

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

KAN GAZI ANALİZ CİHAZLARINDAN AMELİYATHANELERE
BİLGİSAYAR KONTROLLÜ VERİ İLETİMİ

Ergün TOPALOĞLU

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2004

Her hakkı saklıdır.

Yrd. Doç. Dr. Ziya TELATAR danışmanlığında, Ergün TOPALOĞLU tarafından hazırlanan bu çalışma 29/01/2004 tarihinde aşağıdaki jüri tarafından Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. İnan GÜLER

Yrd. Doç. Dr. Ziya TELATAR

Yrd. Doç. Dr. Murat EFE

Yukarıdaki sonucu onaylarım

Prof. Dr. Metin OLGUN
Enstitü Müdürü

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vi
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ	3
3. MATERYAL ve YÖNTEM	5
3.1. Sistemin Donanım Yapısı.....	5
3.1.1. Ağ yapısı.....	7
3.1.2. Kablolama	11
3.2. İletişim Kontrol Protokolü / İnternet Protokolü (TCP/IP)	12
3.2.1. İletişim kontrol protokolü (TCP).....	13
3.2.2. İnternet protokolü (IP).....	14
3.3. Sistem Yazılımı	14
3.3.1. Mesaj formatı	14
3.3.2. Sistemin işleyiş adımları.....	18
3.3.3. Hata denetimi	20
4. ARAŞTIRMA BULGULARI	22
4.1. Sunucu Yazılımı	22
4.2. İstemci Bilgisayar Yazılımı.....	25
4.3. Hasta Kayıtları Menüsü	26
5.TARTIŞMA ve SONUÇ	28
KAYNAKLAR.....	30
EKLER.....	32
EK 1.....	33
EK 2.....	66
ÖZGEÇMİŞ	69

ÖZET

Yüksek Lisans Tezi

KAN GAZI ANALİZ CİHAZLARINDAN AMELİYATHANELERE BİLGİSAYAR KONTROLLÜ VERİ İLETİMİ

Ergün TOPALOĞLU

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman : Yrd. Doç. Dr. Ziya TELATAR

Bu tezde, kan gazı analiz cihazları ile ameliyathaneler arasında kurulan bilgisayar kontrollü veri iletim sistemi anlatılmaktadır. Klasik hastane uygulamalarında, ameliyat sırasında yaklaşık yirmi dakikalık periyotlar ile ameliyat edilen hastanın kan örneği incelenmek üzere bir personel tarafından laboratuara götürülür ve incelemenin ardından sonuçlar yazıcı çıktısı halinde tekrar ameliyatı yapan ekibe ulaştırılır. Bu sistem hem zaman, hem işgücü hem de sterilizasyon kaybına neden olmaktadır.

Geliştirilen yeni sistemde bir adet sunucu bilgisayar ve bir adet de istemci bilgisayar kullanılmıştır. Sunucu bilgisayar seri port aracılığıyla kan gazı cihazına bağlanmıştır. İstemci bilgisayar ameliyathaneye konulmuş ve ethernet kartları aracılığıyla sunucu bilgisayara bağlanmıştır. Geliştirilen yazılım ile kan gazı cihazından alınan sonuçlar sunucu bilgisayarca işlenmekte ve bir saniyeden kısa bir sürede istemci bilgisayara gönderilmektedir. Sunucu bilgisayardaki yazılımla, sonuçlar kaydedilip, yazdırılabilirken eski sonuçlar içinde arama yapılabilmektedir.

Yeni sistem ile işgücü kayıpları, zaman ve sterilizasyon problemleri en aza indirilmiştir. Ayrıca parametre değerlerinin referans değerlerinden yüksek, düşük veya şüpheli olması durumunda ilgili parametreler kırmızı, mavi veya yeşil renklerde ekranda gösterilmektedir. Eğer parametre istenen aralıktaysa rengi siyah olur. Böylece doktorların parametre değerlerindeki kritik durumları kolayca fark etmeleri sağlanmıştır.

2004, 69 sayfa.

ANAHTAR KELİMELER : Veri aktarım, laboratuvar, ameliyathane, kan gazı.

ABSTRACT

Master Thesis

COMPUTER CONTROLLED DATA TRANSMISSION FROM BLOOD GAS DEVICES TO THEATRE ROOMS

Ergün TOPALOĞLU

Ankara University
Graduate School of Natural And Applied Sciences
Department of Electronics Engineering

Supervisor : Asst. Prof. Dr. Ziya TELATAR

This thesis presents a new data transmission system established between the test laboratory and the theatre room. In the classical hospital system, blood samples of patient are taken every twenty minutes and sent to the clinical laboratory then the results are transported by a clinical personel during the operation. Such a procedure results in waste of time and causes some sterilization concerns. Physicians prefer to simultaneously check the parameters on screen during the operation. On the other hand, the time and sterilization condition are two important factors for an operation and recovery. The classical method could not respond for these factors in a desired way. Therefore, more efficient ways are required for communication between the laboratory and the theatre room.

In the developed system, one server computer and a number of client computers are used. The server computer is connected to blood gas device through a serial port. The client computer is installed in theatre room and connected to server computer through the ethernet cards. The results taken from the blood gas device are processed and sent to the client computer in less than one second by the software. The results can be saved, printed and records can also be searched through by the software of server. The loss of data, time and sterilization problems are solved by the proposed system. When the parameter values become higher, lower or suspicious than the reference threshold value, the colors of relevant parameters occur red, blue or green. If the parameter value is within the acceptable limits then the value is displayed in black. Thus, medical staff would see and decide the critical situations of parameter values easily.

2004, 69 pages.

Key Words: data transmission, clinical laboratory, theatre room, blood gas.

TEŐEKKÜR

Üniversite sanayi işbirliđi öneminin giderek arttığı günümüzde, yapılan ortak çalışmalar ile her iki tarafa da büyük yararlar sağlamaktadır. Gerek mevcut sistemlerin geliştirilmesi gerekse ihtiyaçlar doğrultusunda yeni sistemler tasarlanması ile kurumların verimliliđi artırılırken arařtırmacılara da yeni uygulama alanları doğmaktadır. Bu çalışmada da hastanelerde kullanılan kan gazı analiz cihazı ile ameliyathane arasında bilgisayar kontrollü veri haberleşme sistemi uygulaması incelenmiştir.

Çalışmamın her safhasında yakın ilgi ve önerileriyle beni yönlendiren danışman hocam Sayın Yrd. Doç. Dr. Ziya TELATAR (Ankara Üniversitesi Mühendislik Fakültesi)'a, sistemi arařtırma ve geliştirme olanađı sağlayan Sayın Prof. Dr. İnan GÜLER (Türkiye Yüksek İhtisas Hastanesi Tıbbi Aygıtlar Bakım Onarın Merkezi Genel Koordinatörü 2000-2003) 'e, yardımlarından dolayı Sayın Tolga ÇİT (TABOM Elektrik Elektronik Mühendisi) 'e teşekkür ederim.

Ergün TOPALOĐLU
Ankara, Ocak 2004

ŞEKİLLER DİZİNİ

Şekil 3.1.	Sistemin blok diyagramı.....	5
Şekil 3.2.	Yıldız topolojisi.....	8
Şekil 3.3.	Kullanılan yerel ağ sistemi	9
Şekil 3.4.	Ağ topolojisi (mimarisi)	10
Şekil 3.5.	Hub içindeki yineleyici bağlantı.....	10
Şekil 3.6.	Bilgisayar-hub bağlantı şeması.....	11
Şekil 3.7.	UTP kablonun sinyal bozulmasına karşı gösterdiği direnç	12
Şekil 3.8.	TCP/IP katmanları.....	13
Şekil 3.9.	Sistemin akış diyagramı	19
Şekil 3.10.	Hasta kaydı arama fonksiyonu akış diyagramı.....	20
Şekil 4.1.	Sunucu yazılımının başlangıç penceresi.....	22
Şekil 4.2.	Analiz sonuçlarının alınmasından sonra sunucu bilgisayar ekran görüntüsü.....	24
Şekil 4.3.	İstemci bilgisayar yazılımı ekran çıktısı.....	25
Şekil 4.4.	Hasta kayıtları arama menüsü	26
Şekil 4.5.	Veri tabanında Accession Number'ı 2 ile başlayan hastalar.....	27

ÇİZELGELER DİZİNİ

Çizelge 3.1.	DTE hatları ve bacak numaraları.....	6
Çizelge 3.2.	Kurgu (setup) fonksiyonundaki karakter formatları	7
Çizelge 3.3.	Kontrol karakterinin hesaplanması.....	21

1. GİRİŞ

Elektronik ve bilgisayar teknolojisinin ulaştığı büyük aşama sonucunda, bütün uygulama alanlarında ve genel olarak sinyal işleme, haberleşme ve kontrol konularında olduğu gibi özel olarak hastane içi uygulamalarda da yaygın olarak gelişmiş mikroişlemci kontrollü cihazlar, bilgisayarlar ve yazılımlar kullanılmaya başlanmıştır. Bilgisayarların ve programlama dillerinin gelişmesi, sayısal teknolojiye yeni bir boyut kazandırmış, birçok uygulamada, tıpta kullanılan elektronik cihazlarda dahil olmak üzere sayısal kontrol, veri iletim ve görüntüleme yöntemlerinde büyük ilerlemeler elde edilmiştir.

Sağlık sektöründe bilgi teknolojilerinin kullanılması ile hem hizmet veren hem de hizmet alan için etkinliğin ve verimliliğin artırılmasını hedefleyen, iş süreçlerini iyileştirerek klinik ve finansal uygulamalarda entegre çözümler sağlayan sistemler üretilmektedir. Bu amaçla hastane bilgi sistemleri (HIS), laboratuvar bilgi sistemleri (LIS), radyolojik görüntü arşivleme sistemleri (PACS), elektronik hasta kayıtları, teletıp uygulamaları, akıllı kart uygulamaları gibi bilgi teknolojisi altyapı hizmetleri sunulmaktadır.

Günümüz imkanları hem donanım hem de yazılım alanında yeni nesil cihazlara üstün özellikler sağlamaktadır. Donanım ile elde edilen kabiliyetler yazılımla güçlendirilmektedir. Sistemler, genellikle insanların o günkü ihtiyaçlarını tam ve zamanında karşılamak üzere tasarlanmaktadır. Ancak zamanla ihtiyaçlar değişebilmekte ve bu durumda ekonomik ömrünü henüz doldurmamış cihazların yerine yenileri alınmakta veya varolan cihazlara sonradan çeşitli opsiyonlar eklenerek, ihtiyacı karşılayabilir duruma getirilmektedir. Yeni bir cihaz almaktansa varolan cihaza, ihtiyacı karşılayabilecek şekilde bir yetenek kazandırılması çok daha ekonomik olmaktadır.

Bu çalışmada temel amaç, bir kan gazı analiz cihazı ile ameliyathane arasına bilgisayar kontrollü haberleşme sistemi kurmaktır. Hastanelerde ameliyat sırasında yaklaşık yirmi dakikada bir, hastaya ait kan örneği alınmakta ve incelenmek üzere laboratuara gönderilmektedir. Bir ameliyathane personelinin getirdiği kan örneği kan gazı analiz cihazında

incelenmekte ve sonuç yazıcı çıktısı halinde tekrar ameliyathane personeline verilmektedir. Çıktının ameliyathanedeki doktorlara ulaşmasının ardından, kan gazı parametrelerindeki deęişimlere göre ameliyatın gidişatı belirlenmektedir. Klasik olarak bu şekilde gerçekleştirilen bilgiye ulaşma yöntemi ile işgücü kaybı, zaman ve sterilizasyon problemleri ortaya çıkmaktadır. Tasarlanan sistemle bu sorunların ortadan kaldırılması hedeflenmektedir.

Geliştirilen sistem, kan gazı analiz cihazına baęlı bir sunucu bilgisayar ile sunucuya baęlı istemci bilgisayar ve de bu bilgisayarlar üzerinde koşan yazılımlardan oluşmaktadır. Sistem, kurulan bir yerel aę üzerinde haberleşmektedir. Yıldız topolojisi prensibine göre oluşturulan aęda, bilgisayarlar arasındaki trafięi düzenlemek üzere sekiz portu olan bir yıldız göbek (hub) kullanılmıştır. Bu hub üzerinden, aęa toplam yedi adet istemci bilgisayar bağlanabilir durumdadır. Sunucu bilgisayarımız, kan gazı analiz cihazına seri portu (RS232) aracılığı ile bağlanmıştır. Sunucu bilgisayar, analiz sonuçlarını ham veriler halinde alır, düzenler, işler ve istenildiğinde istemci bilgisayara gönderir, yazdırır, veritabanına kaydeder, eski kayıtları araştırır. Sunucu bilgisayarın verileri alıp işleyip ameliyathanedeki istemci bilgisayara göndermesi bir saniyeden daha az sürmektedir. Kan gazı analiz cihazından sunucu bilgisayara alınan parametrelerin referans deęerleri aralığının üstünde veya altında olması durumunda parametreler ve deęerleri sırasıyla kırmızı ve mavi renklerde ekrana gelmektedir. İncelenen deęerin şüpheli olması halinde ekrandaki rengi sarı olmaktadır. Böylece doktorların operasyon sırasında, ekrandaki parametrelerdeki kritik durumları kolayca fark etmeleri de sağlanmıştır.

2. KAYNAK ÖZETLERİ

Sayısal iletişimin hızla gelişmesiyle birlikte tıp alanında teşhis ve tedavide kullanılan cihazlar, hastane ve laboratuvar otomasyon sistemleri ile birlikte teletıp gibi pek çok uygulamada önemli mesafe kat etmiştir.

Son zamanlarda, evde veya iş yerinde hastanın fizyolojik sinyallerinin uzaktan dinlenmesi ile ilgili birçok çalışma yapılmıştır. Elektronik bilgi ve haberleşme teknolojilerinin bir araç olarak kullanıldığı, internet üzerinden hastanın fizyolojik sinyallerinin uzaktan dinlenmesinin birçok tıbbi, ekonomik ve sosyal kazançları vardır (Hardalaç 2002).

