UNIVERSITY OF ÇUKUROVA
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Tuncay ALTUN

SIP PHONE APPLICATION ON SINGLE BOARD COMPUTER WITH ARM MICROPROCESSOR

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2010

UNIVERSITY OF ÇUKUROVA

INSTITUTE OF NATURAL AND APPLIED SCIENCES

SIP PHONE APPLICATION ON SINGLE BOARD COMPUTER WITH ARM MICROPROCESSOR

By Tuncay ALTUN

MSc THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certified that the thesis titled above was reviewed and approved for the award of degree of the Master of Electrical and Electronics Engineering by the board of jury on 02/02/2010.

Signature……………	Signature……………	Signature……………
Assist.Prof.Dr. Murat AKSOY	Assist.Prof.Dr. Ulus ÇEVİK	Assist.Prof.Dr. Mutlu AVCI
Supervisor	Member	Member

This MSc Thesis is performed in Department of Electrical and Electronics Engineering of Çukurova University.

Registration Number:

Prof. Dr. İlhami YEĞİNGİL
Director
The Institute of Natural and Applied Sciences

ÖZ

YÜKSEK LİSANS TEZİ

ARM MİKROİŞLEMCİLİ BİR MİNİ BİLGİSAYARDA SIP TELEFON UYGULAMASI

Tuncay ALTUN

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Yrd.Doç.Dr. Murat AKSOY
Yıl: 2010, Sayfa: 76

Jüri : Yrd.Doç.Dr. Murat AKSOY
Yrd.Doç.Dr. Ulus ÇEVİK
Yrd.Doç.Dr. Mutlu AVCI

Bu Yüksek Lisans Tezinde günlük hayatımızda kullandığımız multimedyalı el aletleri üzerinde çalışılmıştır. Bu aletlerle resim ve video izleme, internette sörf ve alışveriş gibi işlemler yapılabilmesi, kullanımını arttırmaktadır. Çalışmada bunların yanı sıra internet ağı üzerinden SIP protokolü kullanarak VoIP adı verilen sesli görüşme de yapabilmesi üzerinde çalışılmıştır.

Uygulama olarak 32-bit mikroişlemci S3C2440, 64 MB Flash ROM, 64 MB SDRAM bellek kullanılmıştır. Klavye, mouse, bluetooth, flash bellek gibi çevre birimlerini kullanabilmek için bir USB kontrolör, internet bağlantısı için bir LAN kontrolör ve 4,3” büyüklüğünde dokunmatik LCD ekran kullanılmıştır. Bu donanımların yanı sıra depolanabilir veri alanını arttırmak için hafıza kartı eklenmiştir. İnternet üzerinden sesli görüşmelerin rahat yapılabilmesi için mini bilgisayar içerisine ses giriş-çıkış üniteleri monte edilmiştir.

Çalışmalar Microsoft firmasının Platform Builder yazılımı kullanılarak Windows CE 5.0 işletim sistemi üzerinde yapılmıştır. Microsoft Embedded Visual C++ yazılımı kullanılarak tüm donanımlara uygun sürücüler hazırlanmıştır.

Yapılan çalışma sonucunda gerekli donanımlar çalıştırılmış olup hem istenilen multimedya özellikleri kullanılmış hem de internet üzerinden sesli görüşme yapılabilmiştir.

Anahtar Kelimeler : 32-bit mikroişlemci, SBC, SIP, VoIP

ABSTRACT

MSc THESIS

SIP PHONE APPLICATION ON SINGLE BOARD COMPUTER WITH ARM MICROPROCESSOR

Tuncay ALTUN

UNIVERSITY OF ÇUKUROVA
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Asst.Prof. Dr. Murat AKSOY
Year: 2010, Pages: 76

Jury : Assist. Prof. Dr. Murat AKSOY
Assist. Prof. Dr. Ulus ÇEVİK
Assist. Prof. Dr. Mutlu AVCI

In this Master Thesis, the multimedia hand tools that we use in our everyday life have been studied. The applications such as image and video monitoring, internet surfing and shopping with these devices increase their usage. In addition to the applications mentioned above, a VoIP called voice communication technique, using SIP over the internet, has been investigated.

In this application 32-bit microprocessor S3C2440, 64 MB Flash ROM, 64 MB SDRAM memory is utilised. For the connection of the peripherals such as the keyboard, mouse, bluetooth a USB controller is used and for the internet connection a LAN controller and a 4,3” touch-screen LCD display utilised. In addition to the hardware a listed above, a memory card is added to expand the data storage. In order to implement the voice communications conveniently over the internet, voice input&output units are added in the mini-computer.

This work has been done over Windows CE 5.0 operating system using the Microsoft Platform Builder. The drivers are prepared for all hardware on board using the Microsoft Embedded Visual C++.

As a result of the study, all needed hardware was run and both the multimedia features and the voice calls are implemented.

Key Words: 32-bit microprocessor, SBC ,SIP, VoIP

ACKNOWLEDGEMENT

First of all I would like to express my appreciation to my lover wife Mrs. Betül, my unique son Toygar Tuncay, my parents Mr. Mustafa and Mrs. Nimet ALTUN for their unmatched encouragement, good tender and partnership.

I dedicate this project to them with all my heart.

I wish also to express a great scholar to Murat AKSOY, a truthful man and a real scientist. I learn very important things about the science moreover about the life.

I would like to thanks all participants of the Powarm S3C2440 Development Board Project specially to Erdal and Göktuğ.

CONTENTS **PAGE**

TABLE LIST

PAGE

FIGURE LIST

PAGE

SYMBOLS AND ABBREVIATIONS

SBC : Single Board Computer

VoIP : Voice Over IP

SIP : Session Initiation Protocol

RTP : Real-Time Protocol

SDP : Session Description Protocol

TCP : Transport Control Protocol

UDP : User Datagram Protocol

PCM : Pulse Code Modulation

GSM : Global System for Mobile

PSTN : The Public Switched Telephone Network

TDM : Time-Division Multiplexing

NCP : Network Control Protocol

PPP : Point-to-Point Protocol

SNMP : Simple Network Management Protocol

SMTP : Simple Mail Transfer Protocol

IETF : Internet Engineering Task Force

ITU : International Telecommunications Union

PDA : Personal Data Assistant

AMBA: Advanced Micro controller Bus Architecture

CAS : Column Address Strobe

TSOP : Thin Small Outline Package

SSOP : Shrink Small Outline Package

AGC : Automatic Gain Control

DSP : Digital Sound Processing

GPIO : General Purpose I/O

ISA : Industry Standard Architecture

LQPF : Low-profile Quad Flat Package

TFT : Thin Film Transistor

1.INTRODUCTION

1.1. Voice over IP

Voice over IP protocols are used to transport a voice signal over the IP network. This allows to use IP Telephony instead of the dedicated voice transmission telephony lines. With this technology the signal is digitally carried over the Internet, instead of over analog voice transmission lines. When the user dials a VoIP number, signaling is required to determine the status of the called party and to establish the call. Next, when the conversation starts, the analog signal produced by the microphone needs to be encoded in a digital format appropriate for transmission across an IP network. The IP network itself must then ensure that the realtime conversation is transported across the available media in a method that produces acceptable voice quality. Finally, it may be necessary for the IP telephony stream to be converted by a gateway to another format (Hardy, 2003).

1.1.1. Motivation Factors

So the motivating factors that make the case for an enterprise move to an IP infrastructure are:

- Cost Savings
- Productivity Gains
- Customer Intimacy
- Applications

1.1.1.1. Cost Savings

First, a widely held belief exists that significant cost savings can be realized through the conversion from a traditional circuit-switched infrastructure to an IP-packet infrastructure. And although the outlay of real capital may not be as greatly reduced as many would like to think, the savings do come, and in a variety of ways.

In a network that comprises multiple locations, packet switching reduces the cost of voice transmission. Furthermore, when voice migrates to IP, it becomes one more data application and now has the ability ride the same network as the enterprise's data traffic. The result is far more efficient network and traffic management. After all, technical personnel now manage one network, not two. They also now combine voice-based network changes with those of data activity, which results in a measurable cost reduction. One area where VoIP results in significant savings is in the area of staffing and the need to physically locate staff members in specific locations. Because IP telephony is location independent and does not require collocation with the local switch or PBX, the resources of the network and support personnel can be located wherever it makes sense to put them, regardless of the location of the network itself (Hardy, 2003).

1.1.1.2. Productivity Gains

VoIP systems today typically include a range of productivity tools such as voicemail, e-mail, fax forwarding, and embedded directory services. And by taking advantage of so-called presence applications, a caller can tell whether the called party is available and can direct the system to track them down through a variety of contact modalities such as follow-me service to a mobile phone, VoIP running on a PDA, and so on. This capability, implemented through the Session Initiation Protocol (SIP) eliminates the need for the customer to "track down" the called party. Instead, the system does so using a rule set created by the called party and based on whether the person is in a position to be contacted, and if so, the best way to do so. If a caller were to place a call to a person in a business and the person is not at his or her desk, the called party may invoke a rule set that uses the following routing information:

1- Call received by main number at desk.

2- If no answer after three rings, redirect call to mobile number.

3- If no answer after five rings, redirect call to VoIP account on PDA.

4- If no answer, send to voicemail with directions to send a SMS message to a specific identity (Shepard, 2005).

The result is that the customer needs a single number to make contact, and the system takes care of intrasystem routing, taking whatever steps it needs to establish contact between the caller and the called party.

1.1.1.3. Customer Intimacy

Customer relations are what business is all about, and management of those relationships is what makes one company better than its closest competitor. Customer relationship management is one of the most fundamental and important activities that companies can engage in. And IP-based communications systems have the ability to enhance this function dramatically (Shepard, 2005)..

Automatic call routing, mentioned earlier, is one way that companies improve their customer affinity practices; another is through the critical analysis of customer behavior: buying patterns, calling patterns, how they use purchased products, and so on. Because VoIP systems have the ability to generate detailed reports that can be analyzed to yield this kind of information, managers can take action based on the reported data to reduce hold times, provide online information, and, therefore, reduce the total number of dropped calls. Interactive voice response units give the called party the ability to manage incoming calls by routing them to the shortest queue, the operator most informed about particular geography, a specific language requirement, a product expert, or the same person he or she spoke with on a prior call. Similarly, callers who choose to use e-mail or a Web chat as their preferred mode of communications can do so and stil enjoy the proper degree of appropriate routing (Camarillo,2002).

1.1.1.4. Applications

There will unquestionably be some degree of cost savings, but the real advantage, the culminating event that makes a VoIP conversion so powerful, is the convergence of applications. One of the best known of these is “click-though” or “click-to-call” a technique that allows a user to seamlessly move from an IM or SMS into a voice session with a single click. Already the major software houses (PeopleSoft, Siebel, Oracle, Microsoft) are developing applications that will integrate seamlessly into the voice environment, allowing for interactive voice-enabled database queries (Shepard, 2005).

1.1.2. CODEC Standards

The word CODEC is a contraction of the two words “coder” and “decoder,” the two software functions required for voice digitization and in some cases compression. Coding is the process of analog-to-digital conversion; decoding is the opposite, clearly a set of functions that sits between analog and digital systems such as the analog access to the PSTN and an IP-based network. Many CODECs are available, and CODEC selection is typically dependent on choices related to voice quality and processing speed. Figure 1.1 lists the most commonly used CODECs and their characteristics. As an example, in a situation where call quality is of the utmost importance such as in a customer support center, a G.711 CODEC would be an ideal selection because of its extremely high quality characteristics. This is the voice digitization standard used throughout the PSTN to create eight-bit pulse code modulation (PCM) samples at the rate of 8,000 samples per second. The G.711 CODEC creates a 64 Kbps bitstream and has two forms. A-Law G.711 PCM converts 13-bit PCM samples into eight-bit compressed PCM samples. m-Law (Mu-Law) PCM converts 14-bit PCM samples into eight-bit compressed PCM samples. A-Law is used throughout most countries in the world; m-Law is used in the United States, Canada, and a handful of other countries. On the other hand, in situations where voice across the WAN will be used as a way to reduce the cost of long distance, a G.729a CODEC may be perfectly adequate, because voicemail may be the most common form of traffic encoded using the CODEC. Today, vendors perform most of the leg work associated with CODEC selection, choosing the one that best meets the needs of the application they will serve with their installation (Hardy, 2005).

Table 1.1. The most commonly used CODECs

CODEC	Data Rate	Voice Quality
G.711	64 Kbps	High
G.723.1	6.4, 5.3 Kbps	Low
G.726	40, 32, 24, 16 Kbps	Medium
G.728	16 Kbps	Medium
G.729	8 Kbps	Medium

Special cases must be considered during the selection process, however, one of which is known as transcoding. If two or more parties are communicating using different CODECs, such as a cellular user communicating with a VoIP device, the network must translate (or transcode) between the two disparate coding schemes. For example, if the mobile user is talking on a Global System for Mobile communications (GSM) phone, the PSTN will have to translate between G.729a and the full rate or half-rate supported by GSM networks. Each conversion introduces delay and the possibility of introducing noise into the system, which can result in a perceptible degradation in QoS (Shepard, 2005 and Camarillo,2002).

1.1.3. The Public Switched Telephone Network (PSTN)

By understanding the PSTN is gained a better understanding and awareness of exactly what the customer will expect from a voice over IP (VoIP) system. When a customer makes a telephone call, a complex sequence of happenings takes place that finally leads to the creation of a temporary end-to-end connection (Camarillo,2002).

When a caller picks up the telephone, the act of lifting the handset closes a circuit, which allows current to flow from the switch in the local central office that serves the telephone. The switch electronically attaches an oscillator to the circuit called a dial tone generator, which creates the customary sound that we all listen for when we place a call. The dial tone serves to notify the caller that the switch is ready to receive the dialed digits (Camarillo,2002).

The caller now dials the desired telephone number by pressing the appropriate buttons on the phone. Each button generates a pair of tones that are slightly dissonant. This is done to prevent the possibility of a human voice randomly generating a frequency that could cause a misdial. The tone pairs are carefully selected so that they cannot be naturally generated by the human voice. This technique is called Dual Tone Multi Frequency (DTMF) (Camarillo,2002).

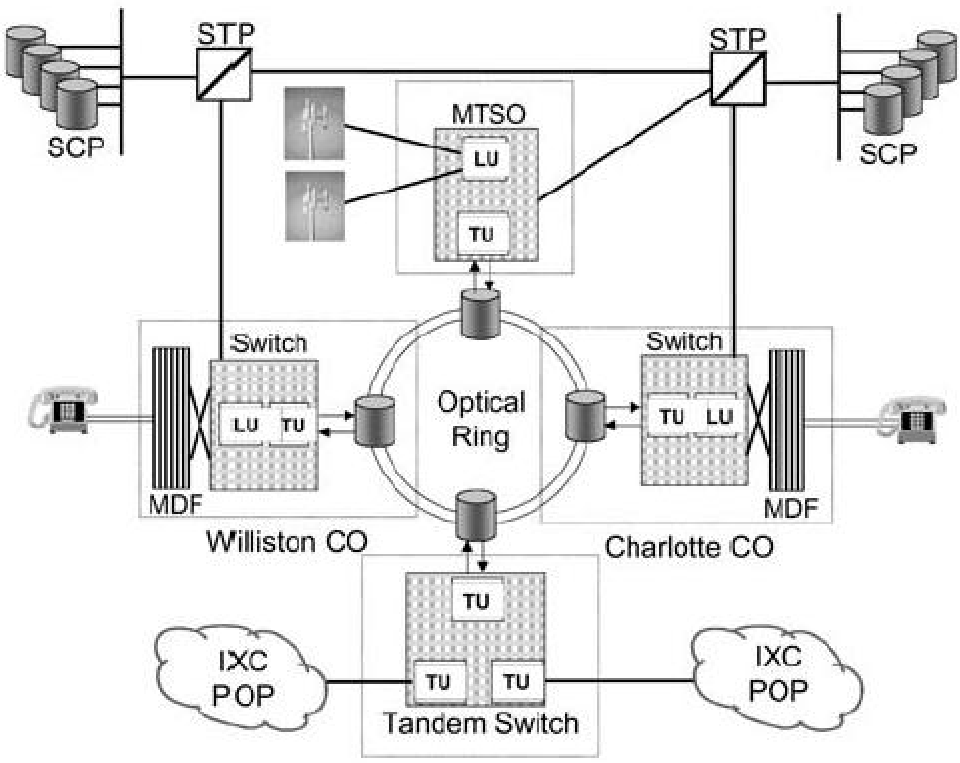

Figure 1.1. Typical a PSTN Network (Shepard, 2005).

