007.
- [32] web: <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols>, 11/01/2016.
- [33] web: <http://quick2wire.com/articles/i2c-and-spi>, 11/01/2016.
- [34] web: [https://tr.wikipedia.org/wiki/Python_\(programlama_dili\)](https://tr.wikipedia.org/wiki/Python_(programlama_dili)), 12/01/2016.
- [35] **Başer, A.**, 2002. Python, Pusula Yayınları.
- [36] web: <https://www.python.org/doc/essays/comparisons>, 16/01/2016.
- [37] web: <http://flask.pocoo.org/docs/0.10/license>, 19/01/2016.
- [38] web: [https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework)), 19/01/2016.
- [39] web: <http://flask.pocoo.org/docs/0.10/installation>, 24/01/2016.
- [40] web: <https://en.wikipedia.org/wiki/HTML>, 24/01/2016.
- [41] web: <http://chimera.labs.oreilly.com/books/1234000000262/ch01.html>, 03/02/2016.

- [42] Anonim, Adobe Flash CS4 Kullanıcı El Kitabı, Adobe.
- [43] web: http://help.adobe.com/tr_TR/as3/learn/as3_learning.pdf, 05/02/2016.
- [44] web: <https://apc.io/products/rock>, 05/02/2016.
- [45] web: <http://beagleboard.org/BLACK>, 05/02/2016.
- [46] web: <http://cubieboard.org/2013/06/19/cubieboard2-is-here>, 05/02/2016.
- [47] web: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G137510300620, 05/02/2016.
- [48] web: <http://www.wandboard.org/details>, 07/02/2016.
- [49] web: <http://pandaboard.org/content/resources/references>, 07/02/2016.
- [50] **Hintermann, M.**, 2007. "Operating System Components for an Embedded Linux System", Technische Universitat Munchen.
- [51] web: https://en.wikipedia.org/wiki/ARM_architecture, 19/02/2016.
- [52] web: <http://www.otomasyondergisi.com.tr/arsiv/yazi/siemens-endustriyel-ethernet>, 24/02/2016.
- [53] web: http://enformatik.cumhuriyet.edu.tr/Html/Linux_Nedir.pdf, 24/02/2016.

ÖZGEÇMİŞ

Çağrı KARAKAŞ

Fırat Üniversitesi

Teknik Eğitim Fakültesi

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı

Bilgisayar Sistemleri Programı

Elazığ

cagrikarakas@gmail.com

- 1985** Elazığ'da doğdu.
- 2003-2007** Fırat Üniversitesi Teknik Eğitim Fakültesi Elektrik ve Bilgisayar Eğitimi Bölümü Bilgisayar Öğretmenliği Programı'ndan mezun oldu.
- 2007-** Milli Eğitim Bakanlığı'na bağlı bir ortaokulda bilişim teknolojileri öğretmeni olarak görev yapmaya devam etmektedir.