Karar vermeye yardım, uzaktan algılama ve hastaların uzaktan gerçek zamanlı değerlendirilmesi gibi uygulamalara sahip teletıp, donanım ve yazılım alanındaki gelişmelerle birlikte artan iletim hızları sayesinde giderek daha çok kullanım imkanına sahip olmaktadır. EKG, X-ray, hasta kayıtları gibi verilerin uzakta bulunan bir uzmana gönderilmesi, sürekli eğitim veya telekonferans amacı ile toplantıların aktarılması genellikle internet üzerinden yapılmaktadır (Akan 2001). Ancak son zamanlarda saniyede 2 megabit lik veri aktarma hızına sahip Eutelsat gibi uluslar arası uydu kuruluşları da teletıp projeleri için sağladıkları alt yapı hizmetlerini tanıtmaya başlamıştır (Turkinternet.com 2002).

Diğer taraftan son yıllarda, hastanelerde tanı amaçlı görüntüleme sistemlerinden elde edilen görüntülerin, film ve kağıt ortamındaki analog görüntüler yerine sayısal ortamda monitörlerden izlenmesi, bilgisayar belleklerinde (HDD, CD, DVD vb.) depolanması ve gerektiğinde bunların tekrar çağrılarak monitörlerden izlenmesi amacıyla PACS (Picture Archiving and Communications System) sistemi geliştirilmiştir. Bu sistem farklı ortamlarda, aynı veya farklı zamanlarda kullanım kolaylığı sağlamaktadır. HIS (Hospital Information System), LIS (Laboratory Information System), RIS (Radiology Information System), ses tanıma ve PACS sistemlerinin entegre kullanımı, kağıtsız, filmsiz, zaman ve bilgi kaybı olmayan bir hastane demektir (Saka 2000).

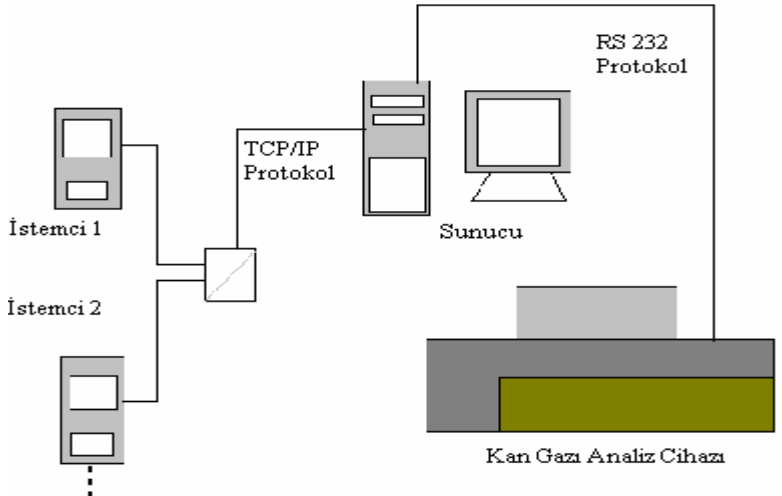
Klinik tedavi uzmanlarının kritik kararları doğru bir şekilde alması, geliştirilen yeni araç ve teknoloji ile daha kolay hale gelmektedir. Teknoloji yalnız başına hayat kurtaramaz iken teknolojinin yardımıyla insanlar hayat kurtarabilir hale gelmektedir. Yoğun bakım ünitelerinde yatan ve ameliyat edilen hastaların solunumlarının düzenlenmesi, yapılan ameliyatın gidişatının belirlenmesi açısından oldukça önemi bulunan kan gazı analiz cihazlarının acil durumlarda da kullanabilmesi için taşınabilir yapıdaki modelleri de imal edilmeye başlanmıştır (Belden *at al.* 1993).

Yoğun bakım ortamlarında, hastaya ait birçok karışık tıbbi verinin hızlı bir şekilde işlenmesi oldukça önemlidir. Bu verilerin gerçek zamanlı olarak görüntülenmesinin yanı sıra son yıllarda verilere dayanarak yorumlama yapan sistemler üzerinde de çalışılmaktadır. Akıllı klinik bilgi yönetimi sistemi (Clinical Information Management System-CIMS) olarak adlandırılan bu tip sistemler klinik laboratuarlarda elde edilen verilerin etkili kullanımına yardımcı olmaktadır. Örnek olarak CIMS, bir kan gazı analiz sonucundaki parametre değerlerine göre asit-baz dengesi ve elektrolit verileri hakkında yorumda bulunabilmektedir. Böylece klinik uzmanlarının giderek artan sayıdaki karmaşık verileri kullanarak hastaya daha doğru tanı koymaları mümkün olmaktadır (Kalogeropoulos *at al.* 1995).

3. MATERYAL VE YÖNTEM

3.1. Sistemin Donanım Yapısı

Bu çalışmada hastane ortamında küçük ölçekli bir yerel alan ağı (LAN) tasarlanmış ve ağ üzerinde çalışacak sunucu bilgisayar, istemci bilgisayar ile kan gazı analiz cihazı kullanılmıştır. Sunucu bilgisayarla kan gazı analiz cihazı seri portları aracılığı ile iletişim yapmaktadır. Sunucu bilgisayarla istemci bilgisayar, kurulan ağ üzerinden ethernet kartları ve TCP/IP protokolleri vasıtasıyla haberleşmektedir. Gerçekleştirilen yazılım hem sunucu bilgisayar hem de istemci bilgisayar üzerinde windows tabanlı tüm işletim sistemlerinde çalışabilmektedir. Sunucu bilgisayar donanım olarak intel Pentium IV 2.0 GHz ana işlemci, 256 Mb DDRam, 40 GB 7200 rpm HDD, 32 Mb Geforce 2 ekran kartı, 17'' monitör, ethernet kartının tümleşik olduğu bir anakart ve Windows XP Professional Edition işletim sistemine sahiptir. İstemci bilgisayar ise intel celeron 333 ana işlemci, 64 Mb SDRam, 3.2 Gb HDD, 8 Mb ekran kartı, 15'' monitör, anakart, ethernet kartı ve Windows ME işletim sistemine sahiptir. Şekil 3.1.'de sistemin blok diyagramı verilmiştir.



Şekil 3.1. Sistemin blok diyagramı

Temel iletişim mimarisi noktadan noktaya olan bu sistemde kan gazı cihazı ile sunucu bilgisayar RS 232 portları üzerinden haberleşmektedir. Cihazlar arasındaki donanımsal hat, asenkron seri iletişim arayüzünü kullanmaktadır. İletişim, kullanıcının seçeceği 19200, 9600, 4800, 2400, 1200 baud rate'lerden herhangi biriyle yapılabilmektedir. Bu çalışmada, cihazlar arasındaki iletişim için veri iletim hızı 9600 baud olarak alınmıştır. Kan gazı cihazının arayüz portları (seri port 1,2 ve 3) 9 bacaklı, D tipi, erkek konnektörlerden oluşmuştur. Sistem, DTE (Data Terminal Equipment) olarak düşünülmüştür. Çizelge 3.1.'de kullanılan DTE hatları verilmiştir.

Çizelge 3.1. DTE hatları ve bacak numaraları

DTE Hatları (sinyal isimleri)	Bacak Numaraları (pin)
Alındı mesajı (Data Carrier Detect)	1 (kullanılmadı)
Alınan veri (Received Data)	2
Gönderilen veri (Transmitted Data)	3
Veri uçbirimi hazır (Data Terminal Ready)	4 (kullanılmadı)
Sinyal şasesi (Signal Ground)	5
Veri (iletişim) aygıtı hazır (Data Set Ready)	6 (kullanılmadı)
Gönderme önerisi (Request to Send)	7 (kullanılmadı)
Gönderi için temizlik (Clear to Send)	8 (kullanılmadı)
Zil göstergesi (Ring Indicator)	9 (kullanılmadı)

Sayısal verilerin alınması ve iletimi RS 232 ara bağlayıcısının 2 ve 3 numaralı bacakları kullanılarak gerçekleştirilmektedir. Asenkron RS 232 haberleşmesi yapılmaktadır. Çizelge 3.2.'de, alıcının gelen veri paketlerini düzgün bir şekilde alabilmesi için iletişimde kullanılan başlama, durma ve eşlik bitleri gösterilmektedir.

Çizelge 3.2. Kurgu (setup) fonksiyonundaki karakter formatları

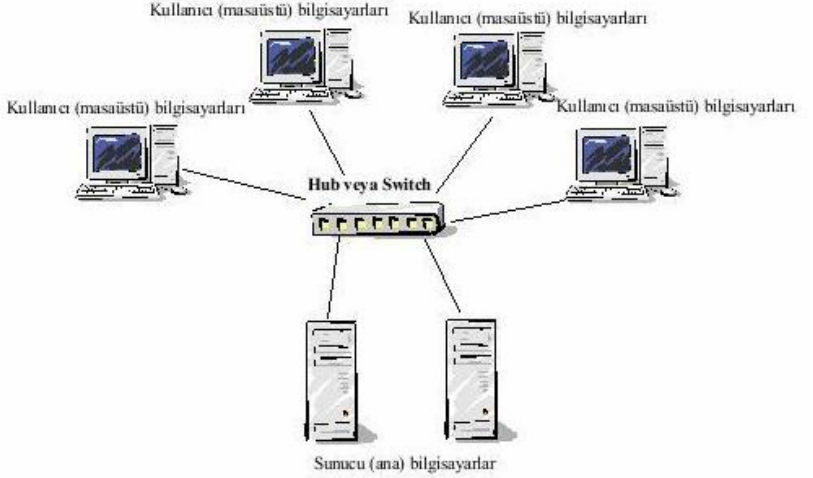
Seçenekler	Seçim
Veri bitleri	7 bit, 8 bit
Eşlik	tek, çift, yok
Durma bitleri	1, 2

Kan gazı cihazı, verilen kan örneğini inceledikten sonra sonuçları rulo kağıt yazıcısına (roll printer), paralel porta bağlı yazıcıya veya LIS'e gönderebilmektedir. Tasarlanan sistemdeki sunucu bilgisayar, LIS sunucusu olarak da isimlendirilebilir. İncelenen kan örneği sonucu, belli bir formatta sunucuya iletilmektedir. Sistemin tasarımı aşamasında analiz sonuçlarının sunucu tarafından alınabilmesi için Delphi programlama dilinin dört farklı seri port okuma bileşeni üzerinde çalışıldı. Çeşitli uygulamalar için geliştirilmiş comport, nrComm ve zComm isimli bu bileşenler ile verilerin sunucuya tam ve güvenilir bir şekilde alınması sağlanamamıştır. Ancak Cport isimli bileşen verilerin alınması için gerekli ihtiyaçları eksiksiz olarak karşılamıştır. Delphi tasarım ekranında, Cport bileşeninin nesne kontrol (object inspector) penceresinde veri bitleri, eşlik ve durma bitleri kan gazı cihazı LIS parametrelerine göre ayarlandı ve verilerin memo1 isimli hafıza ünitesine girişleri sağlandı (Yağimli vd. 2000). Daha sonra programın ilgili adımları takip edilerek bu veriler işlendi. Sunucu, alıp işlediği verileri, 100 Base Tx bağlantı kablosuyla bağlı bulunduğu istemci bilgisayara TCP/IP protokollerini kullanarak iletir.

3.1.1. Ağ yapısı

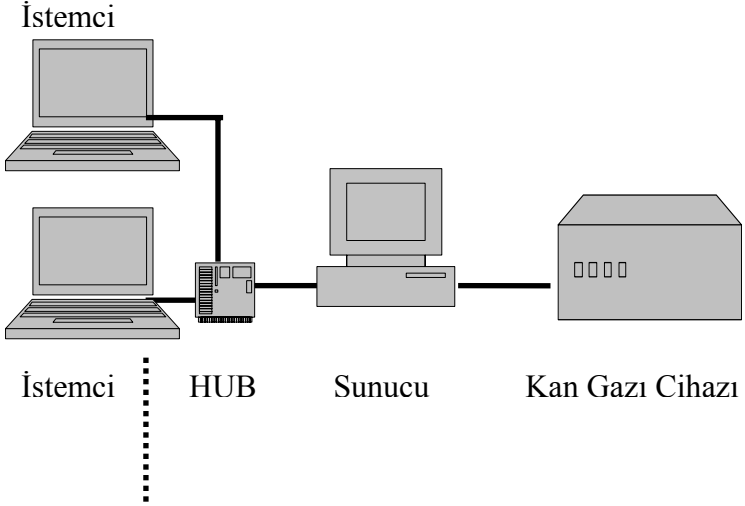
Haberleşme ağları, kaynakların paylaşımını kolaylaştırmak üzere tasarlanmıştır, ancak haberleşme harcamalarını düşürmek, akışı arttırmak ve servislerin gecikmesinin azaltılması da tasarım parametreleridir. Bu nedenle ağın topolojisi göz önüne alınması gereken önemli bir parametredir. Çeşitli ağ topolojileri vardır. Bu çalışmada yıldız topolojisi kullanılmıştır.

Yıldız topolojisi birçok ağda kullanılan bir topolojidir. Şekil 3.2.'de gösterildiği gibi her istasyon bire-bir bağlantı ile merkez siteye bağlanmıştır. Merkez site (hub veya anahtar) istasyonlar arası trafiği düzenleme yeteneğine sahiptir. Şekil 3.2.'de de görüldüğü gibi merkez siteye birden fazla sunucu da bağlanabilir. Örnek olarak sunuculardan birisi laboratuvar haberleşme sisteminin sunucusu iken diğeri, ağa internet hizmetini sağlayan bir sunucu olabilir.