There is a switch on many phones with two positions labeled “TONE” and “PULSE.” When the switch is set to the tone position, the phone generates DTMF. When it is set on pulse, it generates the series of clicks that old dial telephones made when they were used to place a call. When the dial was rotated it caused a switch contact to open and close rapidly three times, sending a series of three electrical pulses or more correctly interruptions to the switch. DTMF has been around since the 1970s, but switches are still capable (Shepard, 2005).

1.1.4. Voice Digitization

The process of converting analog voice to a digital representation in the modern network is a logical and plainly process. It comprises four different steps: *pulse amplitude modulation (PAM)* sampling, in which the amplitude of the incoming analog wave is sampled every 125 microseconds; *companding*, during which the values are weighted toward those most receptive to the human ear; *quantization*, in which the weighted samples are given values on a nonlinear scale; and finally *encoding*, during which each value is assigned a distinct binary value(Shepard, 2005).

The analog signal must be sampled at a rate that is equal to twice the bandwidth of the channel over which the signal is to be transmitted. Because each analog voice channel is allocated 4 KHz of bandwidth, it follows that each voice signal must be sampled at twice that rate, or 8,000 samples per second. The standard multiplexer accepts inputs from 24 analog channels as shown in Figure 1.2 .

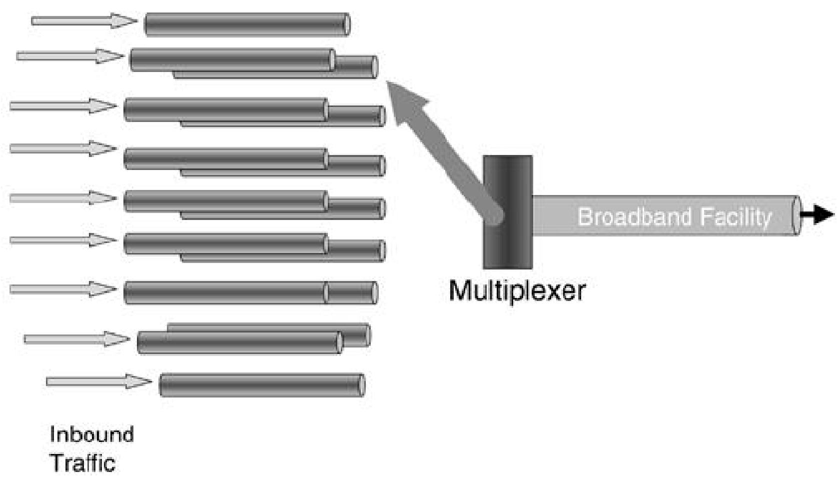

Figure 1.2. Time-division multiplexing (TDM) (Shepard, 2005).

Each channel is sampled in turn, every one eight-thousandth of a second, resulting in the generation of 8,000 pulse amplitude samples from each channel every second. The sampling rate is important. If the sampling rate is too high, too much information is transmitted and bandwidth is wasted; if the sampling rate is too low, then we run the risk of aliasing. Aliasing is the interpretation of the sample points as a false waveform, due to the paucity of samples (Hardy, 2003).

Although PCM is perhaps the best-known, high-quality voice digitization process, it is by no means the only one. Advances in coding schemes and improvements in the overall quality of the telephone network have made it possible for encoding schemes to be developed that use far less bandwidth than do traditional PCM (Hardy, 2003).

1.1.5. IP History

The Internet as we know it today began as a Department of Defense (DoD) project designed to interconnect DoD research sites. In December 1968, the government research agency known as the Advanced Research Projects Agency (ARPA) awarded a contract to Bolt Beranek and Newman to design and build the packet network that would ultimately become the Internet. It had a proposed transmission speed of 50 Kbps, and in September 1969 the first node was installed. Other nodes were installed on roughly a monthly basis at Stanford Research Institute, the University of California at Santa Barbara, and the University of Utah. The ARPANET spanned the continental United States by 1971 and had connections to research facilities in Europe by 1973. The original protocol selected for the ARPANET was called the Network Control Protocol (NCP). It was designed to handle the emergent requirements of the low-volume architecture of the ARPANET network. As traffic grew, however, it proved to be inadequate to handle the load, and in 1974 a more robust protocol suite was implemented. This new Transmission Control Protocol (TCP) was an ironclad protocol designed for end-to-end network communications control (Hardy, 2003).

In 1978, a new design split the responsibilities for end-to-end versus node-to-node transmission among two protocols. The newly crafted IP was designed to route packets from device-to-device, and TCP was designed to offer reliable, end-toend communications. Because TCP and IP were originally envisioned as a single protocol, they are now known as the TCP/IP protocol suite, a name that also incorporates a collection of protocols and applications that also handle routing, QoS, error control, and other functions (Hardy, 2003).

In 1983, the ARPANET was split into two networks. One half, still called ARPANET, continued to be used to interconnect research and academic sites. The other, called MILNET, was specifically used to carry military traffic and ultimately became part of the Defense Data Network. That year was also a good year for TCP/IP. It was included as part of the communications kernel for the University of California’s UNIX implementation, known as 4.2BSD (or Berkeley Software Distribution) UNIX. Extension of the original ARPANET continued (Hardy, 2003).

In 1986, the National Science Foundation (NSF) built a backbone network to interconnect four NSF supercomputing centers and the National Center for Atmospheric Research. This network, known as NSFNET, was originally intended to serve as a backbone for other networks, not as a stand-alone interconnection mechanism. Additionally, the NSF’s Appropriate Use Policy limited transported traffic to noncommercial traffic only. NSFNET continued to expand and eventually became what we know today as the Internet. And although the original NSFNET applications were multiprotocol implementations, TCP/IP was used for overall interconnectivity (Hardy, 2003).

In 1994, a structure was put in place to reduce the NSF’s overall role on the Internet. The new structure consists of three principal components. The first of these was a small number of network access points (NAPs) where Internet service providers (ISPs) would interconnect to the Internet backbone. The NSF originally funded four NAPs in Chicago (operated by Ameritech, now part of SBC), New York (really Pensauken, New Jersey, operated by Sprint), San Francisco (operated by Pacific Bell, now part of SBC), and Washington, D.C. (MAE-East, operated by MFS, now a division of MCI). The second component was the very high-speed backbone network service, a network that interconnected the NAPs and was operated by MCI. It was installed in 1995 and originally operated at OC-3 (155.52 Mbps), but was upgraded to OC-12 (622.08 Mbps) in 1997. The third component was the routing arbiter, designed to ensure that appropriate routing protocols for the Internet were available and properly deployed. ISPs were given 5 years of diminishing funding to become commercially self-sustaining (Shepard, 2005).

The funding ended in 1998. Starting at roughly the same time, a significant number of additional NAPs have been launched. As a matter of control and management, three tiers of ISP have been identified. Tier 1 refers to ISPs that have a national presence and connect to at least three of the original four NAPs. National ISPs include AT&T, Cable & Wireless, MCI, and Sprint. Tier 2 refers to ISPs that have a primarily regional presence and connect to less than three of the original four NAPs. Regional ISPs include Adelphia.net, Verizon.net, and BellSouth.net. Finally, Tier 3 refers to local ISPs, or those that do not connect to a NAP but offer services via the connections of another ISP(Shepard, 2005).

1.1.6. The TCP/IP Protocol

TCP/IP has been around for a long time, and although we often tend to think of it as a single protocol that governs the Internet, it is in reality a fairly exhaustive collection of protocols that cover the functions of numerous layers of the protocol stack (Hardy, 2003).

TCP/IP was created for the Internet with the concept in mind that the Internet would not be a particularly well behaved network. In other words, designers of the protocol made the assumption that the Internet would become precisely what it has become—a network of networks. It uses a plethora of unrelated and often conflicting protocols, and transports traffic with widely varying QoS requirements. The fundamental building block of the protocol, the IP packet, is designed to deal with all of these disparities, whereas TCP (and other related protocols, discussed later) take care of the QoS issues. Two network interface protocols are particularly important to TCP/IP. The Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP) are used to provide data-link layer services in situations where no other data-link protocol is present, such as in leased-line or older dial-up environments. Most TCP/IP software packages for desktop applications include these two protocols, even though dial-up is rapidly fading into near-oblivion in the presence of growing levels of broadband access.With SLIP or PPP, a remote computer can attach directly to a host and connect to the Internet using IP rather than being limited to an asynchronous connection (Hardy, 2003).

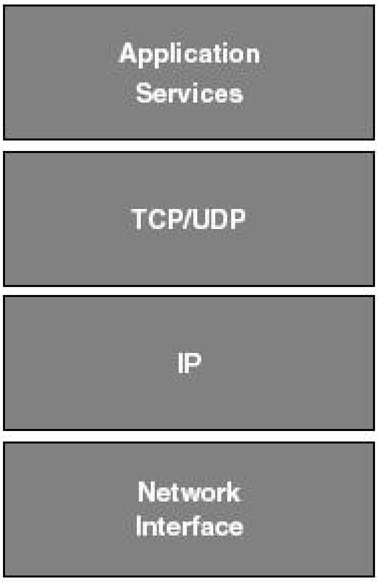

Figure 1.3. Simplified TCP/IP protocol stack

The Total Length field indicates the length of the entire packet, including both the header and the data within the packet. The maximum size of an IP packet is 64 KB, 65,535 bytes. IP addresses are 32 bits long. They are typically written as a sequence of four numbers, which represent the decimal value of each of the address bytes. These numbers are separated by periods (“dots” in telecom parlance), and the notation is referred to as dotted decimal notation. A typical address might be 193.255.192.24 (Shepard, 2005).

Because of the largely unexpected growth of the Internet since 1993, it was roundly concluded that IP version 4 (IPv4) was inadequate for the emerging and burgeoning needs of Internet applications. In 1995 IP version 6 (IPv6) was introduced, designed to deal with the shortcomings of version 4. Changes included increased IP address space from 32 to 128 bits, improved support for differentiable QoS requirements, and improvements with regard to security, confidentiality, and content integrity (Shepard, 2005).

TCP is an ironclad, absolutely guaranteed service delivery protocol, with all of the attendant protocol overhead to expect from such a capable protocol. UDP, on the other hand, is a more lightweight protocol, used for delay-sensitive applications like VoIP. Its overhead component is relatively light. In TCP and UDP messages, higher-layer applications are identified by port identifiers. The port identifier and IP address together form a socket, and the end-to-end communication between two or more systems is identified by a four-part complex address: the source port, the source address, the destination port, and the destination address (Shepard, 2005).

The TCP/IP Application-Layer protocols support the utilities that make the Internet, well, useful. They include the BGP, the DNS, the File Transfer Protocol (FTP), the Hypertext Transfer Protocol (HTTP), OSPF, the Packet Internetwork Groper (Ping), the Post Office Protocol (POP), the Simple Mail Transfer Protocol (SMTP), the Simple Network Management Protocol (SNMP), the Secure Sockets Layer Protocol (SSL), and TELNET. This is a small sample of the many applications that are supported by the TCP/IP Application Layer (Hardy, 2003).

1.1.7. VoIP Standards

The VoIP network delivers a level of service and a consistent set of applications between users that are remarkably similar to those delivered by the public switched telephone network (PSTN). Voice and data are routinely handled by these systems. In the PSTN, VoIP networks must have their own set of protocols that manage call setup, maintenance, and teardown (Camarillo,2002).

Two international standards bodies create and publish the standards that govern the transmission of voice and other services over IP networks: the International Telecommunications Union (ITU) and the IETF. Their approaches are somewhat different but have the same goal. The ITU's primary standard is H.323; the IETF's is the Session Initiation Protocol (SIP) (Camarillo,2002).

1.1.7.1. H.323 Stantard

H.323 started in 1996, an ITU-T standard for the transmission of multimedia content over ISDN. It was the first multimedia signaling Standard and owes its provenance to H.320, the original standard for video conferencing. Its original goal was to connect LAN-based multimedia systems to network-based multimedia systems. It originally defined a network architecture that included gatekeepers, which performed zone management and address conversion; endpoints, which were terminals and gateway devices; and multimedia control units, which served as bridges between multimedia types. H.323 has now been rolled out in *four phases* (Shepard, 2005).

Phase one defined a three-stage call setup process: a precall step, which performed user registration, connection admission, and exchange of status messages required for call setup; the actual call setup process, which used messages similar to ISDN's Q.931; and finally a capability exchange stage, which established a logical communications channel between the communicating devices and identified conference management details (Shepard, 2005).

Phase two allowed for the use of Real-Time Transport Protocol (RTP) over ATM, which eliminated the added redundancy of IP and also provided for privacy and authentication as well as greatly demanded telephony features such as call transfer and call forwarding. RTP has an added advantage: When errors result in packet loss, RTP does not request resends of those packets, thus providing for real-time processing of application-related content. No delays result from errors (Shepard, 2005).

Phase three added the ability to transmit real-time fax after establishing a voice connection (Shepard, 2005).

And phase four, released in May 1999, added call connection over UDP, which significantly reduced call setup time, interzone communications, call hold, park, call pickup, and call and message-waiting features. This last phase bridged the considerable gap between IP voice, which was largely Internet-based catch-as-catch-can and carrier-grade IP telephony. Several Internet telephony interoperability concerns are addressed by H.323. These include gateway-to-gateway interoperability, which ensures that telephony can be accomplished between different vendors' gateways; gatekeeper-to-gatekeeper interoperability, which does the same thing for different vendors' gatekeeper devices; and finally gateway-to-gatekeeper interoperability, which completes the interoperability picture. H.323 offers a suite of functions for audio, video, and data conferencing, and it defines a set of functional modules similar to SIP. They include terminals, which are physical and soft phones; gateways, which provide the protocol conversion between packet and telephony; a gatekeeper, which performs address translation, connection admission control, and bandwidth management; and multipoint conferencing units (MCUs), which enable multiparty voice and videoconferencing. H.323 also supports a variety of collaborative applications, including screensharing and videoconferencing. Figure 1.4 shows a typical H.323-based network (Shepard, 2005).

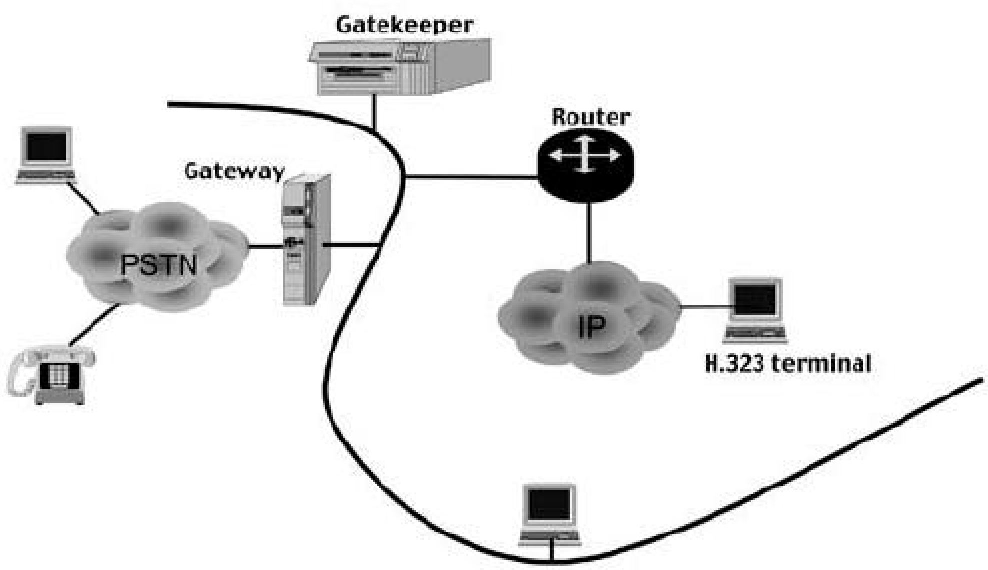

Figure 1.4. Typical H.323 network showing key components (Shepard, 2005)

That being said, a number of downsides exist to H.323 that have resulted in its displacement by SIP as the preferred protocol for multimedia call management. Because the standard supports so many enhanced functions, the protocols are large and complex and therefore expensive to develop. Furthermore, H.323-compliant devices must host a wider range of protocols to be H.323 compliant and therefore must be more complex and expensive (Shepard, 2005).

H.323 is an umbrella protocol, shown in Figure 1.5, meaning that it comprises a range of subtending standards that give it its functionality. H.323 had migrated from being purely a peer-to-peer protocol to having a more traditional, hierarchical design. The greatest advantage that H.323 offers is maturity. It has been available for some time now, and while robust and full featured, it was not originally designed to serve as a peer-to-peer protocol. Its maturity, therefore, is not enough to carry it. It currently lacks a network-to-network interface and does not adequately support congestion control. This is not generally a problem for private networks, but it becomes problematic for service providers who wish to interconnect PSTNs and provide national service among a cluster of providers. As a result many service providers have chosen to deploy SIP instead of H.323 in their national Networks (Camarillo,2002).

1.1.7.2. The Session Initiation Protocol (SIP)

SIP was conceived of as a flexible and adaptable protocol that could serve as an alternative to its more mature cousin, H.323. Whereas H.323 was originally rolled out as a governance protocol to control the delivery of multimedia traffic on LANs, SIP was created specifically with VoIP in mind. SIP defines a set of standard objects and a message hierarchy for communicating among the various elements that comprise the typical VoIP network. SIP is made up of a collection of modules that work together to provide signaling functionality across the network (Shepard, 2005).

These modules, shown in Figure 1.5, include user agent clients; *a location server*, which relates a client device to a specific IP address; *proxy servers*, which are responsible to forward call requests from one server to another on behalf of SIP clients ; and *redirect servers*, which transmit the called party's address back to the calling party so that the connection can be made(Shepard, 2005).

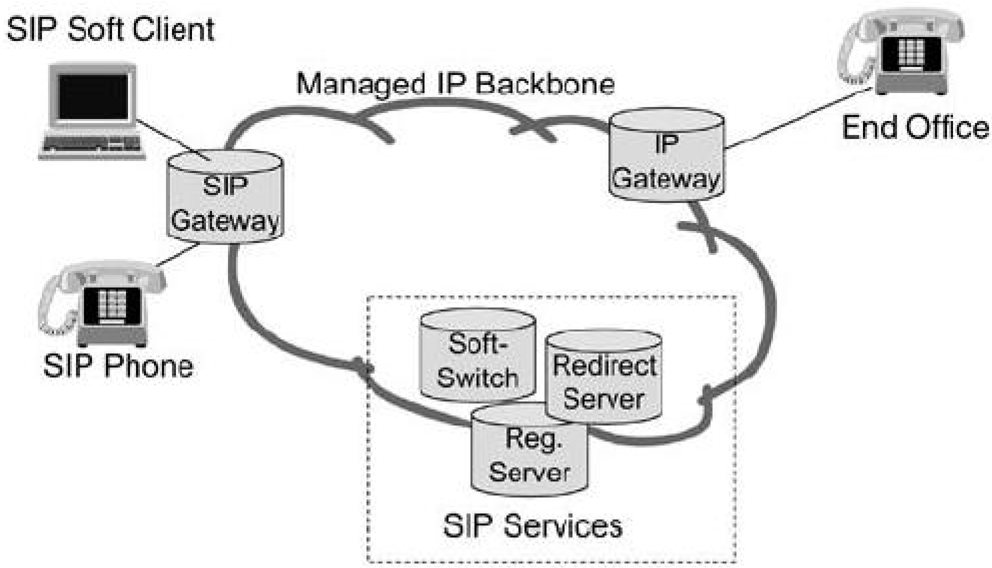

Figure 1.5. Typical SIP network showing key components (Shepard, 2005)

These devices rely on a limited set of SIP messages that are used to perform basic signaling tasks, maintenance, and teardown. They include *Invite*, used to establish or join a call session; *ACK*, used to acknowledge an invitation; *Register*, used to register a user with a server; *Options*, which are used to petition for data about the capabilities of the server; *Cancel*, which cancels a previous request; and *Bye*, which ends a session. Figure 1.6 shows a typical exchange of information using the SIP protocol (Shepard, 2005).

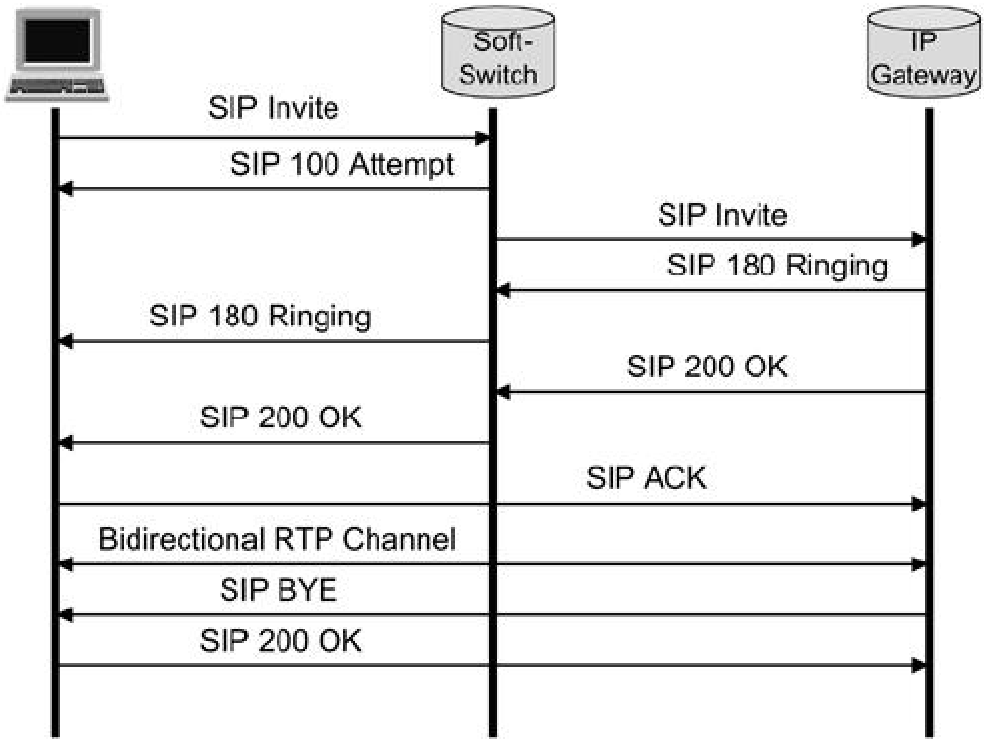

Figure 1.6. Information exchange in SIP (Shepard, 2005)

SIP is designed to establish peer-to-peer sessions between Internet routers. The protocol defines a variety of server types, including feature servers, registration servers, and redirect servers. SIP supports fully distributed services that reside in the actual user devices, and because it is based on existing IETF protocols it provides a seamless integration path for voice/data integration. Ultimately, telecommunications, like any industry, revolves around profitability. Any protocol that allows new services to be deployed inexpensively and quickly will immediately catch the eye of service providers. Like TCP/IP, SIP provides an open architecture that can be used by any vendor to develop products, thus ensuring multivendor interoperability. And because SIP has been adopted by such powerhouses as Lucent, Nortel, Cisco, Ericsson, and 3Com and is designed for use in large carrier networks with potentially millions of ports, its success is reasonably assured. Originally, H.323 was to be the protocol of choice to make this possible. And although H.323 is clearly a capable suite of protocols and is indeed quite good for VoIP services that derive from ISDN implementations, it remains quite complex. As a result it has been relegated for use as a video-control protocol and for some gatekeeper-to-gatekeeper communications functions. Although H.323 continues to enjoy its share of supporters, it is slowly being edged out of the limelight. SIP supporters claim that H.323 is far too complex and rigid to serve as a standard for basic telephony setup requirements, arguing that SIP, which is architecturally simpler and imminently extensible, is a better choice. Like other VoIP-related protocols, SIP is used to set up and tear down multimedia sessions between communicating endpoints. These multimedia sessions can include multiparty conferences, telephone calls, and distribution of multimedia content. These capabilities are central to the routine care and feeding of the enterprise, not to mention the enormous boon they represent for the enterprise call center. SIP is a lightweight, text-based protocol transported via TCP or UDP and is designed to operate according to the ad hoc rules established for all Internet protocols: easy to implement, simple in operation, efficient, and scalable (Camarillo,2002).

1.1.8. Introduction to VoIP

In the PSTN, the analog voice signal is sampled by the digital PSTN and converted to a digital bit stream. The signal can be compressed if desired, but either way the signal is transported through the network in digital format. The process is the same in PBX environment with a few minor changes; the PBX is, after all, simply a remote switch of sorts connected to the PSTN via high-bandwidth facilities. A digital handset connected to a PBX is handled in very similar fashion. In the IP environment, changes occur at this point. At this juncture the data signal must be packetized for transmission across the IP network as shown in Figure 1.7. It is important to note one key difference that exists between traditional PBXs and IP systems (Camarillo,2002).

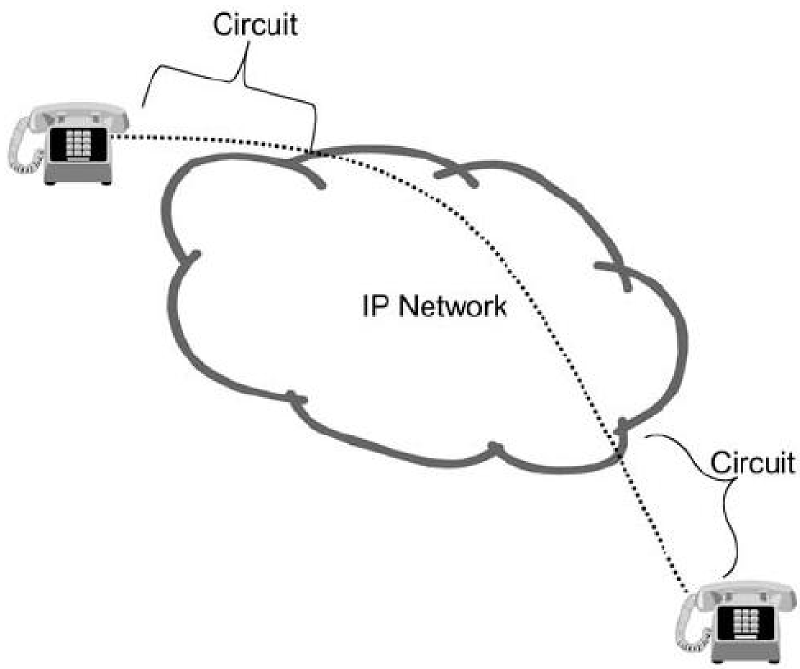

Figure 1.7. Circuit-to-packet conversion (Shepard, 2005)

IP voice has become a reasonable technological alternative to traditional PCM voice. VoIP gateways are in the late stages of development with regard to reliability, features, and manageability. Consequently, service providers wishing to deploy VoIP solutions have a number of options available to them (Camarillo,2002).

1.2. Single Board Computers

An embedded system is an applied computer system, as discriminated from other types of computer systems such as personal computers. Embedded systems are more small in hardware and/or software functionality than a personal computer (PC). In terms of hardware limitations, this can mean limitations in processing performance, hardware functionality, power consumption, memory, and so forth. In software, this typically means limitations relative to a PC, fewer applications, no operating system (OS) or a limited OS, scaled-down applications, or less abstraction-level code (Sloss, Symes, Wright, 2004).

An embedded system is designed to perform a devoted function. Most embedded devices are before designed for one specific function. However, they have seen devices such as personal data assistant (PDA)/cell phone hybrids, which are embedded systems designed to be able to do a variety of primary functions (Vahid, 1999). Today typical a Single Board Computer (SBC) is shown figure 1.8 below.

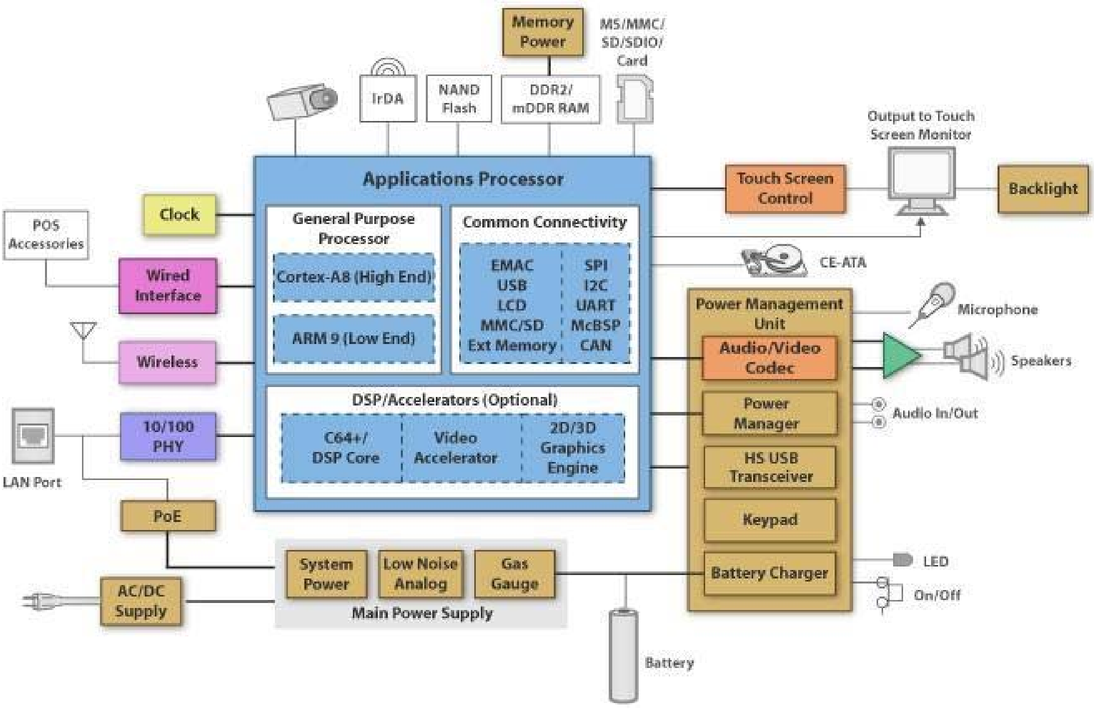

Figure 1.8. A typical single board computer example

An embedded system is a computer system with higher quality and reliability requirements than other types of computer systems. Some families of embedded devices have a very high threshold of quality and reliability requirements. Some devices that are called embedded systems, such as PDAs or web pads, are not really embedded systems. There is some discussion as to whether or not computer systems that meet some, but not all of the traditional embedded system definitions are actually embedded systems or something else. Electronic devices in just about every engineering market segment are classified as embedded systems in table 1.2 (Noergaard, 2005).

Table 1.2. Examples of embedded systems and their markets

Market	Embedded Device
Automotive	Ignition System
	Engine Control
	Brake System (i.e., Antilock Braking System)
Consumer Electronics	Digital and Analog Televisions
	Set-Top Boxes (DVDs, VCRs, Cable Boxes, etc.)
	Personal Data Assistants (PDAs)
	Kitchen Appliances (Refrigerators, Toasters, Microwave Ovens)
	Automobiles
	Toys/Games
	Telephones/Cell Phones/Pagers
	Cameras
	Global Positioning Systems (GPS)
Medical	Infusion Pumps
	Dialysis Machines
	Prosthetic Devices
	Cardiac Monitors
Networking	Routers
	Hubs
	Gateways
Office Automation	Fax Machine
	Photocopier
	Printers
	Monitors
	Scanners

The hardware layer is shown figure 1.9 contains all the major physical components located on an embedded board, whereas the system and application software layers contain all of the software located on and being processed by the embedded system (Noergaard, 2005).

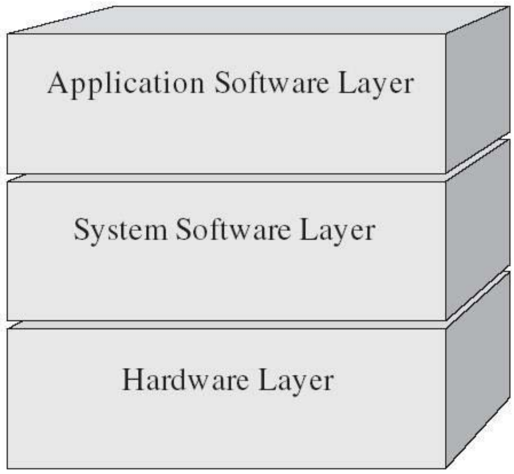

Figure 1.9. The model of Embedded System

The hardware components within an embedded system can only directly transmit, store, and execute machine code, a basic language consisting of ones and zeros. Machine code was used in earlier days to program computer systems, which made creating any complex application a long and tedious ordeal. In order to make programming more efficient, machine code was made visible to programmers through the creation of a hardware-specific set of instructions, where each instruction conformed to one or more machine code operations. These hardware- specific sets of instructions were referred to as assembly language (Noergaard, 2005).