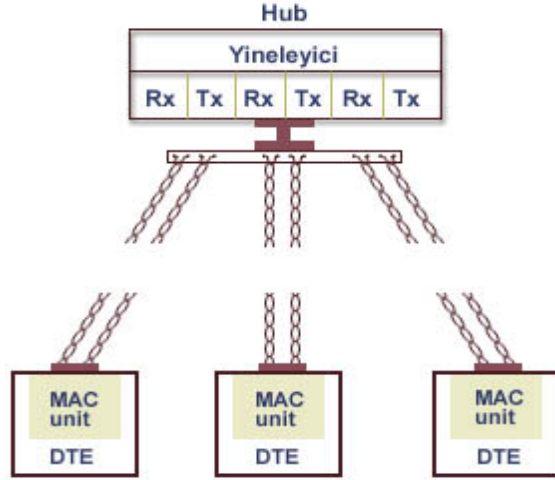
Şekil 3.2. Yıldız topolojisi

Bu çalışmada bir adet hub kullanılarak sistemin haberleşeceği yerel ağ oluşturulmuştur. Sistemde kullanılan yerel ağ bağlantısı şekil 3.3.'de gösterilmiştir. Kullanılan ethernet kablolama sistemi, 10BaseT yıldız topolojisidir. Ağdaki maksimum iletişim hızı 10 Mbit'tir. İstemci ve ana bilgisayarın haberleşmesi 'toplanma merkezi' manasına da gelen hub vasıtasıyla sağlanmaktadır. Ağa eklenecek diğer istemci bilgisayarlar da bu hub üzerinden bağlantı verilebilir şekildedir. Sistem, kurulacağı yerdeki mevcut ağa da bağlanabilir özelliindedir.



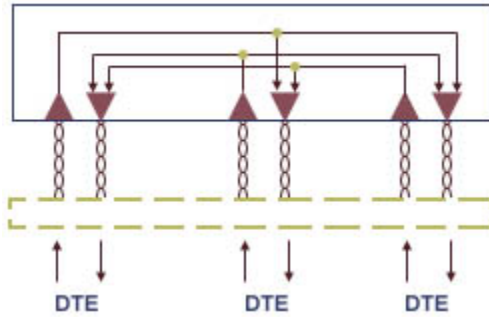
Şekil 3.3. Kullanılan yerel ağ sistemi

Hub en yalın ağ cihazıdır. Kendisine bağlı olan sistemlere paylaşılan bir yol sunar. Hub'a bağlı tüm sistemler aynı yolu kullandığı için, aktarım yapmak isteyen birçok bilgisayar olsa da aynı anda tek bir iletim yapılabilir, diğerleri yolun boş olmasını beklemelidirler. Hub'lar 8, 12, 16 ve 24 portlu olarak üretilirler. Bu portlardan bir veya ikisi 100 Mbps gibi yüksek hızlı iken, diğerleri genellikle 10 Mbps hızında olur. Yüksek hızlı portlar ya omurga bağlantısında ya da sunucu bilgisayar bağlantısında kullanılır. Oluşturulan sistemde ağ kartı olarak 10 Mbps hızlarındaki ethernet kartları kullanılmıştır. Ağ kartları, bağlanacakları ağ cihazlarının (Hub) portları ile aynı teknolojiye ve hızda olmalıdırlar.

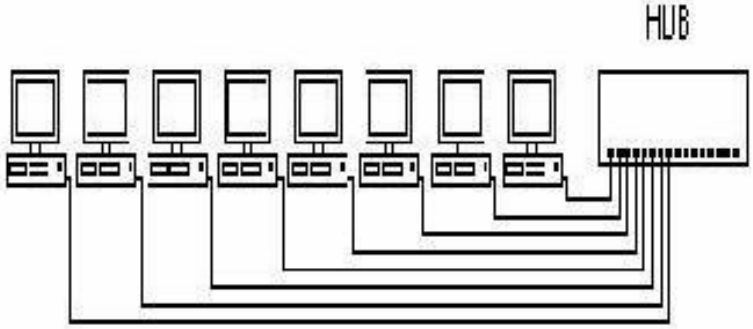


Şekil 3.4. Ağ topolojisi (mimarisi)

Şekil 3.4.'de verilen ağ mimarisinde görülen MAC (DTE) üniteleri, sistemde kullanılan ağ kartları yani ethernet kartlarını sembolize etmektedir. İki adet dolanmış (burgulu) kablo ethernet kartlarını hub'a bağlamaktadır. Kablolardan biri sinyal göndermek için kullanılıyor ise diğeri almak için kullanılır. Çarpışmaların fark edilmesine imkan tanımak için hub'ın içerisindeki tekrarlayıcı devreler bir giriş çifti ve diğere tüm çıkış çiftleri üzerinden alınmış sinyali yeniden gönderir. Bu durum şekil 3.5.'de verilmiştir.



Şekil 3.5. Hub içindeki yineleyici bağlantı

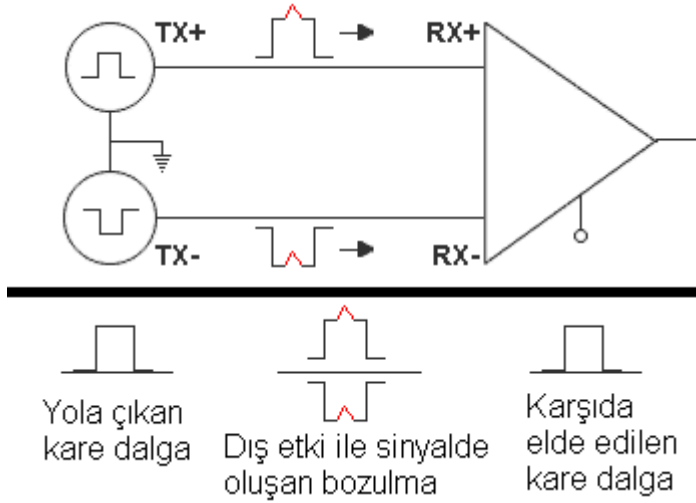


Şekil 3.6. Bilgisayar-hub bağlantı şeması

Şekil 3.6.'da gösterilen sistem için kurulmuş yerel ağa, sunucu ile birlikte toplam sekiz adet bilgisayar bağlanabilir. Sisteme daha fazla bilgisayar bağlamak için çapraz kablolama denen yöntemle ikinci bir hub daha bağlanıp onun üzerinden yeni bilgisayarlarla da bağlantı sağlanabilir. Ancak ağı büyötmek için ikinci bir hub yerine sisteme bir anahtar (switch) eklenmesi daha uygun olacaktır. Anahtara gelen bir sinyal tüm bilgisayarlar yerine, gitmesi gereken sisteme gönderilir. Yani anahtarlar ağı parçalara bölerek ağ performansını artırır. Aynı zamanda ağ durumunu izler, veriyi gönderip, iletim işleminin yapıлып yapılmadığını test eder.

3.1.2. Kablolama

Sistemin ağ yapısı kurulurken sunucu-istemci-hub arasındaki bağlantılar UTP (Unshielded Twisted Pair) yani kaplamasız dolanmış çift kablo ile gerçekleştirilmiştir. UTP kablo koaksiyel kabloya göre elektromanyetik alan etkisine daha az dirençli olsa da, yıldız topoloji için yeteri kadar güvenli ve ucuz bir yöntemdir ve günümüzdeki en yaygın kablolama yöntemidir. UTP kablolamada RJ-45 (Registered Jak) kodlu konnektör (jak) kullanılır. Kablo jak'a takılırken, TX (Transmit) kanalı bir çift kabloyu, RX (Receive) ise başka bir çift kabloyu kullanacak şekilde yapılmalıdır. Bu çalışmada olduğu gibi sayısal iletişimin yapılacağı hatlarda bu kurala uyulmalıdır. Gönderim ve alım kanalında aynı çiftin tellerini kullanmanın faydasını şekil 3.7.'de verilen örnek sinyal üzerinde inceleyelim.



Şekil 3.7. UTP kablonun sinyal bozulmasına karşı gösterdiği direnç

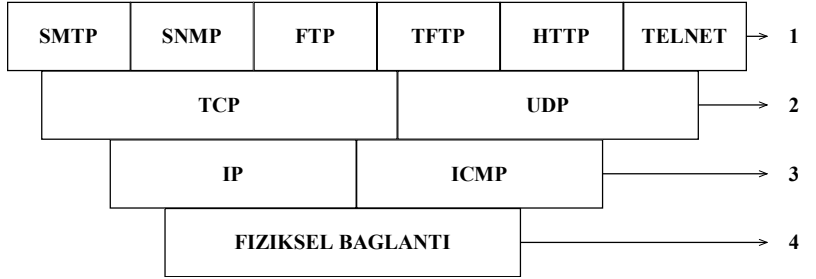
Temel olarak TX+ kanalında toprağa göre artı, TX- kanalında ise toprağa göre eksi değerinde bir elektrik sinyalinin iletildiğini düşünelim. Kablonun geçtiği yerlerde bir noktada kabloya elektromanyetik bir etki oluyor ve yola kare dalga sinyal olarak çıkan elektrik sinyali kare dalga üzerinde gösterildiği gibi bozulmaya uğruyor. Sinyalin üzerinde gittiği teller aynı dolanmış çiftin telleri olduğu için sinyaldeki değişim, tellerin her ikisine de aynı biçimde gerçekleşiyor. Alıcı taraf (RX) bu iki sinyali karşılaştırarak tekrar kare dalga sinyali elde edebiliyor.

3.2. İletişim Kontrol Protokolü / İnternet Protokolü (TCP/IP)

TCP/IP nin tarihi 1970 lerin başlarında İleri Araştırma Projeleri Ajansı (ARPA) nin yürüttüğü paket anahtarlamalı ağ deneylerine kadar dayanır. TCP/IP nin ortaya çıkmasını sağlayan proje, bir felaket anında bilgisayarların birbirleriyle iletişiminin kesilmemesini amaçlıyordu. Şu anda internetin temeli olan TCP/IP nin amacına fazlasıyla ulaştığı söylenebilir.

TCP/IP ile kurulan bir bilgisayar ađında, bir bilgisayarı üç parametreyle tanımlarız. Bu parametreler bilgisayarın adı, IP adresi ve MAC adresidir. TCP/IP protokoller kümesi bu üç parametreyi kullanarak bilgisayarları birbirine bağlar ve güvenli bir iletişim sağlar.

TCP/IP protokol kümesinin sahip olduđu mimari katmanlı yapıdadır. Sırasıyla uygulama, ulaşım, yönlendirme ve fiziksel katmanlardan oluşan bu yapı şekil 3.8.'de verilmiştir.



Şekil 3.8. TCP/IP katmanları

1. Uygulama Katmanı
2. Ulaşım Katmanı
3. Yönlendirme Katmanı
4. Fiziksel Katman

3.2.1. İletişim kontrol protokolü (TCP)

TCP'nin (transmission control protocol-iletişim kontrol protokolu) temel işlevi, üst katmandan (uygulama katmanı) gelen bilginin segmentler haline dönüştürülmesi, iletişim ortamında kaybolan bilginin tekrar yollanması ve ayrı sıralar halinde gelebilen bilginin doğru sırada sıralanmasıdır. IP (internet protocol) ise tek tek datagramların yönlendirilmesinden sorumludur. Bu açıdan bakıldığında TCP katmanının hemen hemen tüm işi

üstlendiđi görölmekle beraber (küçük ağlar için bu doğrudur) büyük ve karmaşık ağlarda IP katmanı en önemli görevi üstlenmektedir. Bu gibi durumlarda deđişik fiziksel katmanlardan geçmek, doğru yolu bulmak çok karmaşık bir iş halini almaktadır.

3.2.2. İnternet protokolü (IP)

TCP katmanına gelen bilgi segmentlere ayrıldıktan sonra IP katmanına yollanır. IP katmanı, kendisine gelen TCP segmenti içinde ne olduđu ile ilgilenmez. Sadece kendisine verilen bu bilgiyi ilgili IP adresine yollamak amacındadır. IP katmanının görevi bu segment için ulaşılmak istenen noktaya gidecek bir “yol” (route) bulmaktır. Arada geçilecek sistemler ve geçiş yollarının, bu paketi doğru yere geçirmesi için kendi başlık bilgisini TCP katmanından gelen segment’e ekler. TCP katmanından gelen segmentlere IP başlığının eklenmesi ile oluşturulan IP paket birimlerine datagram adı verilir.

3.3. Sistem Yazılımı

3.3.1. Mesaj formatı

Bu çalışmada gerçekleştirilen, sunucu ve istemci bilgisayar üzerinde koşan yazılım Borland Delphi Enterprise 6.0 sürümü ile gerçekleştirilmiştir. Delphi temelde Pascal dilinin, grafiksel uygulama geliştirme ortamına dönüştürölmüş şeklidir. Microsoft Windows uygulamaları geliştirmek için yaygın olarak kullanılır. Diđer bir deyişle, stand alone (tek başına) uygulamalar geliştirmek için kullanılan grafik bir araçtır. Özellikle yeni sürümleriyle (Delphi 5, 6 ve 7), veritabanı uygulamaları, ODBC bağlantısı, XML ve web tabanlı uygulamalar geliştirmeyi kolay bir şekilde sağlar.

Oluşturulan yazılımla öncelikle sunucu bilgisayarın seri portuna (comport) aç komutu gönderilerek, haberleşme kanalı açılır ve kan gazı cihazından gelecek veriler beklenmeye başlanır. Kan gazı analiz cihazı, kan örneđini inceledikten sonra sonuçları RS 232 portu üzerinden sunucu bilgisayara gönderir.

Sunucu bilgisayar tarafından alınan veriler LIS 2 mesaj formatındadır. Bir çerçeve, mesajın başlangıcını gösteren bir karakter (start-of-message), mesajın sonunu gösteren bir karakter (end-of-message), iki adet kontrol karakteri ve iletimin bittiğini gösteren (end-of-transmission) bir karakterden oluşur. Çerçeve formatı aşağıda gösterilmiştir.

<STX>{tanımlayıcı-identifier}<RS>{veri-data}<RS><ETX>kontrol-checksum<EOT>

|<-----Uygulama Mesajı-Application Message----->|

Buradaki karakterler;

<STX> :ASCII kontrol karakteri, Start of Text (02 decimal)

<ETX> :ASCII kontrol karakteri, End of Text (03 decimal)

checksum : İki basamaktan oluşan bir değişkendir. Alıcının hata kontrolü yapabilmesini sağlar.

<RS> :ASCII kontrol karakteri, Record Seperator (30 decimal)

<EOT> :ASCII kontrol karakteri, End of Transmission (04 decimal)

Uygulama mesaj formatı sınırları, başlangıç ve bitiş karakterleri ile belirlenmiştir. Tanımlayıcı ve veri bölgeleri, kayıt ayıraçları (RS) ile sonlandırmıştır. Örnek olarak yapısı aşağıda verilmiştir.