Over time, other programming languages, such as Java, C, C++, etc., evolved with instruction sets that were more hardware-independent. These are commonly referred to as highlevel languages because they are semantically further away from the machine code, they more resemble human languages, and are typically independent of the hardware. This is in contrast to a low-level language, such as assembly language, which more closely resembles machine code. Unlike high-level languages, low-level languages are hardware dependent, meaning there is a unique instruction set for processors with different architectures (Ball, 2002).

Machine code is the only language the hardware can directly execute, all other languages need some type of mechanism to generate the corresponding machine code. In figure,it is shown a host and target system diagram .This mechanism usually includes one or some combination of preprocessing, interpretation and translation(Vahid, 1999).

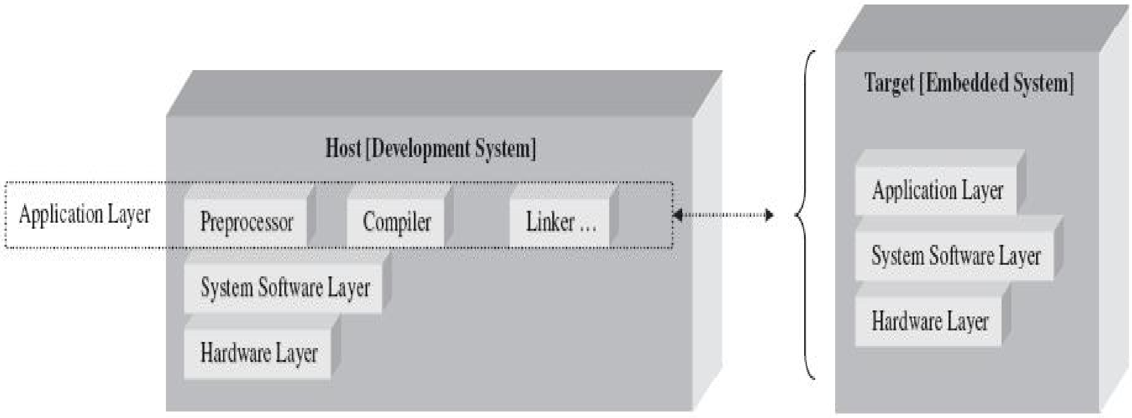

Figure 1.10. Host and Target System Diagram (Noergaard, 2005)

Depending on the language, these mechanisms exist on the programmer's host system (typically a nonembedded development system, such as a PC), or the target system (the embedded system being developed). Many languages convert source code, either directly or after having been preprocessed through use of a compiler, a program that generates a particular target language from the source language in figure 1.11 (Vahid, 1999).

A compiler typically translates all of the source code to some target code at one time. As is usually the case in embedded systems, compilers are located on the programmer's host machine and generate target code for hardware platforms that differ from the platform the compiler is actually running on (Andrews, 2005).

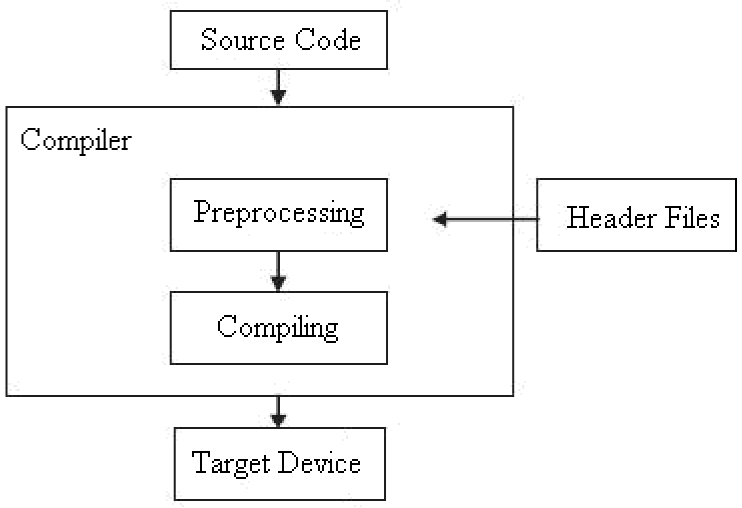

Figure 1.11. Compilation diagram

The compilers are commonly referred to as the cross-compilers. In the case of assembly language, the compiler is simply a specialized cross-compiler referred to as an assembler, and it always generates machine code (Vahid, 1999).

2. PREVIOUS WORKS

VoIP phone manufacturers are competing to quickly add the more efficient new items to list. Recently the multimedia support VoIP products them list have started to increase.

2.1. Literature Survey

2.1.1. Information Model for SIP Device

Considering SIP network management requirements and referring to IETF SIP standard, it is constructed the SIP device information model with Unified Modelling Language (UML) as shown in figure 2.1.

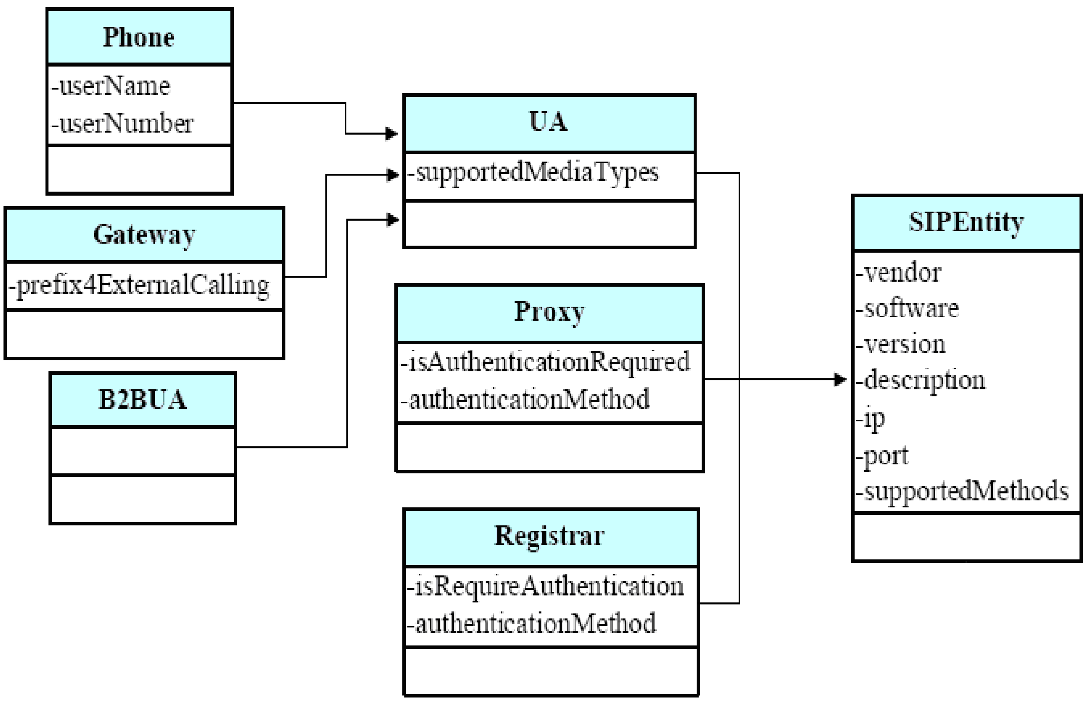

Figure 2.1. Information Model for SIP Device (Zhou, Li, Xia, Cai, Ying, 2008)

The model provides information about SIP devices including their roles, which differentiate the service logics by their proceeding of SIP messages vendors, software, versions, supported methods and so on (Zhou, Li, Xia, Cai, Ying, 2008).

As the scale of SIP network becomes huge, using floodlike probing to discover SIP network is straightforward but inefficient, which brings massive load to the network traffic. So it is needed to find a strategy of SIP message probing for efficient management. The precondition of this method for SIP network discovery is: different types of SIP devices react to the same SIP request message differently while the ones belong to the same type react the same way. So if there are two types of SIP devices A, B. A is an unknown SIP device in the unknown SIP network, and it is need to discover its information model. And B’s information model is of aware and has already been differentiated from others by using a list of SIP messages. If A elicits the same responses as B to the same list of SIP messages, we can determine that A has the same information model as B (Zhou, Li, Xia, Cai, Ying, 2008).

2.1.2. Developing Video Phones with ARM Processor-based

General-purpose applications processors based on ARM9E and ARM11 processor architectures have increased in processing power, to the point where it is possible to move the audio processing tasks usually performed by a DSP to the applications processor. The ARM9E and ARM11 instruction set extensions are highly flexible and can be used to optimize virtually any type of media processing (Ward, 2005).

Using the ‘E’ extensions of the ARM926EJ-S reduces the Mhz required to execute a typical voice encoder by as much as 20%, when compared to its execution on an ARM9 family processor without the ‘E’ extensions. This saving in Mhz translates to either a lower clock speed approximately 5 Mhz over a well optimized implementation for an architecture without these instructions. However, this gain is magnified when implementing a more complex, wide-band codec such as G.722.2, where approximately 20 Mhz can be saved through a DSP extension based implementation (Ward, 2005).

VoIP codecs (G.711, G.729AB, G.723.1, iLBC), audio processing (DTMF and call progress tone detection / generation), voice quality enhancement (line and acoustic echo cancellation, jitter buffers, etc.) and other similar functions can now all be effectively executed on the applications processor if carefully implemented with assembly coded and hand-optimized software, while utilizing some form of hardware acceleration for the video encode and decode as shown figure 2.2 (Ward, 2005).

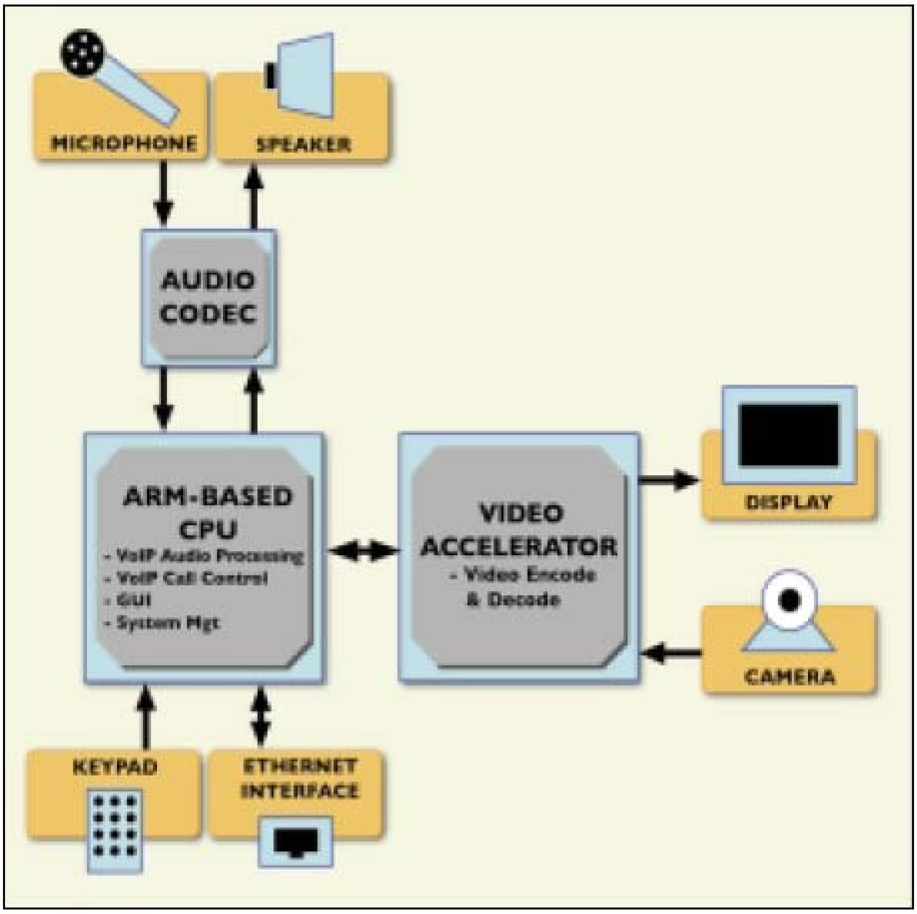

Figure 2.2. Architecture of a Soft-VoIP Based Video Phone (Ward, 2005).

2.2. The Example VoIP Phones from the Producer

In this section some examples have realized is given about VoIP phones.

- **Linksys Voice System' SPA 901 Series IP Phones**

Figure 2.3. Linksys Voice System' SPA 901 (www.linksysbycisco.com)

The SPA901 provides an entry-level IP phone that can be wall mounted in figure 2.3. The hardware features provided by the SPA901 are High Quality Handset and Cradle, Ethernet LAN.

- **Linksys Voice System' SPA 962 Series IP Phones**

The SPA962 has a 320 x 240 true color, four-inch, LCD, provides up to six telephone extensions, and supports PoE in figure 2.4.

Figure 2.4. Linksys Voice System' SPA 962 (www.linksysbycisco.com)

Full-color LCD display Lists device status and configuration options. Telephone keypad enters numeric digits for initiating a call or for entering configuration information. Navigation button Scrolls between display and configuration options in the LCD display. Soft keys 1-4 Selects options on the LCD display. Line status indicators 1-6 Displays status of each extension.

In back panel, phone jack Connects to the handset. Ethernet ports connect to the SPA9000 through a local switch. Use the second port to connect to a PC or other LAN device. AUX RJ-11 port connects to other Linksys IP Phones.

- **Linksys Voice System' SPA9000 IP PBX System**

The SPA9000 IP PBX System along with the SPA400, which is Internet Telephony Gateway, provides for an analog line and voicemail.

The system provides auto attendant features for multiple extensions. Features include Auto Attendant, Shared line appearances, Configurable call routing, Multiple DID numbers per VoIP line, Call hunting.

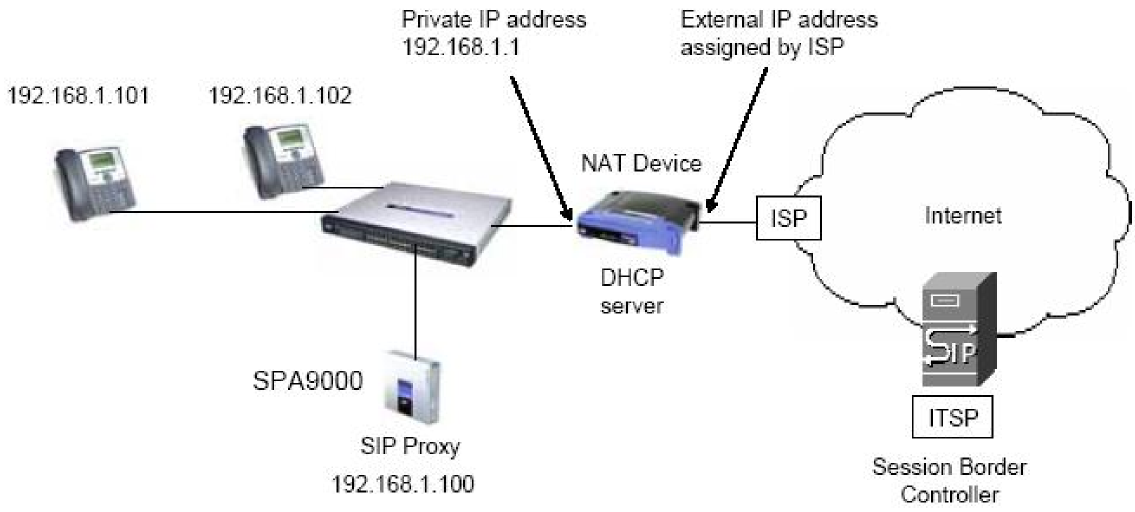

Figure 2.5. Linksys' SPA9000 IP PBX System (www.linksysbycisco.com)

SPA9000 administrative tasks can be perform using an Interactive Voice Response (IVR) system or a built-in Web Server.

IP phones configured, each call requires 55 to 110 kbps in each direction. Therefore, using G.729 as the voice codec setting, and with an average business-grade broadband Internet connection supporting 1.5 Mbps downstream and 384 kbps upstream, a total of seven simultaneous conversations can be reliably supported with adequate bandwidth available for file downloads.

Linksys recommends using the Linksys IP phone with QoS-capable networking equipment that can prioritize the VoIP application traffic. QoS features are available on many Linksys data networking switches and routers. A QoS-enabled router prioritizes the packets going upstream to the ISP. Table 2.1. illustrates the bandwidth budget using different codecs.

Table 2.1. Ethernet Bandwidth Budget for VoIP Calling (www.linksysbycisco.com)

Codec	Approximate bandwidth budget for each side of conversation	2 calls	4 calls	6 calls	8 calls
G.711	110 kbps	220 kbps	440 kbps	660 kbps	880 kbps
G.726-40	87 kbps	174 kbps	348 kbps	522 kbps	696 kbps
G.726-32	79 kbps	158 kbps	316 kbps	474 kbps	632 kbps
G.726-24	71 kbps	142 kbps	284 kbps	426 kbps	568 kbps
G.726-16	63 kbps	126 kbps	252 kbps	378 kbps	504 kbps
G.729	55 kbps	110 kbps	220 kbps	330 kbps	440 kbps

This table is based on the following assumptions Bandwidth Calculated with No Silence Suppression and 20 Millisecond of payload per RTP packet.

- **Philips VoIP-enabled baseband for digital cordless phones PCD8072x**

Web-enabled technology and an advanced ARM7/DSP architecture let consumers use a cordless phone, rather than a computer, to make inexpensive domestic and international calls using the Internet.

Key features supported include High-end ARM7/DSP basebands with VoIP protocol stack, fast, simple way to upgrade existing cordless designs to VoIP, advanced cordless phone functionality, full compatibility with DECT (Digitally Enhanced Cordless Telecommunications), US-DECT, Bluetooth and Wi-Fi.

These VoIP-enabled basebands are the latest addition to the well-established PCD807xx family of highly integrated basebands for DECT cordless handsets. They can be used to create standalone VoIP handsets or expansion handsets for a broadband residential gateway. Also, because they are fully compatible with other members in the PCD807xx family, they can be used to quickly upgrade an existing design for VoIP functionality.Adding a VoIP stack to the baseband means that a cordless phone can use the Internet to place domestic and international calls.

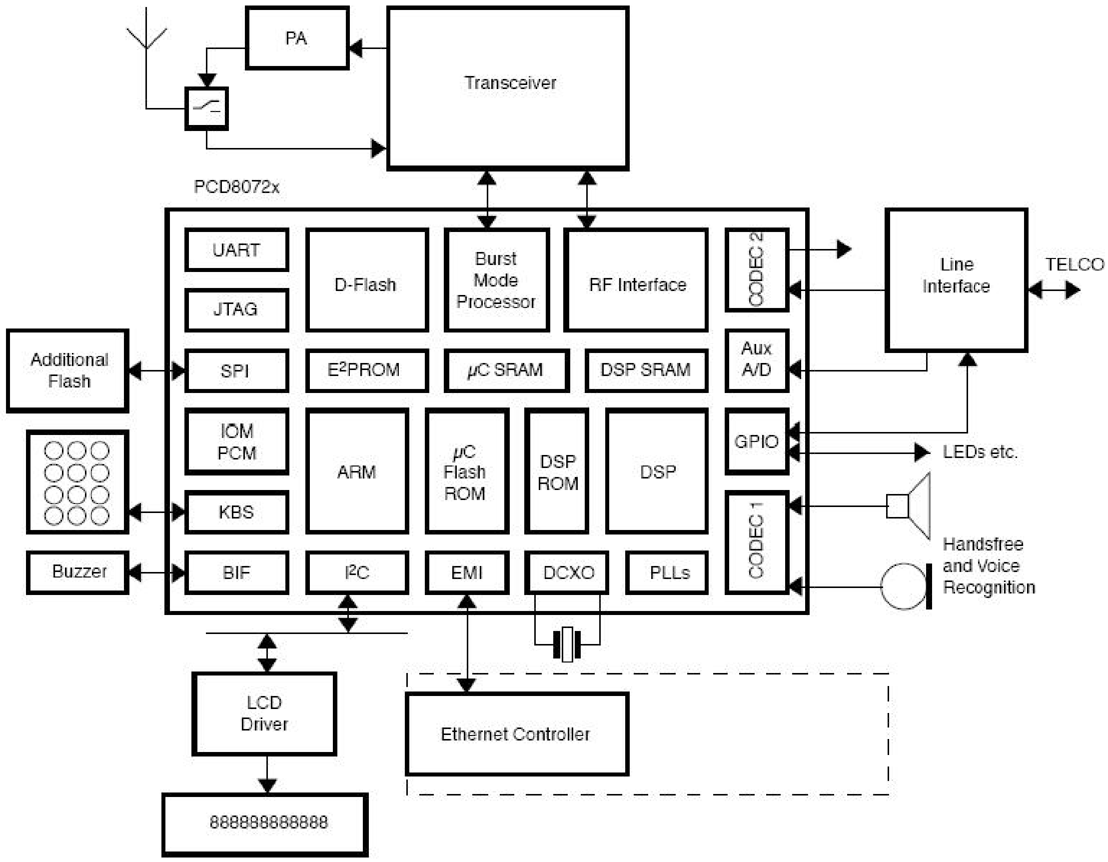

Figure 2.6. Philips PCD807xx cordless phones (www.philips.com)

Most residential gateways are NAT (Network Address Translation) based and the SW IP stack includes the unique STUN (Simple Traversal of UDP through NAT) support to overcome NAT & firewalls constraint for the end user when establishing calls for VoIP. The stack also supports the SIP ID. The PCD8072x architecture uses a high-end ARM7 and DSP technology. The baseband runs a complete VoIP stack and efficiently integrates the codecs as part of the firmware. The VoIP functionality builds on the familiar cordless architecture, so it's easy for digital cordless manufacturers to construct a complete VoIP system.

The architecture has also been optimized for very low power consumption, so the basebands deliver extended talk and standby times.

- **TI' TNETV1600 WLAN IP Phone Solution**

Texas Instruments' new Wireless Local Area Network (WLAN) IP Phone Platform offers a complete, fully integrated solution that supports a wide range of WLAN IP phones, from fully featured phones to simple, lower cost solutions. The full featured enterprise WLAN IP Phone market includes products with a graphical LCD display, a rich feature set and GUI.

TI's flexible solution, along with a comprehensive product and architectural roadmap, will allow customers to move smoothly and quickly from WLAN IP Phone solutions today to dual/converged mode phones in the future. With TI's expertise in VoIP, broadband and wireless technologies, this WLAN IP Phone platform provides the highest performance and integration with the lowest power consumption available.

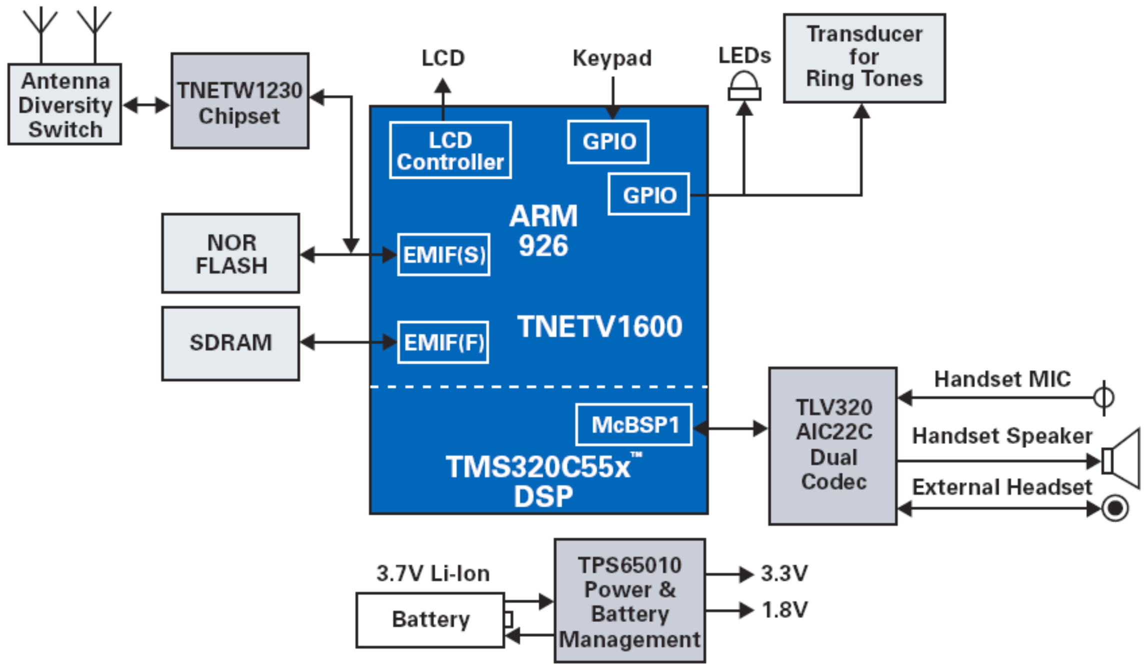

Figure 2.7. TNETV1600 Wireless LAN IP Phone Solution (www.ti.com/voip)

TI’s WLAN IP Phone evaluation platform is a handset form factor solution that includes everything needed for customer software development. The evaluation platform allows customers to develop their application software and man-machine interface in advance of hardware delivery.

Key features supported include Color LCD display (176 x 220 Portrait Mode), Handset and detachable headset support, standard 12 key telephony dial keypad, support for a detachable and rechargeable Li-Ion Battery with power gage indication, support for WLAN RF signal level indication, support for USB OTG connector and generic client driver.

3. MATERIAL AND METHODS

Through this part, the elements which are used to design the hardware are considered and the properties are discussed. The Single Board Computer (SBC) block diagram of the design needed for this study is shown below:

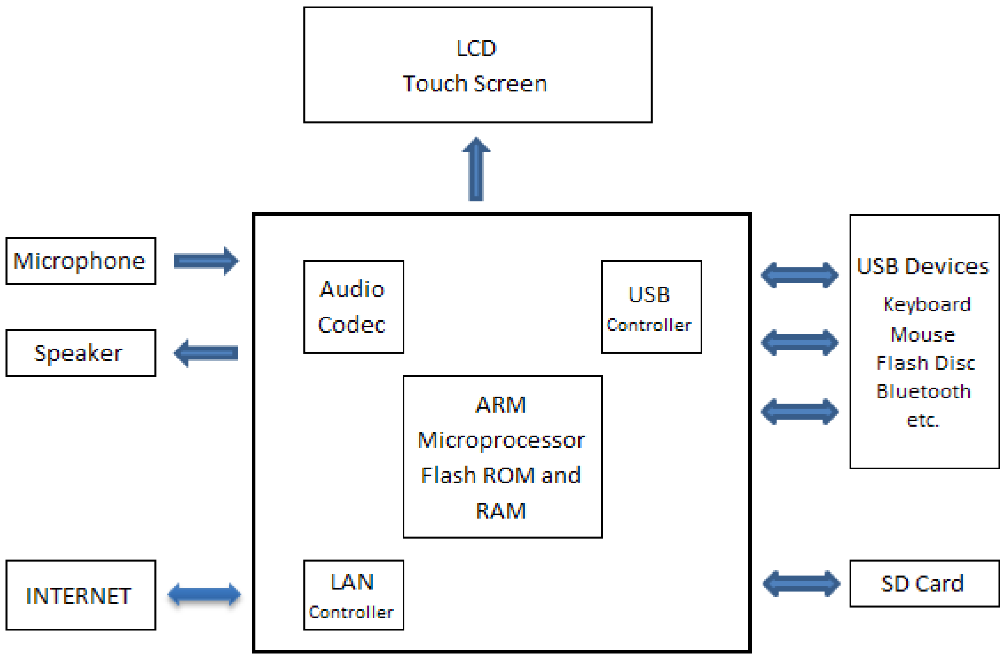

Figure 3.1. SBC Block Diagram

As the first occupation, it is considered for a SBC with the 32-bit microprocessor which is including the necessary hardware for our study.

In order to create good marketing opportunities for such a board, we have been in collaboration with the engineers interested in similar projects and we have decided to design the board shown in figure 3.2.

The rationale in preferring to design such a card rather than using the available boards in the market was to enable the hardware and software compatibility. This approach will also relatively help us in programming software in the future work.

Initially hardware tests have been performed for the board avoiding the matter in a small way PCB and solder problems. The hardware drivers on the board in order to work proper with the operating system (Windows CE 5.0) were written in embedded visual C++ programming language. Later all the components of the block diagram were tested. Finally the open source a VoIP application was tested on the board and these final tests included the operation with a Personal Computer and a mobile phone with WiFi.

3.1. The Hardware of the Single Board Computer (SBC)

The main materials used on the mini computer are as follows:

- Samsung S3C2440A 32-bit Microcontroller
- 64 MB Flash ROM
- 2x32 MB SDRAM
- UDA1341TS Audio Codec
- GL850A USB Hub Controller
- CS8900A LAN Controller
- TFT-LCD Module with Touch Screen

The detailed plan and photo of the board are shown in figure 3.2. In addition to the components specified above Zigbee, CAN, RS485T Controller modules are also available. The hardware on the board are designated according to participant requests and added on the board.

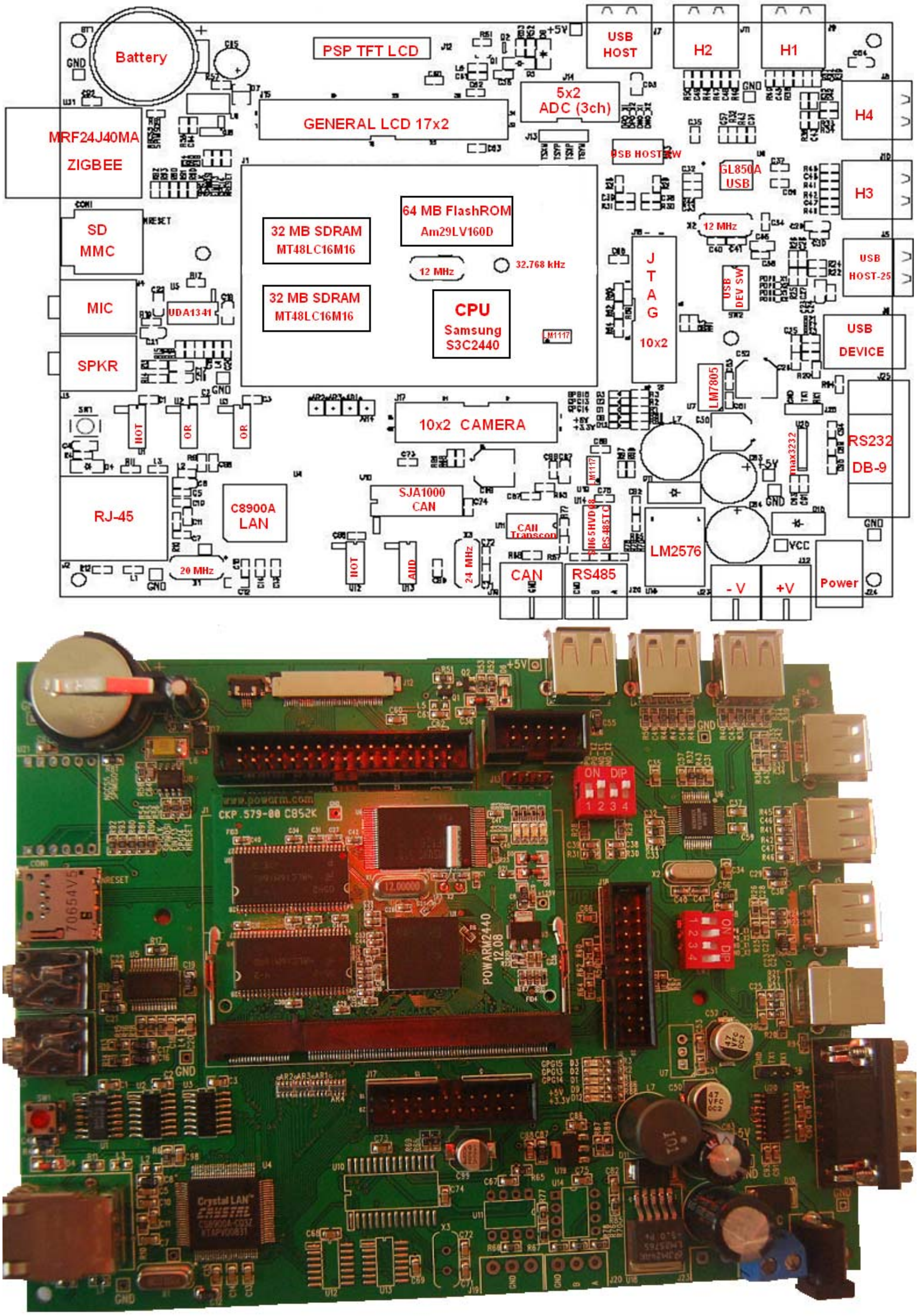

Figure 3.2. The detailed plan and photo of the board

3.1.1. Samsung S3C2440A 32-bit Microcontroller

S3C2440A is designed to provide hand-held devices and general applications with low-power, and high-performance microcontroller solution in a small die size (Samsung Electronic, 2004).