{tanımlayıcı}<RS>{veri}<RS>

Tanımlayıcı bilgidен sonra (tanımlayıcı bölgenin bitiminden önce) yine ASCII kontrol karakterlerinden alan ayıracı (FS) gelmektedir. Bu da bir örnekle gösterilirse,

SMP_NEW_DATA<FS>

<FS> :ASCII kontrol karakteri, Field Separator (28 decimal)

Veri kayıt bölgelerinin formatı biraz daha değişiktir. Aşağıda veri kayıtlarının gelebileceği format biçimi gösterilmiştir.

{isim}<GS>{değeri}<GS>{birimi}<GS>{referans dışı bilgisi}<GS><FS>

<GS> :ASCII kontrol karakteri, Group Separator (29 decimal)

Tüm gurupların ismi, değeri, birimi ve referansdışı bilgisi yazdırılabilir ASCII karakterlerinden (decimal 20 den 126 ya kadar) oluşmaktadır. Bir parametrenin değeri referans aralığının dışında ise referans dışı bilgisi gönderilir. Değer, referansın üst sınırından büyük ise H, küçük ise L karakteri gönderilir. Referans aralığında ise herhangi bir uyarı bilgisi gönderilmez.

<GS> Referans dahilinde,
{referans dışı bilgisi}<ETB><GS> Referans dışında.

<ETB> :Exception (23 decimal)

<GS> karakteri, boş guruplarda dahil olmak üzere tüm gurupları sonlandırır. Burada her bir kayıt alanı dört adet <GS> karakteri içermelidir. <FS> karakteri ile kayıt alanları sonlandırılır.

Aşağıda örnek, bir veri mesajının yapısı gösterilmektedir. Gönderilebilecek tüm parametreler verilmiştir.

```
<STX>SMP_NEW_DATA<FS><RS>aMOD<GS>850/270<GS><GS><FS>
>iIID<GS>1001/222<GS><GS><GS><FS>rTYPE<GS>SAMPLE<GS><
GS><GS><FS>rSEQ<GS>101<GS><GS><GS><FS>iACC<GS>4567<G
S><GS><FS>rDATE<GS>16Feb94<GS><GS><GS><FS>rTIME<G
S>08:33<GS><GS><GS><FS>iPID<GS>158-
665232<GS><GS><GS><FS>iDATE<GS>16Feb94<GS><GS><GS><FS>
>iTIME<GS>08:29<GS><GS><GS><FS>iOID<GS>147<GS><GS><GS>
<FS>iROOM<GS>CCU-
4<GS><GS><GS><FS>iAGE<GS>68<GS><GS><GS><FS>iDOB<GS>2
4Dec1925<GS><GS><GS><FS>iSEX<GS>F<GS><GS><GS><FS>iDID
<GS>007<GS><GS><GS><FS>iSOURCE<GS>ARTERIAL<GS><GS><
GS><FS>rDEVICE<GS>CAPILLARY<GS><GS><GS><FS>mpH<GS>7
.381<GS><GS><GS><FS>mPCO2<GS>48.5<GS>mmHg<GS>H<ETB><
GS><FS>mPO2<GS>112.4<GS>mmHg<GS>H<ETB><GS><FS>mBP<G
S>760<GS>mmHg<GS><GS><FS>cHCO3std<GS>26.3<GS>mmol/L<G
S><GS><FS>cHCO3std<GS>27.6<GS>mmol/L<GS><GS><FS>cBE(vt)<
GS>1.9<GS>mmol/L<GS><GS><FS>cBE(vv)<GS>2.5<GS>mmol/L<GS
><GS><FS>cpH<GS>7.381<GS><GS><GS><FS>cPO2<GS>48.5<GS>m
mHg<GS><GS><FS>cPO2<GS>112.4<GS>mmHg<GS><GS><FS>cA-
aDO2<GS>44.5<GS>mmHg<GS><GS><FS>ca/A<GS>72<GS>%<GS><
```

GS<FS>mtHb<GS>11.2<GS>g/dL<GS>L<ETB><GS><FS>cHet<GS>3
2<GS>%<GS><GS><FS>mO2Hb<GS>97.1<GS>%<GS><GS><FS>mC
OHb<GS>1.9<GS>%<GS>H<ETB><GS><FS>mMetHb<GS>0.7<GS>%
<GS><GS><FS>mHHb<GS>0.3<GS>%<GS><GS><FS>msO2<GS>99.7
<GS>%<GS><GS><FS>cO2CAP<GS>15.2<GS>mL/dL<GS>L<ETB><G
S><FS>cctO2(a)<GS>15.5<GS>mL/dL<GS><GS><FS>mNa+<GS>132<
GS>mmol/l<GS>L<ETB><GS><FS>mK+<GS>4.94<GS>mmol/L<GS><
GS><FS>mCl-
<GS>102<GS>mmol/L<GS><GS><FS>cAnGap<GS>7.4<GS>mmol/L<G
S><GS><FS>mCa++<GS>1.25<GS>mmol/L<GS><GS><FS>cCa++<GS
>1.24<GS>mmol/L<GS><GS><FS>iP50<GS>50.0<GS>mmHg<GS><GS
><FS>cRI<GS>0.40<GS><GS><GS><FS>iRR<GS>15.0<GS>b/min<GS
><GS><FS>iTEMP<GS>37.0<GS>C<GS><GS><FS>iFIO2<GS>30.0<G
S>%<GS><GS><FS>iFlow<GS>2.00<GS>L/min<GS><GS><FS><RS><
ETX>42<EOT>

İletilen her bir karakterin büyük veya küçük olması önemlidir. İsimlerin ilk karakteri her zaman küçük harfle başlar. Bu karakterlerin yanlarındaki verinin değerinin nasıl oluşturulduğunu belirtir.

m: sistem tarafından ölçülmüştür.
c: sistem tarafından hesaplanmıştır.
i: operatör tarafından girilmiştir.
a,r,s: sistem tarafından atanmıştır.

Gönderilen parametreler nümerik değerlere sahiptir. Diğer değişkenler alfanümeriktir. Tarih bilgisi, ggAaayy formatındadır. gg karakterleri ayın kaçınıcı günü olduğunu, Aaa ayın kısaltılmış ismini, yy ise yılı ifade eder. Doğum tarihi bilgisi formatı ise ggAaayyyy şeklindedir.

Ay isimlerinin kısaltılmış halleri ise; Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec şeklindedir.

Zaman bilgisi nn:nn formatında, standart 24 saat bilgisidir.

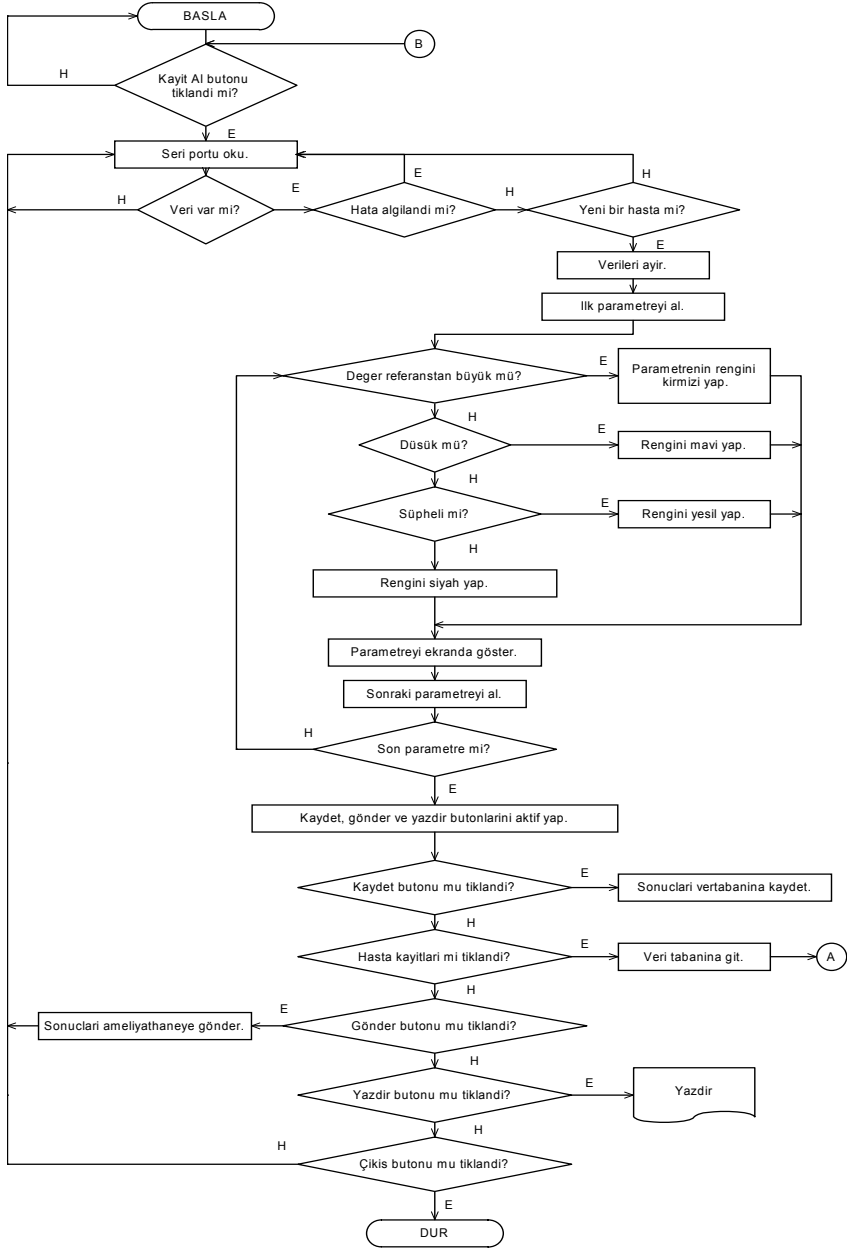
Birim isimleri sabit bilgi halinde gönderilmektedir. Ancak F₁O₂ ve pH parametresinin birimi rapor edilmemektedir. Bu durumda ilgili birim gurupları arası boştur.

Cihazı tanımlayıcı bilgi (iIID, instrument identifier) LIS sistemlerinde oldukça önemlidir. Bağlantı kurulacak cihazı tanımlar. Sistem kurulurken cihazın seri numarası, ID numarası olarak cihazın setup ayarlarına girilmektedir.

3.3.2. Sistemin işleyiş adımları

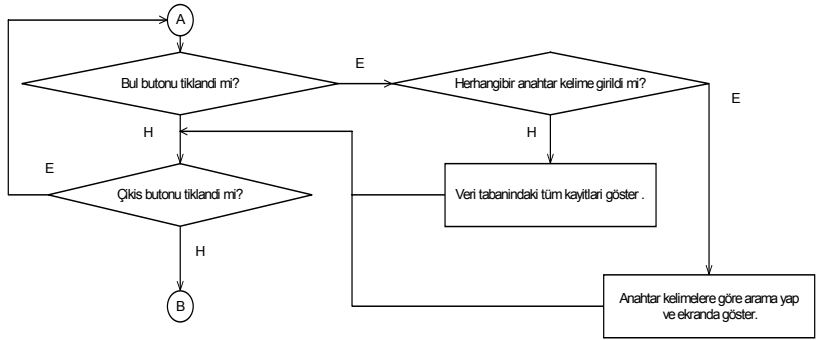
Sunucu bilgisayar ilk açıldığında kayıt al butonuna tıklanarak seri haberleşme portunun açılması sağlanır. Sürekli, porta bilgi gelip gelmediğinin kontrolü yaptırılır. Eğer yeni bir bilgi gelirse yazılım her bir parametreyi ismine, değerine, birimine, referanstan düşük veya yüksek olmasına göre tek tek ayıklar ve bir string grid'de toplar (Yanık 2000). Aynı parametreye ait bilgiler aynı satırda farklı hücelere yerleştirilir. Cihazın kendisiyle, hasta kimliğiyle, zaman bilgileriyle, hastanın analiz sonuçlarıyla ilgili tüm veriler işlenir. Ayrıca hastanın kan örneği parametrelerinin referans değerlerinden yüksek veya düşük olması durumlarında parametre, sayısal değeri ve varsa biriminin rengi değiştirilir. Sunucu ve istemci bilgisayarda herbir parametrenin ekranın neresinde gösterileceği belirlidir. Kan gazı cihazında, incelemesi yapılacak parametreler seçilebilmektedir. Eğer bir parametreye ait sayısal değer sunucuya ulaşmaz ise söz konusu parametre(ler) ekranda gösterilmez.

Sunucu bilgisayara ulaşan veriler istenildiğinde veri tabanına kaydedilebilir. Kan gazı cihazının sınırlı sayıdaki kayıt saklayabilme yeteneğine karşılık sunucu bilgisayarın kapasitesi çok fazladır. Sunucu bilgisayarda ayrıntılı bir şekilde eski hasta kaydı araması da yaptırılabilir. İstenirse sonuçlar yazdırılabilir. Şekil 3.9.'da oluşturulan sistemin işleyişine ait, verilerinin alınmasından itibaren sırasıyla gerçekleştirilen adımları gösteren akış diyagramı verilmiştir.



Şekil 3.9. Sistemin akış diyagramı

Hasta kayıtları arama butonu tıkladığında, daha önceden incelenmesi yapıp veri tabanına kaydedilmiş eski hasta kayıtları arasında arama yapılabilir. Yedi farklı kriter kutusuna anahtar kelime girilerek aranan sonuca daha rahat ulaşılabilir. Program eski hasta kayıtlarını paradox veri tabanında saklamaktadır (Karagülle vd. 2001). Girilen anahtar kelimeye en yakın olan sonuçlar ekranda gösterilecektir. Hiçbir anahtar kelime girilmeden arama yaptırılırsa, program veri tabanındaki tüm hasta kayıtlarını ekrana getirecektir. Şekil 3.10.'da, programın hasta kayıtları butonuna basıldıktan sonra işlenen adımlar gösterilmektedir.