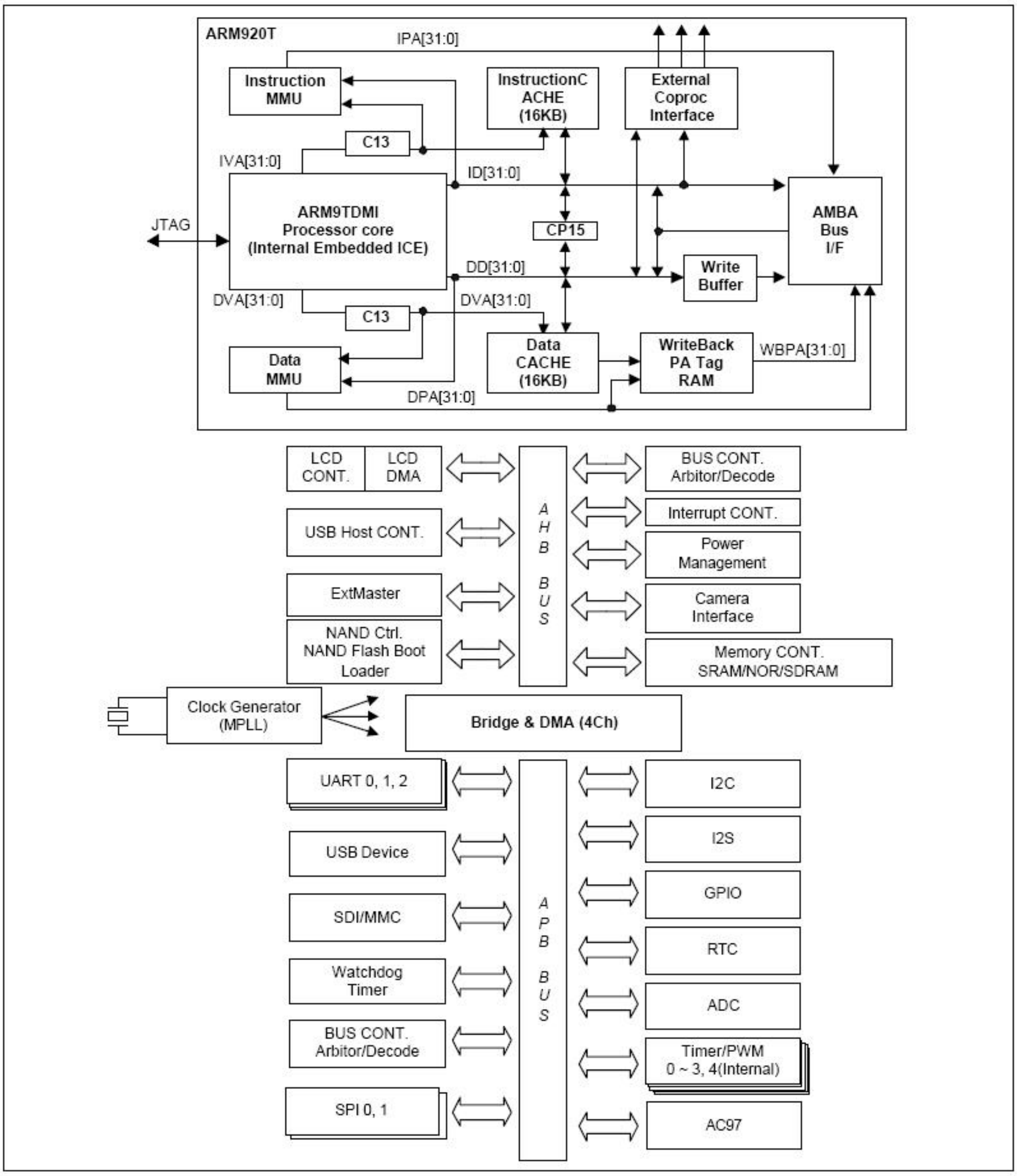

Figure 3.3. The Structure of the Samsung S3C2440A (Samsung Electronic, 2004)

The S3C2440A is developed with ARM920T core, 0.13um CMOS standard cells and a memory complier. Its low power, simple, elegant and fully static design is particularly suitable for cost and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA). The S3C2440A offers outstanding features with its CPU core, a 16/32-bit ARM920T RISC processor designed by Advanced RISC Machines, Ltd (Samsung Electronic, 2004).

The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length. By providing a complete set of common system peripherals, the S3C2440A minimizes overall system costs and eliminates the need to configure additional components. The integrated on-chip functions and some features of S3C2440A which is the important for our study are as follows :

- Around 1.2V internal, 1.8V/2.5V/3.3V memory, 3.3V external I/O microprocessor with 16KB I-Cache/16KB DCache/MMU
- IIS Audio CODEC interface
- AC’97 CODEC interface
- SD Host interface version 1.0 & MMC Protocol version 2.11 compatible
- 8-ch 10-bit ADC and Touch screen interface
- 130 General Purpose I/O ports / 24-ch external interrupt source
- Integrated system for hand-held devices and general embedded applications
- Enhanced ARM architecture MMU to support WinCE, EPOC 32 and Linux
- Supports various types of ROM for booting (NOR/NAND Flash, EEPROM, and others)
- Supports storage memory for NAND flash memory after booting
- Generates the clock to operate MCU at maximum 400Mhz at 1.3V
- 60 Interrupt sources (One Watch dog timer, 5 timers, 9 UARTs, 24 ext. interrupts, 4 DMA, 2 RTC, 2 ADC, 1 IIC, 2 SPI, 1 SDI, 2 USB, 1 LCD, 1 Battery Fault, 1 NAND and 2 Camera), 1 AC97
- Supports multiple screen size (Typical screen size: 640x480, 320x240, 160x160, and others.) (Samsung Electronic, 2004).

- 289-FBGA Package

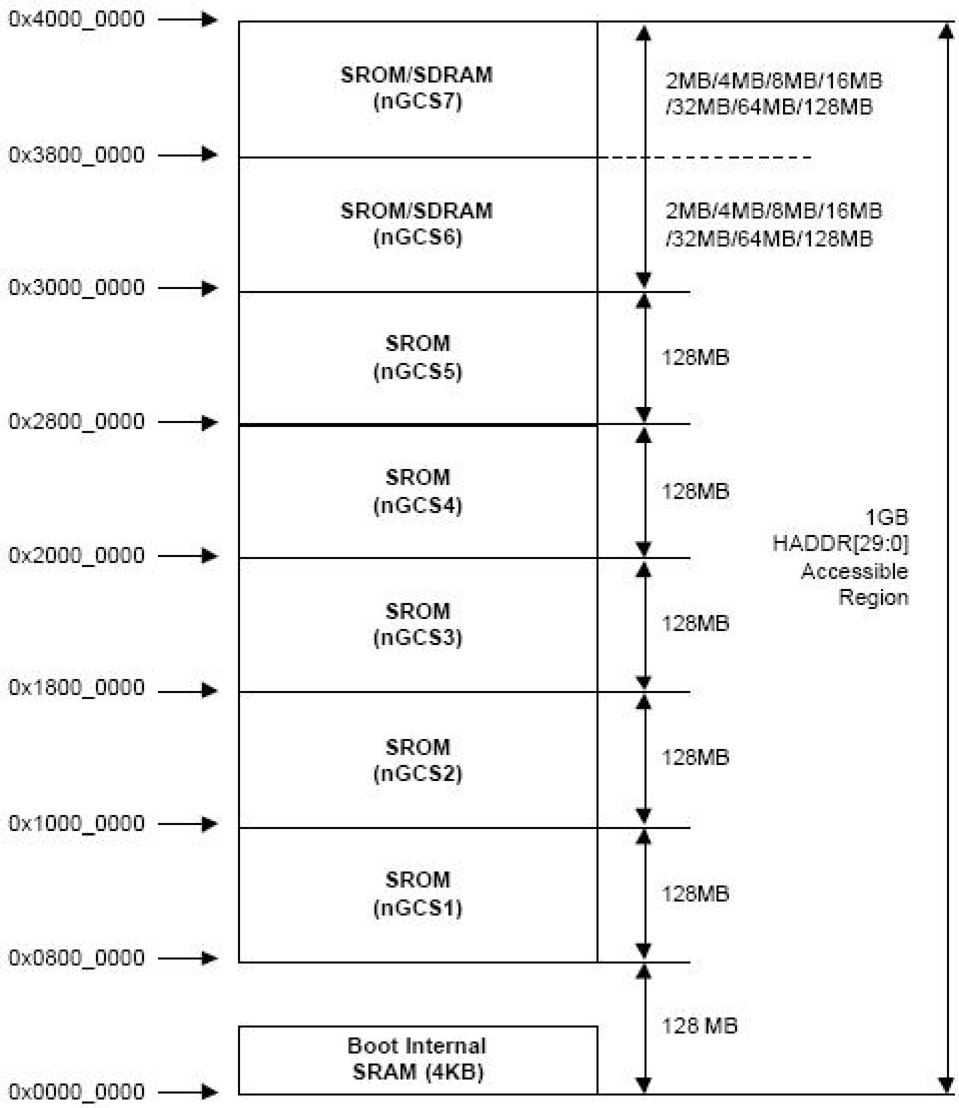

Figure 3.4. S3C2440A Memory Map (Samsung Electronic, 2004)

After reset S3C2440A Memory Map is shown above in figure 3.4. This address map is valid using NAND Flash for Boot ROM as we make .

3.1.2. Flash ROM

In this study the Am29LV160D memory chip is used as Flash ROM. The Am29LV160D is a 16 Mbit, 3.0 Volt-only Flash memory organized as 2,097,152 bytes or 1,048,576 words. Here is used 44-pin SO package. The word-wide data (x16) appears on DQ15–DQ0; the byte-wide (x8) data appears on DQ7–DQ0. This device is designed to be programmed in-system with the standard system 3.0 volt VCC supply. A 12.0 V VPP or 5.0 VCC are not required for write or erase operations. The device can also be programmed in Standard EPROM programmers (AMD Electronic, 2000).

The device requires only a single 3.0 volt power supply for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations (AMD Electronic, 2000).

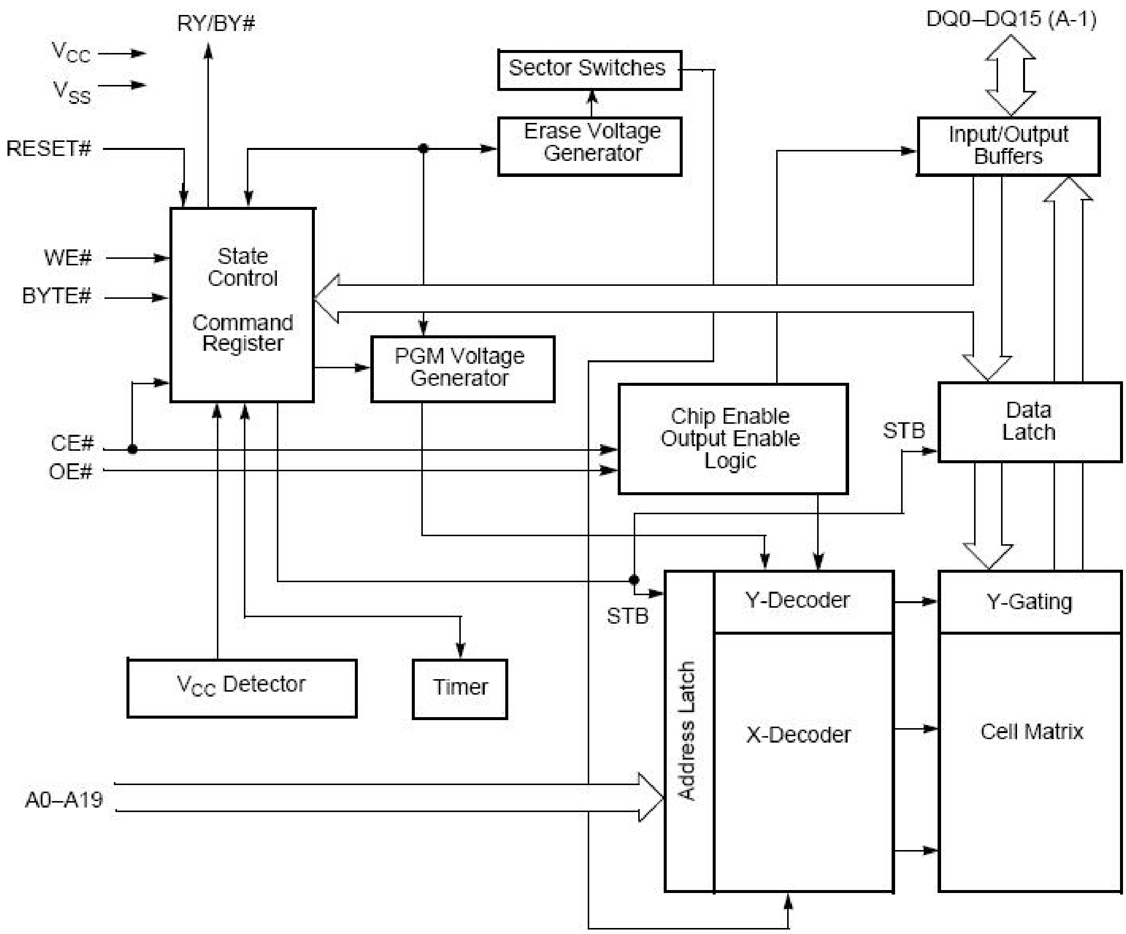

Figure 3.5. The Am29LV160D Flash ROM (AMD Electronic, 2000)

Some distinctive characteristic:

- Regulated voltage range: 3.0 to 3.6 volt read and write operations and for compatibility with high performance 3.3 volt microprocessors
- Access times as fast as 70 ns
- 200 nA Automatic Sleep mode current, 200 nA standby mode current, 9 mA read current, 20 mA program/erase current
- Supports full chip erase
- Top or bottom boot block configurations available
- Embedded Program algorithm automatically writes and verifies data at specified addresses
- Provides device-specific information to the system, allowing host software to easily reconfigure for different Flash devices (Philips Semiconductors, 2002)

The device electrically erases all bits within a sector simultaneously via Fowler-Nordheim tunneling. The data is programmed using hot electron injection (Philips Semiconductors, 2002).

3.1.4. SDRAM

The HY57V561620T is a 268,435,456bit CMOS Synchronous DRAM, ideally suited for the main memory applications which require large memory density and high bandwidth. HY57V561620 is organized as 4 banks of 4,194,304x16 (Hynix Electronic, 2001).

The HY57V561620T is offering fully synchronous operation referenced to a positive edge of the clock. All inputs and outputs are synchronized with the rising edge of the clock input. The data paths are internally pipelined to achieve very high bandwidth. All input and output voltage levels are compatible with LVTTL (Hynix Electronic, 2001).

Programmable options include the length of pipeline (CAS latency of 2 or 3), the number of consecutive read or write cycles initiated by a single control command (Burst length of 1, 2, 4, 8 or full page), and the burst count sequence(sequential or interleave). A burst of read or write cycles in progress can be terminated by a burst terminate command or can be interrupted and replaced by a new burst read or write command on any cycle (Hynix Electronic, 2001).

It's certain features:

- Single 3.3V ± 0.3V power supply
- Internal four banks operation
- Auto refresh and self refresh
- 8192 refresh cycles / 64ms
- 54 pin TSOP package

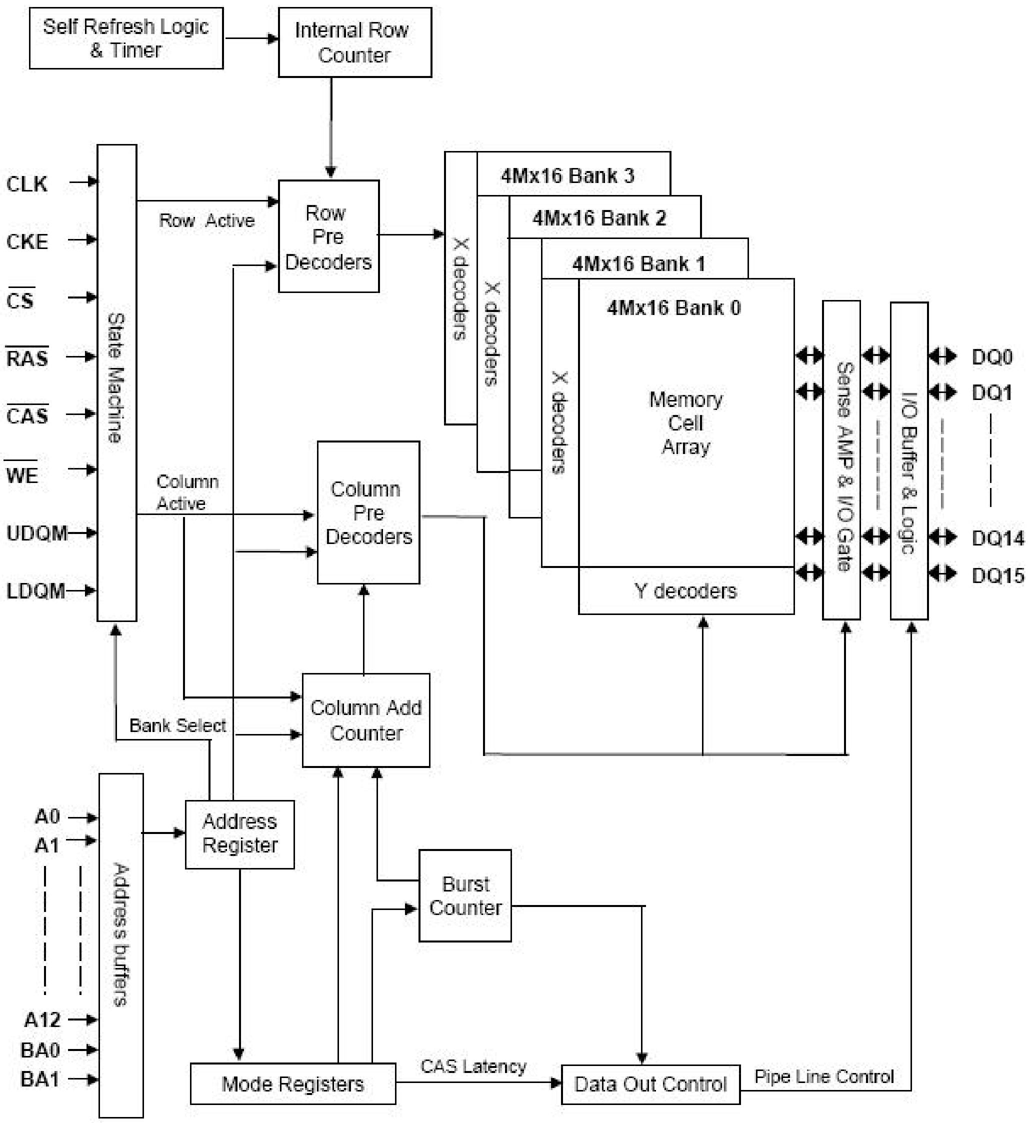

Figure 3.6. The HY57V561620T SDRAM (Hynix Electronic, 2001)

3.1.5. UDA1341TS Audio Codec

The UDA1341TS is a single-chip stereo Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) with signal processing features employing bitstream conversion techniques (Philips Semiconductors, 2002).

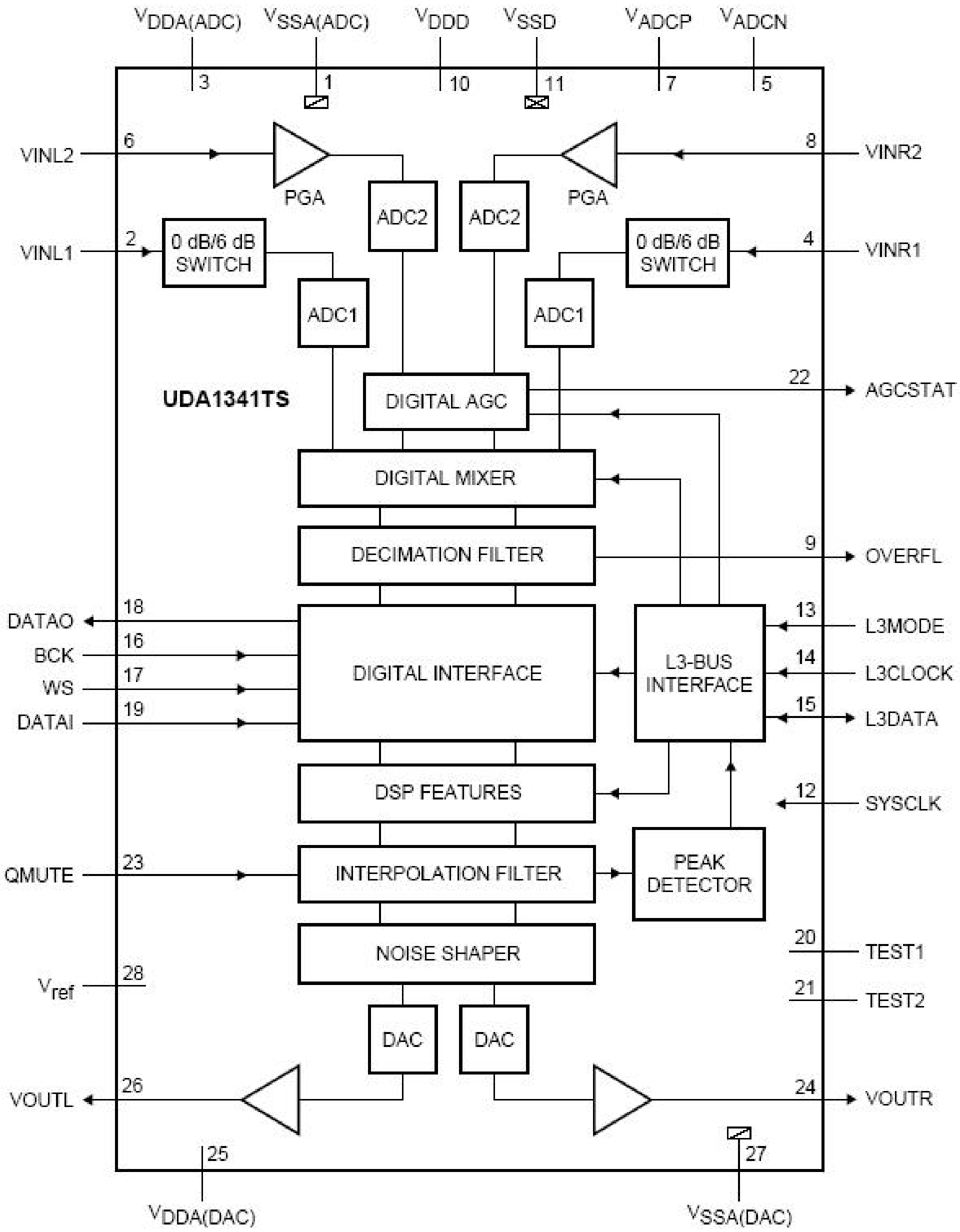

Figure 3.7. The UDA1341TS Audio Codec (Philips Semiconductors, 2002)

It is a plastic shrink small outline package (SSOP). It has fully integrated analog front end, including Programmable Gain Amplifier (PGA) and a digital Automatic Gain Control (AGC) (Philips Semiconductors, 2002).

Digital Sound Processing (DSP) feature makes the device an excellent choice for primary home stereo MiniDisc applications, but by virtue of its low power and low voltage characteristics it is also suitable for portable applications such as MD/CD boomboxes, notebook PCs and digital video cameras (Philips Semiconductors, 2002).

The UDA1341TS supports the I2S-bus data format with word lengths of up to 20 bits, the MSB-justified data format with word lengths of up to 20 bits, the LSB-justified serial data format with word lengths of 16, 18 and 20 bits and three combinations of MSB data output combined with LSB 16, 18 and 20 bits data input. The UDA1341TS has DSP features in playback mode like de-emphasis, volume, bass boost, treble and soft mute, which can be controlled via the L3-interface with a microcontroller (Philips Semiconductors, 2002).

Usage of the UDA1341TS on the board is shown in Fig.3.6.

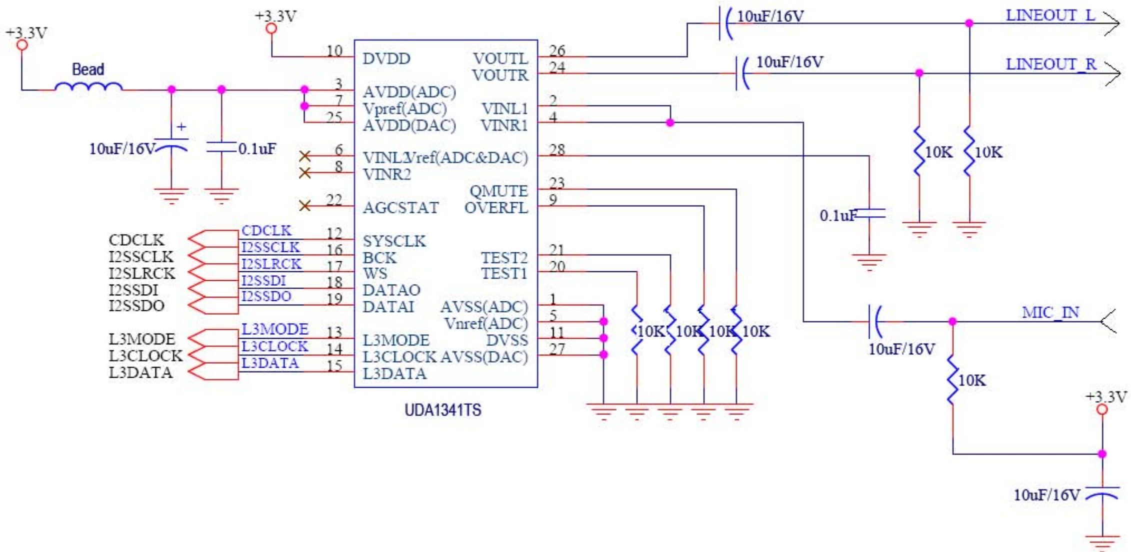

Figure 3.8. The Circuit of the UDA1341TS

3.1.6. GL850A USB Hub Controller

GL850A is Genesys Logic's advanced version Hub solutions which fully comply with Universal Serial Bus Specification Revision 2.0 (Genesys Logic, 2008).

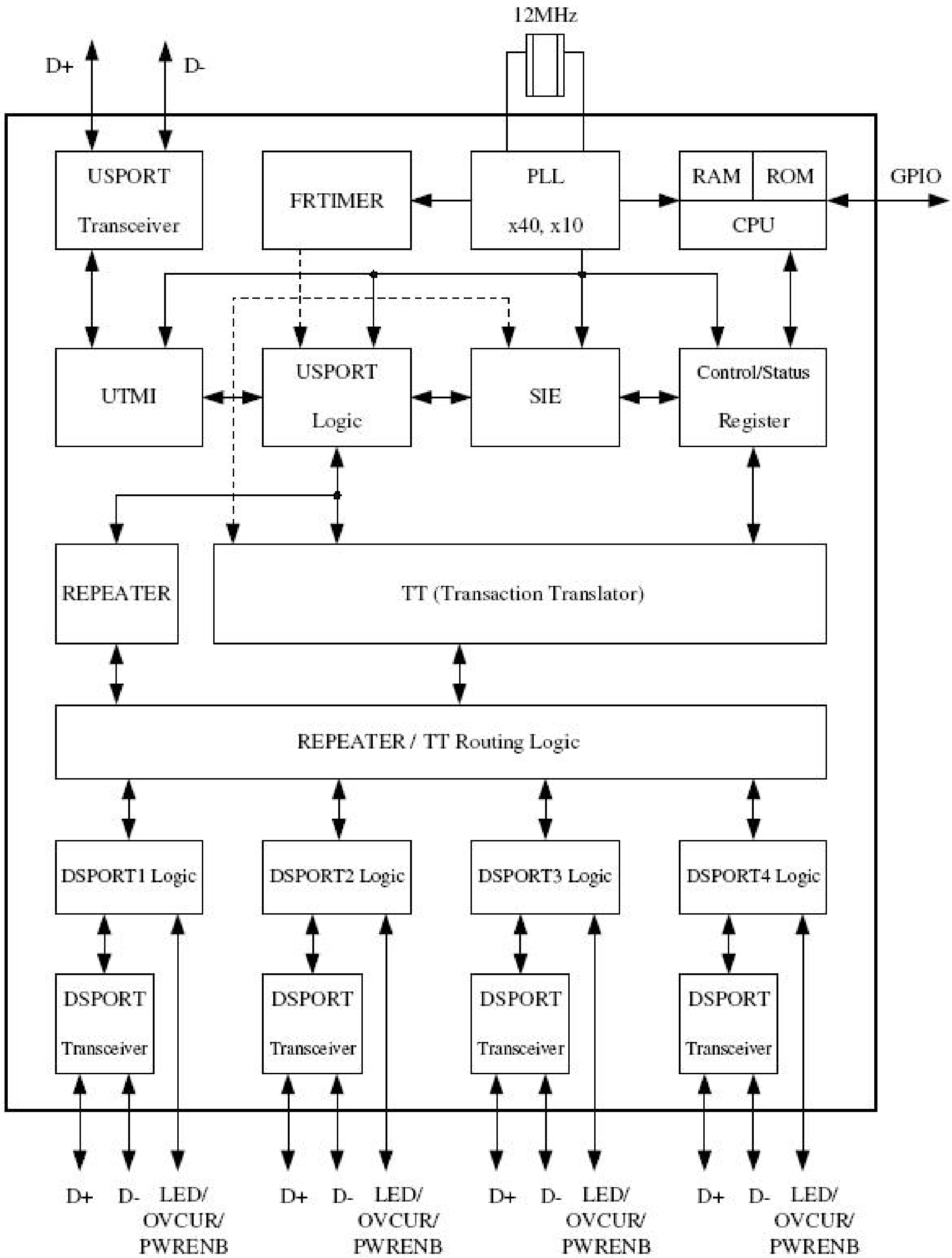

Figure 3.9. GL850A USB Hub Controller (Genesys Logic, 2008)

GL850A embeds an 8-bit RISC processor to manipulate the control/status registers and respond to the requests from USB host. Firmware of GL850A will control its general purpose I/O (GPIO) to access the external EEPROM and then respond to the host the customized PID and VID configured in the external EEPROM (Genesys Logic, 2008).

Default settings in the internal mask ROM is responded to the host without having external EEROM. GL850A is designed for customers with much flexibility. The more complicated settings such as PID, VID, and number of downstream ports settings are easily achieved by programming the external EEPROM (Genesys Logic, 2008).

Each downstream port of GL850A supports two-color (green/amber) status LEDs to indicate normal/abnormal status. GL850A also support both Individual and Gang modes (4 ports as a group) for power management. The GL850A (64-pin) is a full function solution which supports both Individual/Gang power management modes and the two-color (green/amber) status LEDs. The low pin-count version GL850A (48-pin) only supports Gang mode. In this study is used 48-pin LQFP package as Gang mode (Genesys Logic, 2008).

Usage of the GL850A on the board is shown in figure 3.10.