Şekil 3.10. Hasta kaydı arama fonksiyonu akış diyagramı

3.3.3. Hata denetimi

Kullanılan LIS protokolleri, onaltılı sayı sisteminde ifade edilmiş, 00 ile FF arasında değişen iki basamaklı bir ASCII kodu ile hata denetimi yapılmasını sağlamaktadır. İki basamaklı bu kontrol sayısı, <STX>.....<ETX> arasındaki (<STX> ve <ETX> dahil) tüm karakter değerlerinin toplamı kullanılarak üretilmektedir. Karakterlerin ASCII koduna karşılık gelen değerlerin toplamı, mod 256 sayı sistemine dönüştürülür. Bu işlemin sonunda elde edilen değer de onaltılı sayı sistemine çevrilir. Çıkan sonuç <ETX> ile <EOT> arasına yazılır.

Veriler sunucu bilgisayara ulaştıktan sonra da aynı işlem sunucu yazılımı tarafından tekrarlanır. Eğer elde edilen değer kontrol karakterinden farklı

ise, verilerin hatalı ulaştığı anlaşılır ve program, kullanıcıya verileri tekrar göndermesi gerektiğini bildirir.

Bir örnek veri üzerinde hata algılama sistemini inceleyelim.

```
<STX>ID DATA<FS><RS>aMOD<GS>LIS<GS><GS><GS><FS>iIID<GS>333<GS><GS><GS><FS><RS><ETX>84<EOT>
```

Yukarıdaki verinin her bir karakterinin değeri toplanıp önce mod 256 ya ardından da onaltılı sayı sistemine çevrilirse sonuç çizelge 3.3.3.1.'deki gibi olur.

Çizelge 3.3. Kontrol karakterinin hesaplanması

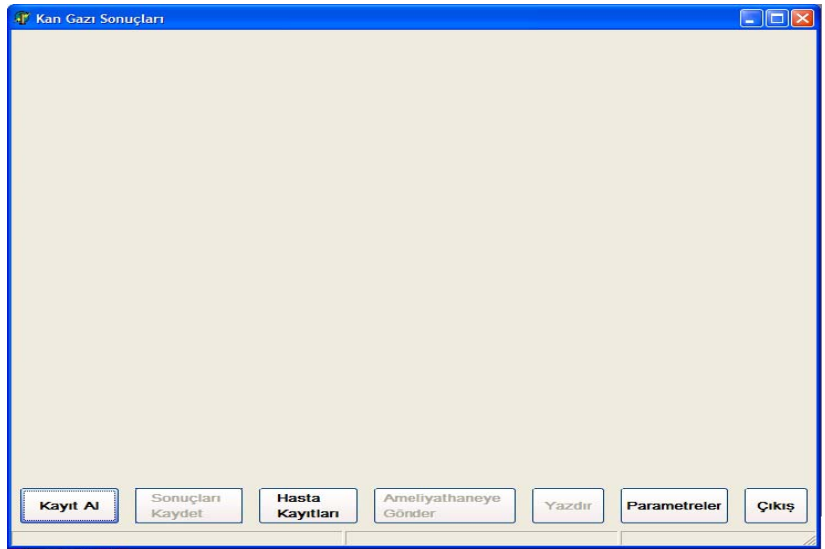
<u>Yapılan işlem</u>	<u>Sonuç</u>
Karakter değerlerinin toplamı	1924
Mod 256_{10} çevrilmiş değeri	132
İki basamaklı onaltılı sayı sistemi değeri	84

Örnek verinin kontrol karakter değeri onaltılı sayı sisteminde 84 olarak gönderilmiştir. Yukarıda gösterilen şekilde, yazılım tarafından hesaplanan kontrol değeri de 84 çıkar ise sistem veri iletiminde herhangi bir hata algılamayacak ve iletimin başarı ile gerçekleştiği anlaşılacaktır.

4. ARAŐTIRMA BULGULARI

4.1. Sunucu Yazılımı

Kan gazı laboratuvarı personeli tarafından kullanacak sunucu bilgisayar yazılımı kolay anlaşılır bir arayüze sahiptir. Ayrıca sunucu ekranında parametrelerin sıralanışı, kan gazı cihazının orijinal analiz çıktı görüntüsüne sadık kalınarak tasarlanmıştır. Yazılımda tüm pencerelerin sol üst köşesinde ilgili menünün adı yer almaktadır. Sunucu bilgisayar yazılımının çalıştırılması ile ekrana gelen görüntü şekil 4.1.'de verilmiştir.



Şekil 4.1. Sunucu yazılımının başlangıç penceresi

Programın çalıştırılmasının ardından ekrana ilk olarak şekil 4.1.'deki görüntü gelmektedir. Sunucunun verileri almaya hazır bir hale gelmesi için "kayıt al" butonuna basmak yeterlidir. Butona basıldığında, pencerenin en altındaki durum çubuğunda (status bar) "Sonuçlar bekleniyor." bilgisini alırız. "Kayıt Al" butonuna tekrar basıldığında ise aynı yerde "Haberleşme

Kapalı” ibaresiyle seri portun iletişime kapandığını anlarız. Böylelikle kullanıcı, sistemin o anki durumu hakkında bilgi sahibi olmaktadır. Seri portun haberleşmeye açılmasından sonra program, verileri beklemeye başlar. Henüz bir veri alınmadığı durumda, ekranda da görüldüğü üzere sadece “Kayıt Al”, “Hasta Kayıtları”, “Parametreler” ve “Çıkış” butonları aktif durumdadır. Bu anda ekranda bir veri olmadığı için “Sonuçları Kaydet”, “Ameliyathaneye Gönder” ve “Yazdır” butonları ile herhangi bir işlem yapılamamaktadır. Ancak eski hasta kayıtları için arama yapılabilir ve parametreler butonuyla istemci bilgisayar IP leri değiştirilebilir.

Program penceresinin sağ üst köşesindeki, simge durumuna küçült, ekranı kapla ve kapat sekmeleri de kullanılabilir durumdadır ancak program dahilinde bu sekmelerde değişiklik yapılabilir. İstenildiğinde bilgisayarın açılması ile birlikte bu programın tam ekran biçiminde çalışması sağlanabilir ve bilgisayar üzerinde başka bir windows uygulamasının çalışması da engellenebilir.

Çıkış butonu programı kapatmak için kullanıldığı gibi kaynak kodlara eklenecek bir komutla, programla birlikte bilgisayarı da kapatmak için kullanılabilir.

Kan gazı cihazından sunucu bilgisayara verilerin gelmesiyle birlikte sunucu bilgisayar ekranı görüntüsü şekil 4.2.’deki gibi değişir.

Kan Gazı Sonuçları

MODEL845 IID 03802 SEQ# 68599 22Sep02 23:48
ACC# 2613902

Blood Gas

pH 7.345
pCO2 48.7 mmHg
pO2 31.0 mmHg
BP 690 mmHg

CO-OXIMETRY

tHb 9.3 g/dL
sO2 57.0 %
O2Hb 54.6 %
COHb 3.9 %
MetHb 0.3 %
HHb 41.2 %
ctO2(a) 7.2 mL/dL

Other Results

HC03act 26.0 mmol/L
HC03std 23.9 mmol/L
tCO2 27.5 mmol/L
BE(vt) 0.0 mmol/L
BE(w) 0.3 mmol/L
Hct 27 %
O2CAP 12.4 mL/dL
P50 26.1 mmHg

Kayıt Al Sonuçları Kaydet Hasta Kayıtları Ameliyathaneye Gönder Yazdır Parametreler Çıkış

Sonuçlar Alındı

Şekil 4.2. Analiz sonuçlarının alınmasından sonra sunucu bilgisayar ekran görüntüsü

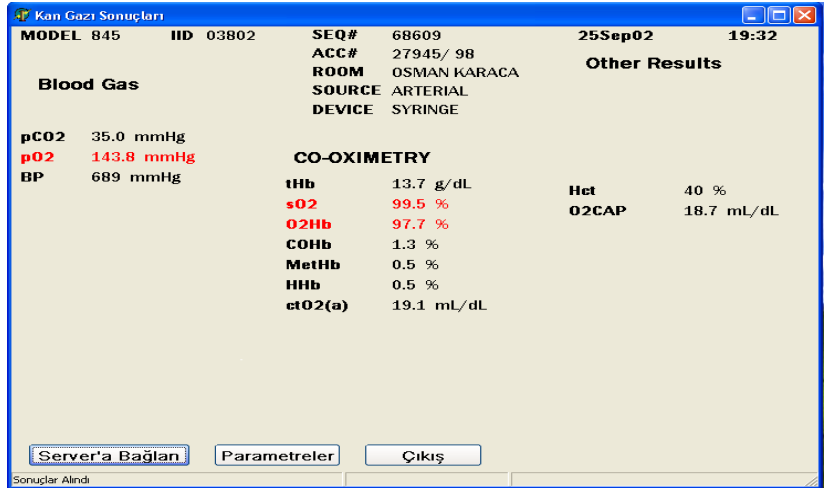
Kan gazı cihazında kan örneğinin incelenmesinin ardından analiz sonuçları sunucu bilgisayar tarafından alınır ve başlangıç ekran görüntüsünde değişiklik olur. Cihaza ait model ve IID numarası, hastaya ait seq, acc numaraları, örneğin alındığı kaynak ve alet, tarih ve saat bilgileriyle birlikte hastanın kan gazı analiz sonuçları ekrandaki yerini alır. Hastaya ait analiz sonuçları blood gas, co-oximetry ve other results ana başlıkları altında sıralanır. Durum çubuğunda “Sonuçlar Alındı” ibaresiyle kullanıcıya bilgi verilir. Yine dikkat edilirse bazı parametreler, değerleri ve birimleri ile birlikte ekrana farklı renklerde yansımaktadır. Bu, analiz sonuçlarının olması gerektiği aralığın dışında olduğunu ifade eder.

Ekrandaki bir diğer önemli değişiklikte, veriler gelmeden önce herhangi bir fonksiyonu olmayan “Sonuçları Kaydet”, “Ameliyathaneye Gönder” ve “Yazdır” butonlarının sonuçların gelmesiyle birlikte aktif duruma geçmesidir. Diğer butonların konumunda bir değişiklik olmamıştır.

Hasta verilerini, oluşturulan veri tabanına kaydetmek için “Sonuçları Kaydet” tuşuna basmak yeterlidir. Böylelikle istenildiğinde eski analiz sonuçlarına rahatlıkla ulaşılabilir. “Yazdır” butonuyla da yazıcı çıktıyı alma imkanı vardır. “Ameliyathaneye gönder” butonuna basılması ile birlikte analiz sonuçları sunucu bilgisayar ethernet kartı, hub ve istemci bilgisayar ethernet kartı üzerinden istemciye ulaşır. Ameliyathane istemci bilgisayar ekranında aynı sonuçlar görüntülenir.

4.2. İstemci Bilgisayar Yazılımı

Kan gazı laboratuvarında kan örneği incelendikten sonra analiz sonuçlarının ameliyathane personeline iletilmesi için “Ameliyathaneye Gönder” butonunun tıklanması yeterlidir. Şekil 4.3.’de gösterilen başka bir hastaya ait analiz sonuçlarının ameliyathane istemci bilgisayar ekranındaki görüntüsünü inceleyelim.



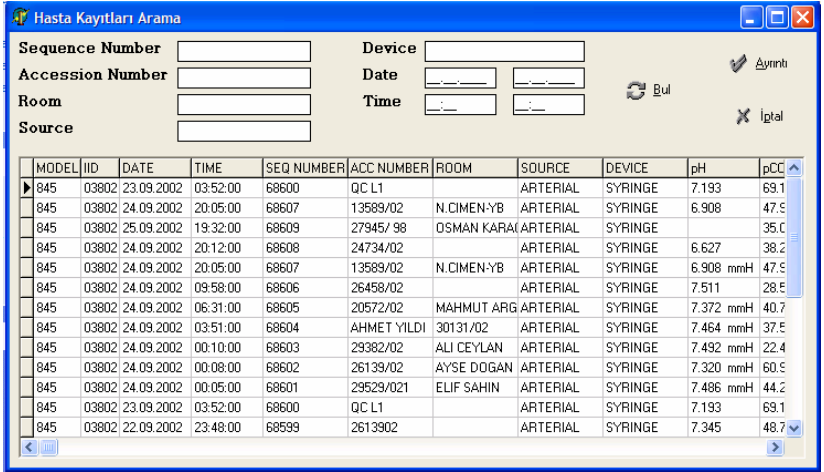
Şekil 4.3. İstemci bilgisayar yazılımı ekran çıktısı

İstemci için hazırlanan yazılımın kullanıcı arayüzünde “server’a bağlan”, “parametreler” ve “çıkış” butonları görülmektedir. İstemci bilgisayar ilk

açıldığında sunucu bilgisayara bağlanması için “server’a bağlan” tuşunun tıklanması gerekir. Bu tuş tıklandığında istemci sunucuya bağlanır ve sunucu yazılımının durum çubuğunda “istemci bilgisayar bağlantı isteği kabul edildi” şeklinde bir bilgi kan gazları laboratuvarı teknisyenine iletilir. Böylece kullanıcı istemci bilgisayarın, analiz sonuçlarını almaya hazır olduğunu anlar. “Parametreler” butonunda sunucu IP numaralarında değişiklik olması halinde yeniden giriş için düşünülmüştür. “Çıkış” butonu yazılımı kapatmak için kullanılmaktadır.

4.3. Hasta Kayıtları Menüsü

Sunucu bilgisayar yazılımındaki “Hasta Kayıtları” menüsünün fonksiyonu aşağıda anlatılmaktadır.

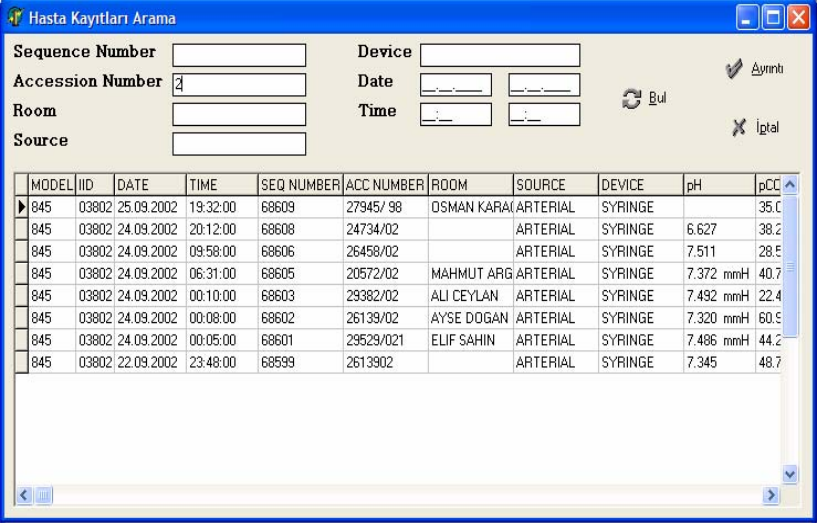


MODEL	ID	DATE	TIME	SEQ NUMBER	ACC NUMBER	ROOM	SOURCE	DEVICE	pH	pCC
845	03802	23.09.2002	03:52:00	68600	QC L1		ARTERIAL	SYRINGE	7.193	69.1
845	03802	24.09.2002	20:05:00	68607	13589/02	N.CIMEN-YB	ARTERIAL	SYRINGE	6.908	47.9
845	03802	25.09.2002	19:32:00	68609	27945/ 98	OSMAN KARAI	ARTERIAL	SYRINGE		35.0
845	03802	24.09.2002	20:12:00	68608	24734/02		ARTERIAL	SYRINGE	6.627	38.2
845	03802	24.09.2002	20:05:00	68607	13589/02	N.CIMEN-YB	ARTERIAL	SYRINGE	6.908 mmH	47.9
845	03802	24.09.2002	09:58:00	68606	26458/02		ARTERIAL	SYRINGE	7.511	28.9
845	03802	24.09.2002	06:31:00	68605	20572/02	MAHMUT ARG	ARTERIAL	SYRINGE	7.372 mmH	40.7
845	03802	24.09.2002	03:51:00	68604	AHMET YILDI	30131/02	ARTERIAL	SYRINGE	7.464 mmH	37.9
845	03802	24.09.2002	00:10:00	68603	29382/02	ALI CEYLAN	ARTERIAL	SYRINGE	7.492 mmH	22.4
845	03802	24.09.2002	00:08:00	68602	26139/02	AYSE DOGAN	ARTERIAL	SYRINGE	7.320 mmH	60.9
845	03802	24.09.2002	00:05:00	68601	29529/021	ELIF SAHIN	ARTERIAL	SYRINGE	7.486 mmH	44.2
845	03802	23.09.2002	03:52:00	68600	QC L1		ARTERIAL	SYRINGE	7.193	69.1
845	03802	22.09.2002	23:48:00	68599	2613902		ARTERIAL	SYRINGE	7.345	48.7

Şekil 4.4. Hasta kayıtları arama menüsü

Şekil 4.4.’deki görüntü, hasta kayıtları arama menüsünde herhangi bir anahtar kelime girilmeden “bul” butonunun tıklanmasından sonra alınmıştır. Bu durumda yazılım, veri tabanındaki tüm hasta kayıtlarını ekrana getirmektedir. Eğer aranan hastayla ilgili bir anahtar bilgi menüdeki ilgili kutulara girilir ise yazılım bize doğru hastayı veya anahtar bilgiye en yakın

hastaları ekrana getirecektir. Örnek olarak şekil 4.5.’de “accession number” değeri 2 ile başlayan hastaların aratılıp bulunan sonuçların gösterildiği pencere sunulmuştur.



MODEL	IID	DATE	TIME	SEQ NUMBER	ACC NUMBER	ROOM	SOURCE	DEVICE	pH	pCC
845	03802	25.09.2002	19:32:00	68609	27945/ 98	OSMAN KARAI	ARTERIAL	SYRINGE		35.0
845	03802	24.09.2002	20:12:00	68608	24734/02		ARTERIAL	SYRINGE	6.627	38.2
845	03802	24.09.2002	09:58:00	68606	26458/02		ARTERIAL	SYRINGE	7.511	28.5
845	03802	24.09.2002	06:31:00	68605	20572/02	MAHMUT ARG	ARTERIAL	SYRINGE	7.372 mmH	40.7
845	03802	24.09.2002	00:10:00	68603	29382/02	ALI CEYLAN	ARTERIAL	SYRINGE	7.492 mmH	22.4
845	03802	24.09.2002	00:08:00	68602	26139/02	AYSE DOGAN	ARTERIAL	SYRINGE	7.320 mmH	60.5
845	03802	24.09.2002	00:05:00	68601	29529/021	ELIF SAHIN	ARTERIAL	SYRINGE	7.486 mmH	44.2
845	03802	22.09.2002	23:48:00	68599	2613902		ARTERIAL	SYRINGE	7.345	48.7

Şekil 4.5. Veri tabanında Accession Number’ı 2 ile başlayan hastalar

Hasta kayıtları arama menüsü sayesinde laboratuvar ve hastalarla ilgili istatistiki bilgilerde elde edilebilir. Örneğin verilen tarihler arasında kaç hastanın kan örneğinin incelendiği bilgisine rahatlıkla ulaşılabilir. Bu sayede laboratuvardaki çalışma yoğunluğu dönem dönem incelenebilir.

5. TARTIŞMA VE SONUÇ

Bilişim teknolojisi, bilginin bilgisayar teknolojisiyle elde edilmesi, işlenmesi, saklanması ve gerekli yerlere aktarılması şeklinde tanımlanabilir. Sağlık uygulamalarında ve araştırmalarında bilişim teknolojileri giderek vazgeçilmez öğeler olarak yerini almaktadır. Bu hızlı gelişme cihaz ve yazılım sorunlarını, veri tabanları ve kod alt yapısında uyum gereksinimlerini, hasta merkezli uygulamaları ve en önemlisi tüm bu uygulamalarda eşgüdüm ve maliyet etkinlik unsurlarını ön plana çıkarmaktadır.

Hastanelerde genel olarak ameliyathane ünitesi, yoğun bakım ünitesi ve laboratuvar üniteleri bina içerisinde ayrı ayrı birimlere bazen de farklı katlara kurulmak zorunda kalınmaktadır. Ameliyat sırasında yaklaşık 20 dakikada bir hastanın kan örneği alınıp incelenmek üzere laboratuvarlara gönderilmektedir. Yine aynı şekilde yoğun bakım ünitelerindeki hastalarında sık sık kan örnekleri incelenmek üzere laboratuvarlara gönderilmektedir. Bu durumda laboratuvarlardan ameliyathanelere ve yoğun bakım ünitelerine doğru analiz sonuçlarının elle taşınması gerekmektedir. Bu şekilde bir iletim hem zaman hem de sterilizasyon problemini beraberinde getirmektedir.

Bu çalışmada kan gazı analiz cihazlarından ameliyathanelere bilgisayar kontrollü veri iletimi incelenmiştir. Gerçekleştirilen bir uygulama ve sonuçları açıklanmıştır. Tasarlanan sistem ile zaman ve sterilizasyon problemi ortadan kaldırılmaktadır.

Laboratuvarında, kan gazı analiz cihazının yanına kurulan sunucu bilgisayar bir saniyeden kısa bir sürede seri haberleşme portu üzerinden analiz sonuçlarını almakta, işlemekte ve istemci bilgisayara gönderilebilmektedir. Sunucu bilgisayar ana formundaki parametreler menüsünden seçim yapılarak analiz sonuçlarının birden fazla istemci bilgisayara gönderilebilmesi mümkündür. Tasarlanan sistem, oluşturulan yerel ağ üzerinde haberleşebildiği gibi varolan hastane ağı üzerinde de bilgi iletimi yapabilmektedir. Sonuçlar, sunucu bilgisayar yazılımı ile veri tabanına kaydedilip arşivlenmesi sağlanabilir ve yazdırılabilir durumdadır. Ayrıca eski hasta kayıtları oluşturulan veri tabanında aranabilmektedir. Parametre değerlerinin referans aralığının dışında olması durumunda sunucu ve istemci ekranlarında ilgili

sonular renkli olarak grlmektedir. Parametre deęeri referans aralıęının stnde veya altında ise rengi sırasıyla kırmızı veya mavi olur. Eęer Őüpheli ise rengi yeŐil olmaktadır. Deęer normal aralık dahilinde ise rengi siyah olur. Bylelikle ameliyat ekibinin parametrelerdeki kritik durumları kolaylıkla farketmeleri saęlanmaktadır. Tasarlanan sistem bir laboratuvar bilgi sistemi olarak da adlandırılabilir.

OluŐturulan sistem, gelecekte daha da geliŐtirilerek hasta sonularının hastanenin her yerinden veya internet zerinden takibi yapılabilir hale getirilebilir. Kullanıcı yetkilendirmesi ile sadece yetkili kiŐilerin sonu grmesi veya hastalarında, kendi sonularını kimlik bilgisi doęrulaması sonrasında grebilmesi saęlanabilir.

KAYNAKLAR

Akan, H. 2001. Türkiye’de Teletıp, Tıp Eğitimi Bülteni, Sayı 2; 6,
http://www.medicine.ankara.edu.tr/basic_medical/basicmed/Tip%20dosya/bulten2.pdf

Anonymous. 1994. Overview of the Data Communication
Protocols, Handbook, Ciba Corning’s 800 Series Systems,
Chiron Diagnostics.

Anonymous. 1994. Operator’s Manual, Manual Book,
RapidLab800 Critical Care Systems, Chiron Diagnostics.

Belden, T., Sembrowich, W.L., Deetz, D.W., Solomon, F.A.
1993. Developing an advanced ‘tool’ for the clinician; using
industrial design and interface design together to bring
technology into the hand of the user, Computer-Based

Medical Systems, Proceedings of Sixth Annual IEEE
Symposium.229-234.

Çölkesen, R. 2000. Network-TCP/IP-Unix, Papatya
Yayıncılık, 49-60, İstanbul.

Hardalaç, F., Polat, H., Barışçı, N. and Ergün, U. 2002.
Web-based Telemedicine: Remote Auscultation System On
The Internet, Turkish Journal of Telecommunication vol
1;33-39.

Johanson, M., Gustafon, M., Johanson, L. 2002. A Remote
Auscultation Tool for Advanced Home Healthcare, Journal
of Telemedicine and Telecare, vol 8.

Jurgen, R.K. 1992. Technology 1992-medical electronics. Spectrum IEEE,29; 61-62.

Kalogeropoulos, D., Carson, E.R., Collinson, P.O., Cramp, D.G., 1995. Development of an object-oriented intelligent clinical information management system. Engineering in Medicine and Biology Society. IEEE 17th Annual Conference. 1; 745-746.

Karagülle, İ. ve Pala, Z. 2001. Borland Delphi 5, Türkmen Yayınevi, İstanbul.

Karagülle, İ. ve Pala, Z. 2001. Borland Delphi ile Veri Tabanı, Türkmen Yayınevi, İstanbul.

Lautier, A., Dehe, T., Gaillard, D., In line measurements of blood gases during ECC-fluorescence vs. absorptiometry Engineering in Medicine and Biology Society. 1988.

Proceedings of the Annual International Conference of the
IEEE, 2; 829–830.

Saka, O. 2000. Sayısal Görüntü Depolama Sistemlerinin
Tıp Uygulamaları ve Eğitimindeki Önemi, Türkiye Bilişim
Derneği, Sayı 73, www.tbd.org.tr/sayi73_html/saglik_saka.htm

Turkinternet.com, 2002. Türkiye’de Teletıp Dönemi,
<http://turk.internet.com/haber/yazigoster.php3?yaziid=4267>

Yağımlı, M. ve Akar, F. 2000. Dephi 5 Görsel Program
Tasarımı, Beta Yayınları, İstanbul.

Yanık, M. 2000. Borland Delphi 5 Görsel Programlama ile
Tasarım, Beta Yayınları, İstanbul.

EKLER

Ek 1. Sunucu Yazılım Kodları

Ek 2. Tasarım Aşamasında Sistemin Görünüşü

EK 1. Sunucu Yazılım Kodları

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
Dialogs, StdCtrls, CPortCtl, CPort, dxCore, dxButton, ComCtrls, Grids,
DB, DBTables, ScktComp;
type
TForm1 = class(TForm)
ComPort1: TComPort;
dxButton1: TdxButton;
dxButton2: TdxButton;
dxButton3: TdxButton;
StatusBar1: TStatusBar;
dxButton4: TdxButton;
dxButton5: TdxButton;
dxButton6: TdxButton;
dxButton7: TdxButton;
Label2: TLabel;
Label3: TLabel;
StringGrid1: TStringGrid;
Memo1: TMemo;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
.
.
.
Label105: TLabel;
Label106: TLabel;
Label107: TLabel;
Label108: TLabel;
Table1: TTable;
Table1MODEL: TStringField;
Table1IID: TStringField;
Table1DATE: TDateField;
Table1TIME: TTimeField;
Table1SEQN: TStringField;
```

EK 1 (devam)

```
Table1ACCN: TStringField;
Table1ROOM: TStringField;
Table1SOURCE: TStringField;
Table1DEVICE: TStringField;
Table1P1: TStringField;
Table1P2: TStringField;
Table1P3: TStringField;
.
.
.

Table1P43: TStringField;
Table1P44: TStringField;
Memo2: TMemo;
ServerSocket1: TServerSocket;
Label1: TLabel;
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure dxButton1Click(Sender: TObject);
procedure dxButton2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure dxButton3Click(Sender: TObject);
procedure dxButton4Click(Sender: TObject);
procedure ServerSocket1Accept(Sender: TObject;
Socket: TCustomWinSocket);
procedure ServerSocket1ClientConnect(Sender: TObject;
Socket: TCustomWinSocket);
procedure dxButton5Click(Sender: TObject);
procedure ServerSocket1ClientRead(Sender: TObject;
Socket: TCustomWinSocket);
procedure ServerSocket1ClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
procedure FormActivate(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
```

EK 1 (devam)

```
Str2:string;
j,t:integer;
v:array[0..4] of string;
implementation
uses Unit2,Unit5;
{$R *.dfm}
procedure isleme(str2: string);
var
y,x,l,z,t,k,m,n,p:integer;
str5:array [0..700] of string;
str4,str6,str7,str8,str9,cont:string;
begin
for y:=0 to Length(str2) do
str5[y]:=Copy(str2,y,1);
x:=Length(str2);
str2:="";
str4:=Chr(28);
str7:=Chr(29);
str8:=Chr(30);
str9:=Chr(29)+Chr(29);
l:=0;
z:=2;
t:=0;
k:=0;
While l=0 do
begin
str6:=str6+str5[z];
z:=z+1;
if str5[z]=str4 then
l:=1;
end;
if str6='SMP_NEW_DATA' then
begin
Form1.Label2.Visible:=False;
Form1.Label3.Visible:=False;
.
.

Form1.Label106.Visible:=False;
```

EK 1 (devam)

```
Form1.Label107.Visible:=False;
Form1.Label108.Visible:=False;
l:=0;
z:=z+2;
str6:="";
While l=0 do
begin
str6:=str6+str5[z];
z:=z+1;
if (str5[z]=str4) or (str5[z]=str7) then
break;
end;
l:=0;
if Copy(str6,2,Length(str6)-1)='MOD' then
begin
Form1.Label2.Visible:=True;
Form1.Label2.Caption:=Copy(str6,2,Length(str6)-1)+'EL';
z:=z+1;
str6:="";
While l=0 do
begin
str6:=str6+str5[z];
z:=z+1;
if str5[z]=str7 then
break;
end;
form1.Label3.Visible:=true;
Form1.Label3.Caption:=Copy(str6,1,Length(str6));
end;
l:=0;
z:=z+1;
str6:="";
while z< x-5 do
begin
if str5[z]=str4 then
begin
z:=z+1;
While l=0 do
begin
```


EK 1 (devam)

```
str6:=str6+str5[z];
z:=z+1;
if ((str5[z]+str5[z+1])=str9)or((str5[z]+str5[z+1])=(Chr(23)+Chr(29)))or
((str5[z]+str5[z+1])=(Chr(23)+'Q')) then
begin
Form1.StringGrid1.Cells[k,t]:=str6;
k:=0;
t:=t+1;
str6:="";
break;
end;
if str5[z]=str7 then
begin
z:=z+1;
Form1.StringGrid1.Cells[k,t]:=Copy(str6,1,Length(str6));
k:=k+1;
str6:="";
end;
end;
end;
z:=z+1;
end;
end;
Form1.Memo2.Text:=Form1.Memo1.Text;
Form1.Memo1.Text:="";
//-----
m:=0;
n:=0;
for p:=0 to 60 do
begin
cont:=Copy(Form1.StringGrid1.Cells[n,p],2,Length(Form1.StringGrid1.Cells[n,p])-1);
if cont='DATE' then
begin
form1.Label48.Visible:=True;
form1.Label48.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='IID' then
begin
```

EK 1 (devam)

```
form1.Label94.Visible:=True;
form1.Label95.Visible:=True;
form1.Label95.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='TIME' then
begin
form1.Label49.Visible:=True;
form1.Label49.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='SEQ' then
begin
form1.Label96.Visible:=True;
form1.Label97.Visible:=True;
form1.Label97.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='ACC' then
begin
form1.Label98.Visible:=True;
form1.Label99.Visible:=True;
form1.Label99.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='ROOM' then
begin
form1.Label100.Visible:=True;
form1.Label103.Visible:=True;
form1.Label103.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='SOURCE' then
begin
form1.Label101.Visible:=True;
form1.Label104.Visible:=True;
form1.Label104.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
if cont='DEVICE' then
begin
form1.Label102.Visible:=True;
form1.Label105.Visible:=True;
form1.Label105.Caption:=Form1.StringGrid1.Cells[n+1,p];
end;
```

EK 1 (devam)

```
if cont='pH' then
begin
form1.Label37.Visible:=True;
form1.Label50.Visible:=True;
form1.Label50.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label37.Font.Color:=clRed;
Form1.Label50.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label37.Font.Color:=clBlue;
Form1.Label50.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label37.Font.Color:=clGreen;
Form1.Label50.Font.Color:=clGreen;
end;
end;
if cont='PCO2' then
begin
form1.Label36.Visible:=True;
form1.Label51.Visible:=True;
form1.Label51.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label36.Font.Color:=clRed;
Form1.Label51.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label36.Font.Color:=clBlue;
Form1.Label51.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
```

EK 1 (devam)

```
begin
Form1.Label36.Font.Color:=clGreen;
Form1.Label51.Font.Color:=clGreen;
end;
end;
if cont='PO2' then
begin
form1.Label38.Visible:=True;
form1.Label52.Visible:=True;
form1.Label52.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label38.Font.Color:=clRed;
Form1.Label52.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label38.Font.Color:=clBlue;
Form1.Label52.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label38.Font.Color:=clGreen;
Form1.Label52.Font.Color:=clGreen;
end;
end;
end;
if cont='BP' then
begin
form1.Label9.Visible:=True;
form1.Label53.Visible:=True;
form1.Label53.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label9.Font.Color:=clRed;
Form1.Label53.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
```

EK 1 (devam)

```
begin
Form1.Label9.Font.Color:=clBlue;
Form1.Label53.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label9.Font.Color:=clGreen;
Form1.Label53.Font.Color:=clGreen;
end;
end;
if cont='Flow' then
begin
form1.Label20.Visible:=True;
form1.Label54.Visible:=True;
form1.Label54.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label20.Font.Color:=clRed;
Form1.Label54.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label20.Font.Color:=clBlue;
Form1.Label54.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label20.Font.Color:=clGreen;
Form1.Label54.Font.Color:=clGreen;
end;
end;
if cont='TEMP' then
begin
form1.Label45.Visible:=True;
form1.Label55.Visible:=True;
form1.Label55.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
```

EK 1 (devam)

```
begin
Form1.Label45.Font.Color:=clRed;
Form1.Label55.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label45.Font.Color:=clBlue;
Form1.Label55.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label45.Font.Color:=clGreen;
Form1.Label55.Font.Color:=clGreen;
end;
end;
if cont='Na+' then
begin
form1.Label30.Visible:=True;
form1.Label56.Visible:=True;
form1.Label56.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label30.Font.Color:=clRed;
Form1.Label56.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label30.Font.Color:=clBlue;
Form1.Label56.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label30.Font.Color:=clGreen;
Form1.Label56.Font.Color:=clGreen;
end;
end;
if cont='K+' then
begin
```

EK 1 (devam)

```
form1.Label27.Visible:=True;
form1.Label57.Visible:=True;
form1.Label57.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label27.Font.Color:=clRed;
Form1.Label57.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label27.Font.Color:=clBlue;
Form1.Label57.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label27.Font.Color:=clGreen;
Form1.Label57.Font.Color:=clGreen;
end;
end;
if cont='C++' then
begin
form1.Label10.Visible:=True;
form1.Label58.Visible:=True;
form1.Label58.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label10.Font.Color:=clRed;
Form1.Label58.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label10.Font.Color:=clBlue;
Form1.Label58.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label10.Font.Color:=clGreen;
```

EK 1 (devam)

```
Form1.Label58.Font.Color:=clGreen;
end;
end;
if cont='Cl-' then
begin
form1.Label11.Visible:=True;
form1.Label59.Visible:=True;
form1.Label59.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label11.Font.Color:=clRed;
Form1.Label59.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label11.Font.Color:=clBlue;
Form1.Label59.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label11.Font.Color:=clGreen;
Form1.Label59.Font.Color:=clGreen;
end;
end;
if cont='H+' then
begin
form1.Label22.Visible:=True;
form1.Label60.Visible:=True;
form1.Label60.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label22.Font.Color:=clRed;
Form1.Label60.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label22.Font.Color:=clBlue;
```


EK 1 (devam)

```
Form1.Label60.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label22.Font.Color:=clGreen;
Form1.Label60.Font.Color:=clGreen;
end;
end;
if cont='Lactate' then
begin
form1.Label28.Visible:=True;
form1.Label61.Visible:=True;
form1.Label61.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label28.Font.Color:=clRed;
Form1.Label61.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label28.Font.Color:=clBlue;
Form1.Label61.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label28.Font.Color:=clGreen;
Form1.Label61.Font.Color:=clGreen;
end;
end;
if cont='Glucose' then
begin
form1.Label21.Visible:=True;
form1.Label62.Visible:=True;
form1.Label62.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label21.Font.Color:=clRed;
```

EK 1 (devam)

```
Form1.Label62.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label21.Font.Color:=clBlue;
Form1.Label62.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label21.Font.Color:=clGreen;
Form1.Label62.Font.Color:=clGreen;
end;
end;
if cont='tHb' then
begin
form1.Label46.Visible:=True;
form1.Label63.Visible:=True;
form1.Label63.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label46.Font.Color:=clRed;
Form1.Label63.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label46.Font.Color:=clBlue;
Form1.Label63.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label46.Font.Color:=clGreen;
Form1.Label63.Font.Color:=clGreen;
end;
end;
if cont='sO2' then
begin
form1.Label43.Visible:=True;
form1.Label64.Visible:=True;
```

EK 1 (devam)

```
form1.Label64.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label43.Font.Color:=clRed;
Form1.Label64.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label43.Font.Color:=clBlue;
Form1.Label64.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label43.Font.Color:=clGreen;
Form1.Label64.Font.Color:=clGreen;
end;
end;
if cont='O2Hb' then
begin
form1.Label33.Visible:=True;
form1.Label65.Visible:=True;
form1.Label65.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label33.Font.Color:=clRed;
Form1.Label65.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label33.Font.Color:=clBlue;
Form1.Label65.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label33.Font.Color:=clGreen;
Form1.Label65.Font.Color:=clGreen;
end;
```

EK 1 (devam)

```
end;
if cont='COHb' then
begin
form1.Label12.Visible:=True;
form1.Label66.Visible:=True;
form1.Label66.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label12.Font.Color:=clRed;
Form1.Label66.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label12.Font.Color:=clBlue;
Form1.Label66.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label12.Font.Color:=clGreen;
Form1.Label66.Font.Color:=clGreen;
end;
end;
if cont='MetHb' then
begin
form1.Label29.Visible:=True;
form1.Label67.Visible:=True;
form1.Label67.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label29.Font.Color:=clRed;
Form1.Label67.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label29.Font.Color:=clBlue;
Form1.Label67.Font.Color:=clBlue;
end;
end;
```

EK 1 (devam)

```
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label29.Font.Color:=clGreen;
Form1.Label67.Font.Color:=clGreen;
end;
end;
if cont='HHb' then
begin
form1.Label26.Visible:=True;
form1.Label68.Visible:=True;
form1.Label68.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label26.Font.Color:=clRed;
Form1.Label68.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label26.Font.Color:=clBlue;
Form1.Label68.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label26.Font.Color:=clGreen;
Form1.Label68.Font.Color:=clGreen;
end;
end;
if cont='ctO2(a)' then
begin
form1.Label14.Visible:=True;
form1.Label69.Visible:=True;
form1.Label69.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label14.Font.Color:=clRed;
Form1.Label69.Font.Color:=clRed;
end;
end;
```

EK 1 (devam)

```
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label14.Font.Color:=clBlue;
Form1.Label69.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label14.Font.Color:=clGreen;
Form1.Label69.Font.Color:=clGreen;
end;
end;
if cont='ctO2' then
begin
form1.Label13.Visible:=True;
form1.Label70.Visible:=True;
form1.Label70.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label13.Font.Color:=clRed;
Form1.Label70.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label13.Font.Color:=clBlue;
Form1.Label70.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label13.Font.Color:=clGreen;
Form1.Label70.Font.Color:=clGreen;
end;
end;
if cont='ctO2(a-v)' then
begin
form1.Label15.Visible:=True;
form1.Label71.Visible:=True;
form1.Label71.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
```

EK 1 (devam)

```
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label15.Font.Color:=clRed;
Form1.Label71.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label15.Font.Color:=clBlue;
Form1.Label71.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label15.Font.Color:=clGreen;
Form1.Label71.Font.Color:=clGreen;
end;
end;
if cont='ctO2[a-v]/a' then
begin
form1.Label16.Visible:=True;
form1.Label72.Visible:=True;
form1.Label72.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label16.Font.Color:=clRed;
Form1.Label72.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label16.Font.Color:=clBlue;
Form1.Label72.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label16.Font.Color:=clGreen;
Form1.Label72.Font.Color:=clGreen;
end;
end;
end;
if cont='ctO2(Hb)' then
```

EK 1 (devam)

```
begin
form1.Label17.Visible:=True;
form1.Label73.Visible:=True;
form1.Label73.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label17.Font.Color:=clRed;
Form1.Label73.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label17.Font.Color:=clBlue;
Form1.Label73.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label17.Font.Color:=clGreen;
Form1.Label73.Font.Color:=clGreen;
end;
end;
if cont='ctO2(v)' then
begin
form1.Label18.Visible:=True;
form1.Label74.Visible:=True;
form1.Label74.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label18.Font.Color:=clRed;
Form1.Label74.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label18.Font.Color:=clBlue;
Form1.Label74.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
```


EK 1 (devam)

```
Form1.Label18.Font.Color:=clGreen;
Form1.Label74.Font.Color:=clGreen;
end;
end;
if cont='HCO3act' then
begin
form1.Label23.Visible:=True;
form1.Label75.Visible:=True;
form1.Label75.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label23.Font.Color:=clRed;
Form1.Label75.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label23.Font.Color:=clBlue;
Form1.Label75.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label23.Font.Color:=clGreen;
Form1.Label75.Font.Color:=clGreen;
end;
end;
if cont='HCO3std' then
begin
form1.Label24.Visible:=True;
form1.Label76.Visible:=True;
form1.Label76.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label24.Font.Color:=clRed;
Form1.Label76.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
```

EK 1 (devam)

```
Form1.Label24.Font.Color:=clBlue;
Form1.Label76.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label24.Font.Color:=clGreen;
Form1.Label76.Font.Color:=clGreen;
end;
end;
if cont='tCO2' then
begin
form1.Label44.Visible:=True;
form1.Label77.Visible:=True;
form1.Label77.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label44.Font.Color:=clRed;
Form1.Label77.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label44.Font.Color:=clBlue;
Form1.Label77.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label44.Font.Color:=clGreen;
Form1.Label77.Font.Color:=clGreen;
end;
end;
if cont='BE(vt)' then
begin
form1.Label7.Visible:=True;
form1.Label78.Visible:=True;
form1.Label78.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
```

EK 1 (devam)

```
Form1.Label7.Font.Color:=clRed;
Form1.Label78.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label7.Font.Color:=clBlue;
Form1.Label78.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label7.Font.Color:=clGreen;
Form1.Label78.Font.Color:=clGreen;
end;
end;
if cont='BE(vv)' then
begin
form1.Label8.Visible:=True;
form1.Label79.Visible:=True;
form1.Label79.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label8.Font.Color:=clRed;
Form1.Label79.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label8.Font.Color:=clBlue;
Form1.Label79.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label8.Font.Color:=clGreen;
Form1.Label79.Font.Color:=clGreen;
end;
end;
if cont='Hct' then
begin
form1.Label25.Visible:=True;
```

EK 1 (devam)

```
form1.Label80.Visible:=True;
form1.Label80.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label25.Font.Color:=clRed;
Form1.Label80.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label25.Font.Color:=clBlue;
Form1.Label80.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label25.Font.Color:=clGreen;
Form1.Label80.Font.Color:=clGreen;
end;
end;
end;
.
.
.

if cont='RI' then
begin
form1.Label41.Visible:=True;
form1.Label92.Visible:=True;
form1.Label92.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label41.Font.Color:=clRed;
Form1.Label92.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label41.Font.Color:=clBlue;
Form1.Label92.Font.Color:=clBlue;
```

EK 1 (devam)

```
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label41.Font.Color:=clGreen;
Form1.Label92.Font.Color:=clGreen;
end;
end;
if cont='RR' then
begin
form1.Label42.Visible:=True;
form1.Label93.Visible:=True;
form1.Label93.Caption:=Form1.StringGrid1.Cells[n+1,p]+'
'+Form1.StringGrid1.Cells[n+2,p];
if Form1.StringGrid1.Cells[n+3,p]='H' then
begin
Form1.Label42.Font.Color:=clRed;
Form1.Label93.Font.Color:=clRed;
end;
if Form1.StringGrid1.Cells[n+3,p]='L' then
begin
Form1.Label42.Font.Color:=clBlue;
Form1.Label93.Font.Color:=clBlue;
end;
if Form1.StringGrid1.Cells[n+3,p]='QV' then
begin
Form1.Label42.Font.Color:=clGreen;
Form1.Label93.Font.Color:=clGreen;
end;
end;
form1.Label106.Visible:=True;
form1.Label107.Visible:=True;
form1.Label108.Visible:=True;
Form1.dxBUTTON3.Enabled:=True;
Form1.dxBUTTON5.Enabled:=True;
Form1.dxBUTTON7.Enabled:=True;
end;
//-----
end;
```

EK 1 (devam)

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  Str,Str2,b: String;
begin
  ComPort1.ReadStr(Str, Count);
  Str2:=Str2+Str;
  Memo1.Text:=Memo1.Text+Str;
  b:=Copy(Str,Length(Str),1);
  if b=CHR(4) then
  begin
    Label2.Visible:=False;
    Label3.Visible:=False;
    .
    .

    Label106.Visible:=False;
    Label107.Visible:=False;
    Label108.Visible:=False;
    StatusBar1.Panels.Items[0].Text:='Sonuçlar Alındı';
    isleme(Memo1.Text);
  end;
end;
procedure TForm1.dxBUTTON1Click(Sender: TObject);
var
  i,l,top,m:integer;
  v,s,er:string;
  c:array[0..2000] of WideChar;
begin
  //if ComPort1.Connected then
  //begin
  //  ComPort1.Close;
  //  StatusBar1.Panels.Items[0].Text:='Haberleşme Kapalı';
  //end
  // else
  //begin
  //  ComPort1.Open;
  //  StatusBar1.Panels.Items[0].Text:='Sonuçlar Bekleniyor..!';
  //end;
  top:=0;
```

EK 1 (devam)

```
Memo1.Lines.LoadFromFile('C:\hastalar\2.txt');
v:=Memo1.Text;
StringToWideChar(v, c, Length(v)+1);
for l:=0 to Length(v)-4 do
begin
i:=Ord(c[l]);
top:=top+i;
end;
//Edit1.Text:=inttostr(top);
m:=top Mod 256;
//Edit2.Text:=inttostr(m);
er:=IntToHex(m,2);
s:=copy(v,Length(v)-2,2);
//Edit3.Text:=s;
if er=s then
begin
StatusBar1.Panels.Items[0].Text:='Veriler Doğru Geldi..';
end
else
begin
ShowMessage('Veriler Hatalı Geldi! Lütfen Tekrar Gönderin.');
```

```
end;
isleme(Memo1.Text);
end;
procedure TForm1.dxBUTTON2Click(Sender: TObject);
begin
Halt;
end;
procedure TForm1.FormCreate(Sender: TObject);
var
Buf : array[0..256] of char;
T : TSearchRec;
begin
//Session.NetFileDir := ExtractFilePath(Application.ExeName) + 'network\';
// GetTempPath(255, Buf);
// Session.PrivateDir := Buf; Session.Open;
Label2.Visible:=False;
Label3.Visible:=False;
Label4.Visible:=False;
```

EK 1 (devam).

```
Label105.Visible:=False;
Label106.Visible:=False;
Label107.Visible:=False;
Label108.Visible:=False;
dxbutton3.Enabled:=False;
dxbutton5.Enabled:=False;
dxbutton7.Enabled:=False;
//ServerSocket1.Active:=True;
end;
procedure TForm1.dxBUTTON3Click(Sender: TObject);
var
c:word;
date,da:String;
begin
c:=Application.MessageBox('Kayıt Yapılsın mı?','UYARI',36);
if c=IDYES then
begin
if ComPort1.Connected then
ComPort1.Close;
Table1.Open;
Table1.Append;
Table1MODEL.Text:=Label3.Caption;
Table1IID.Text:=Label95.Caption;
Table1TIME.Value:=StrtoTime(Label49.Caption);
da:=Copy(Label48.Caption,0,2);
date:=da;
da:=Copy(Label48.Caption,3,3);
if da='Jan' then
da:='01';
if da='Feb' then
da:='02';
if da='Mar' then
da:='03';
if da='Apr' then
da:='04';
if da='May' then
da:='05';
if da='Jun' then
```


EK 1 (devam)

```
da:='06';
if da='Jul' then
da:='07';
if da='Aug' then
da:='08';
if da='Sep' then
da:='09';
if da='Oct' then
da:='10';
if da='Nov' then
da:='11';
if da='Dec' then
da:='12';
date:=date+'.'+da+'.';
da:=Copy(Label48.Caption,6,2);
date:=date+'20'+da;
Table1DATE.Value:=StrToDate(date);
if label96.Visible=True then
Table1SEQN.Text:=Label97.Caption;
if label98.Visible=True then
Table1ACCN.Text:=Label99.Caption;
if label100.Visible=True then
Table1ROOM.Text:=Label103.Caption;
if label101.Visible=True then
Table1SOURCE.Text:=Label104.Caption;
if label102.Visible=True then
Table1DEVICE.Text:=Label105.Caption;
if label137.Visible=True then
Table1P1.Text:=Label50.Caption;
if label136.Visible=True then
Table1P2.Text:=Label51.Caption;
if label138.Visible=True then
Table1P3.Text:=Label52.Caption;
if label9.Visible=True then
Table1P4.Text:=Label53.Caption;
if label20.Visible=True then
Table1P5.Text:=Label54.Caption;
if label45.Visible=True then
Table1P6.Text:=Label55.Caption;
```

EK 1 (devam)

if label30.Visible=True then
Table1P7.Text:=Label56.Caption;
if label27.Visible=True then
Table1P8.Text:=Label57.Caption;
if label10.Visible=True then
Table1P9.Text:=Label58.Caption;
if label11.Visible=True then
Table1P10.Text:=Label59.Caption;
if label22.Visible=True then
Table1P11.Text:=Label60.Caption;
if label28.Visible=True then
Table1P12.Text:=Label61.Caption;
if label21.Visible=True then
Table1P13.Text:=Label62.Caption;
if label46.Visible=True then
Table1P14.Text:=Label63.Caption;
if label43.Visible=True then
Table1P15.Text:=Label64.Caption;
if label33.Visible=True then
Table1P16.Text:=Label65.Caption;
if label12.Visible=True then
Table1P17.Text:=Label66.Caption;
if label29.Visible=True then
Table1P18.Text:=Label67.Caption;
if label26.Visible=True then
Table1P19.Text:=Label68.Caption;
if label14.Visible=True then
Table1P20.Text:=Label69.Caption;
if label13.Visible=True then
Table1P21.Text:=Label70.Caption;
if label15.Visible=True then
Table1P22.Text:=Label71.Caption;
if label16.Visible=True then
Table1P23.Text:=Label72.Caption;
if label17.Visible=True then
Table1P24.Text:=Label73.Caption;
if label18.Visible=True then
Table1P25.Text:=Label74.Caption;
if label23.Visible=True then

EK 1 (devam)

Table1P26.Text:=Label75.Caption;
if label24.Visible=True then
Table1P27.Text:=Label76.Caption;
if label44.Visible=True then
Table1P28.Text:=Label77.Caption;
if label7.Visible=True then
Table1P29.Text:=Label78.Caption;
if label8.Visible=True then
Table1P30.Text:=Label79.Caption;
if label25.Visible=True then
Table1P31.Text:=Label80.Caption;
if label31.Visible=True then
Table1P32.Text:=Label81.Caption;
if label35.Visible=True then
Table1P33.Text:=Label82.Caption;
if label19.Visible=True then
Table1P34.Text:=Label83.Caption;
if label32.Visible=True then
Table1P35.Text:=Label84.Caption;
if label34.Visible=True then
Table1P36.Text:=Label85.Caption;
if label4.Visible=True then
Table1P37.Text:=Label86.Caption;
if label5.Visible=True then
Table1P38.Text:=Label87.Caption;
if label47.Visible=True then
Table1P39.Text:=Label88.Caption;
if label6.Visible=True then
Table1P40.Text:=Label89.Caption;
if label39.Visible=True then
Table1P41.Text:=Label90.Caption;
if label40.Visible=True then
Table1P42.Text:=Label91.Caption;
if label41.Visible=True then
Table1P43.Text:=Label92.Caption;
if label42.Visible=True then
Table1P44.Text:=Label93.Caption;
Table1.UpdateRecord;
Table1.Refresh;

EK 1 (devam)

```
Table1.Close;
end;
if ComPort1.Connected then
begin end
else begin
ComPort1.Open;
StatusBar1.Panels.Items[0].Text:='Sonuçlar Bekleniyor..';
end; end;
procedure TForm1.dxBUTTON4Click(Sender: TObject);
begin
Application.CreateForm(TForm2, Form2);
Form2.Show;
end;
procedure TForm1.ServerSocket1Accept(Sender: TObject;
Socket: TCustomWinSocket);
begin
StatusBar1.Panels.Items[1].Text:='Socket.RemoteHost+'   bağlantı   isteği
kabul edildi.';
j:=1; end;
procedure TForm1.ServerSocket1ClientConnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
StatusBar1.Panels.Items[1].Text:='Socket.RemoteHost+'   bağlantı   isteği
geldi.';
end;
procedure TForm1.dxBUTTON5Click(Sender: TObject);
var
p:integer;
begin
t:=ServerSocket1.Socket.ActiveConnections;
if not (t=0) then
begin
if ComPort1.Connected then
begin
ComPort1.Close;
StatusBar1.Panels.Items[0].Text:='Haberleşme Kapalı';
end; for p:=0 to t-1 do
begin
v[p]:=ServerSocket1.Socket.Connections[p].RemoteHost;
```

EK 1 (devam)

```
end;
Application.CreateForm(TForm5, Form5);
for p:=0 to t-1 do
begin
if v[p]='SEK' then
Form5.RadioButton1.Enabled:=True;
end;
Form5.Show;
end; end;
procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
Socket: TCustomWinSocket);
begin
StatusBar1.Panels.Items[1].Text:=Socket.ReceiveText;
end;
procedure TForm1.ServerSocket1ClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
StatusBar1.Panels.Items[1].Text:=Socket.RemoteHost+'den          bağlantı
kesildi.';
j:=0; end;
procedure TForm1.FormActivate(Sender: TObject);
var
p:integer;
begin
for p:=0 to t-1 do
begin
if v[p]=Label1.Caption then
ServerSocket1.Socket.Connections[p].SendText(Memo2.Text);
end;
Label1.Caption:="";
//if not (ComPort1.Connected) then
//begin
//ComPort1.Open;
//StatusBar1.Panels.Items[0].Text:='Sonuçlar Bekleniyor..!';
//end;
end;
end.
```

EK 2. Tasarım Aşamasında Sistemin Görünüşü

Sunucu Bilgisayar ve Kan Gazı Analiz Cihazı



EK 2 (devam)

Kan Gazı Analiz Cihazı RS232 Portu ile Sunucu Bilgisayar Seri Portu Bağlantısı



EK 2 (devam)

Kan Gazı Analiz Cihazı Portları



ÖZGEÇMİŞ

1977 yılında Hatay'ın Dörtöyol ilçesinde doğdu. İlkokulu Dörtöyol'da, orta ve lise eğitimini Ceyhan'da tamamladı. 1996 yılında girdiği Ankara Üniversitesi Fen Fakültesi Elektronik Mühendisliği Bölümü'nden 2000 yılında mezun oldu.

Ekim 2000 - Kasım 2003 yılları arasında Türkiye Yüksek İhtisas Hastanesi Tıbbi Aygıtlar Bakım Onarım Merkezinde Elektronik Mühendisi olarak görev yaptı. Kasım 2003 tarihinden bu yana Türk Telekomünikasyon A.Ş. Uydur Haberleşme ve Yönetim Dairesi Başkanlığı'nda Telekom Uzman Yardımcısı ünvanı ile görev almaktadır.