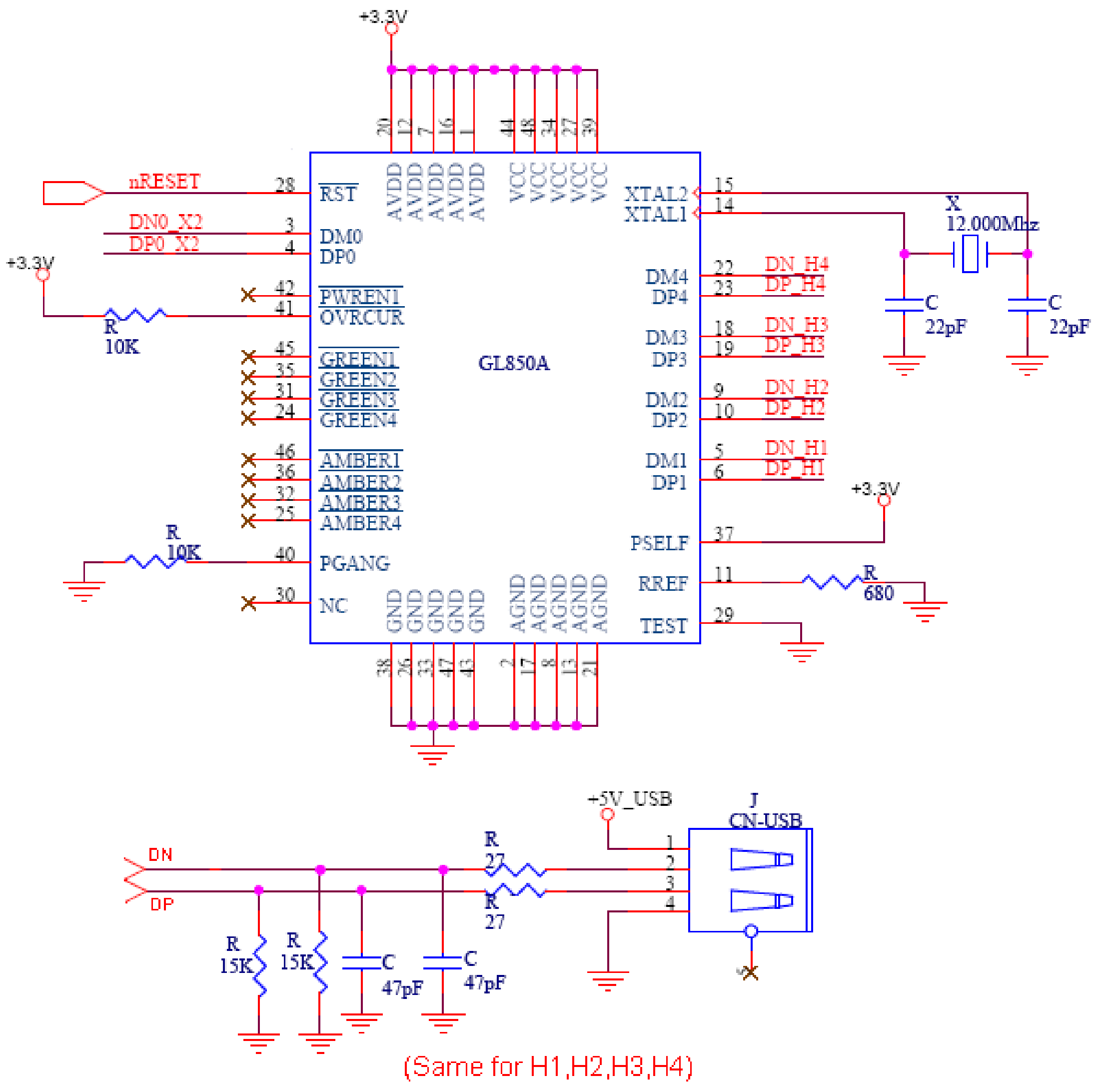

Figure 3.10. The Circuit of the GL850A

3.1.6. CS8900A LAN Controller

The CS8900A is a low-cost Ethernet LAN Controller optimized for the Industry Standard Architecture (ISA) bus and general purpose microcontroller busses. Its highly integrated design eliminates the need for costly external components required by other Ethernet controllers. The CS8900A includes on-chip RAM, 10BASE-T transmit and receive filters, and a direct ISA-Bus interface with 24 mA Drivers (Cirrus Logic, 2001).

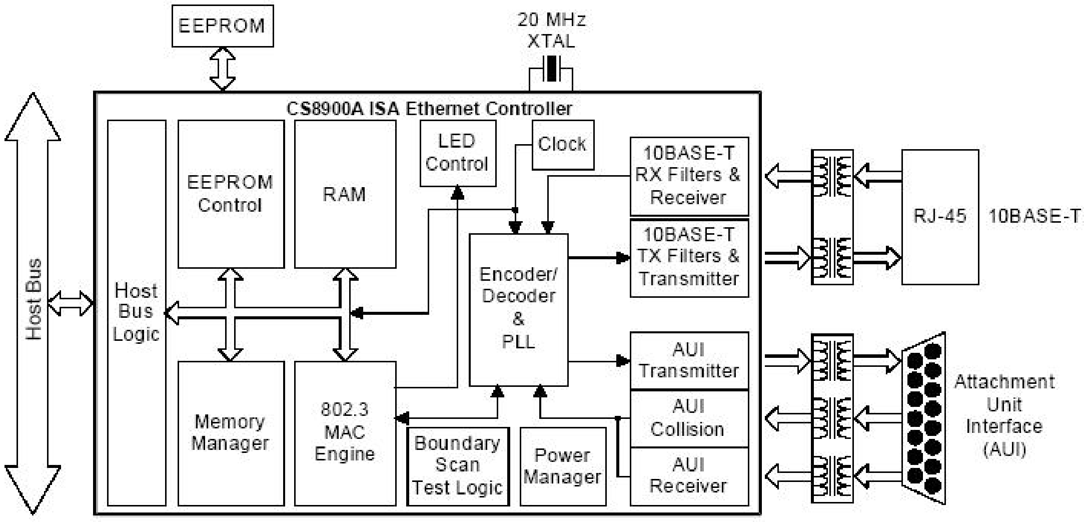

Figure 3.11. CS8900A LAN Controller (Cirrus Logic, 2001)

In addition to high integration, the CS8900A offers a broad range of performance features and configurationoptions. Its unique PacketPage architecture automatically adapts to changing network traffic patterns and available system resources. The result is increased system efficiency (Cirrus Logic, 2001).

The CS8900A is available in a 100-pin LQFP package ideally suited for small form-factor, cost-sensitive Ethernet applications. With the CS8900A, system engineers can design a complete Ethernet circuit that occupies less than 1.5 square inches of board space (Cirrus Logic, 2001).

Usage of the GL850A on the board is shown in Fig.3.6.

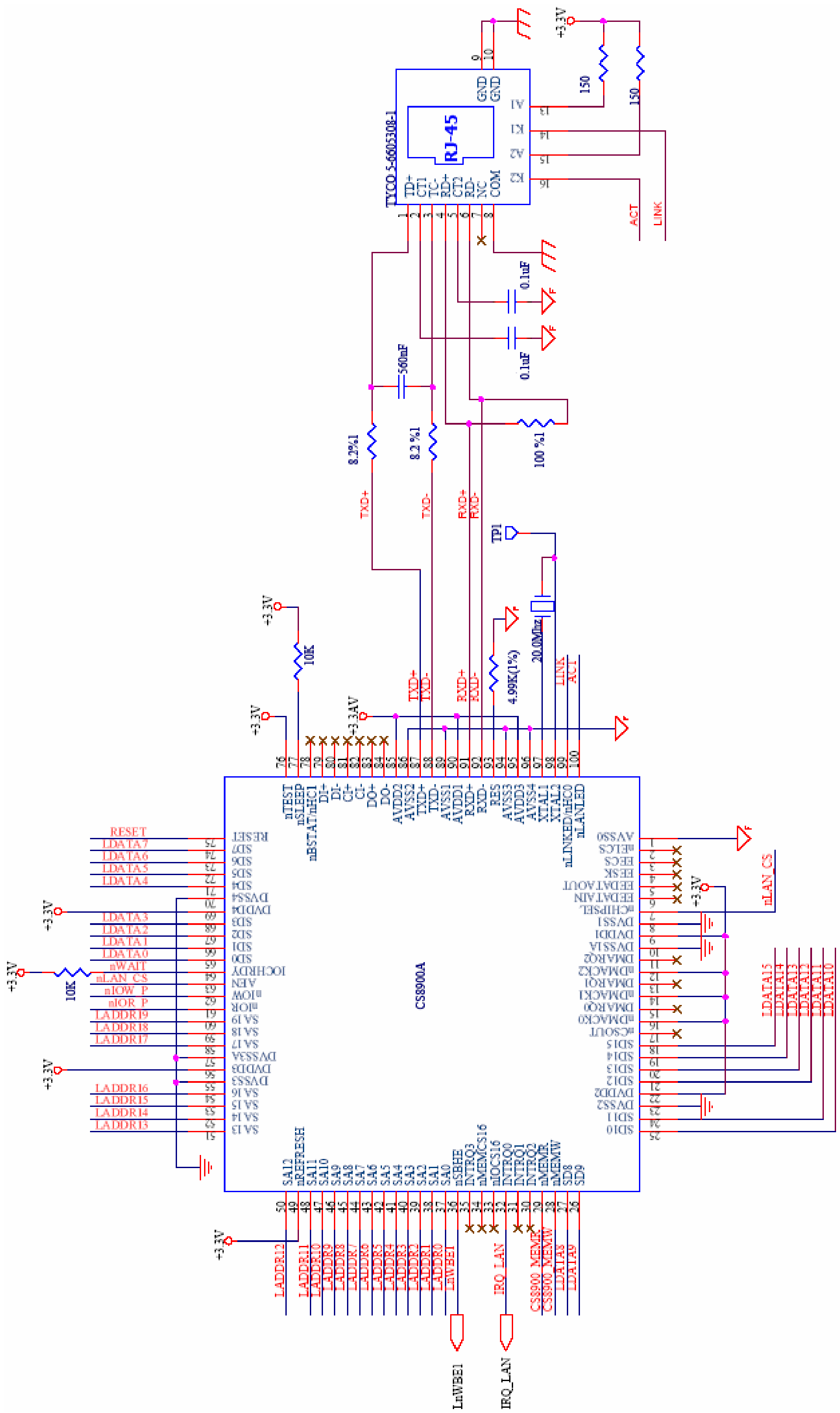

Figure 3.12. The Circuit of the CS8900A

3.1.7. TFT-LCD Module with Touch Screen

This module is a color active matrix LCD module incorporating amorphous silicon TFT(Thin Film Transistor). It is composed of a color TFT-LCD panel, driver ICs, Input FPC and a back light unit.

Graphics and texts can be displayed on a 480x3x272 dots panel with about 16 million colors by supplying 24 bit data signals (8bitxRGB). Four timing signals, logic (typ. +2.5V), analog (typ. +5V) supply voltages for TFT-LCD panel driving and supply voltage for back light (Sharp Microelectronic, 2005).

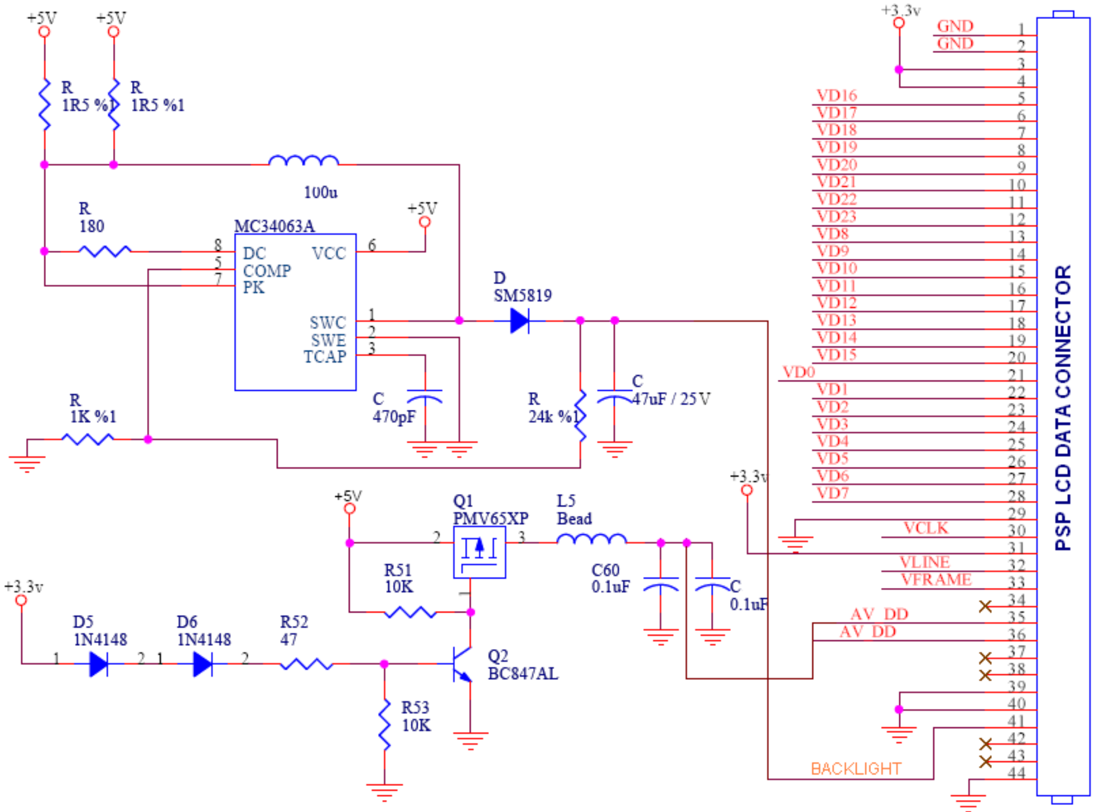

Figure 3.13. The Circuit of the TFT-LCD Module

Usage of the Step-Up/Down/Inverting Switching Regulator for the TFT-LCD and for the backlight on the board is shown in figure 3.13.

3.2. The Software of the Single Board Computer (SBC)

The operating system used on the board is Microsoft Windows CE 5.0. Hardware Layer and three layers are added on the board. These are,

- OEM (Original Equipment Manufacturer) Layer
- Operating System Layer
- Application Layer (Microsoft Developer Network(MSDN), 2010)

3.2.1. OEM Layer

General an OEM Layer is shown in figure 3.14. below.

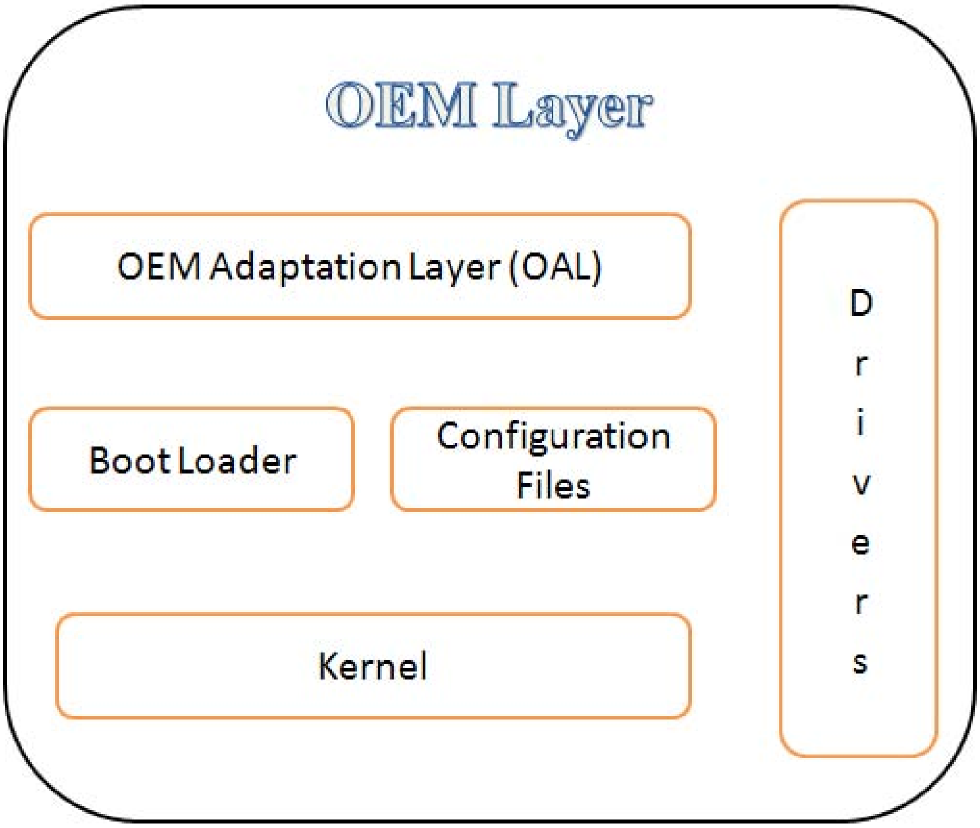

Figure 3.14. OEM Layer

- **OEM Adaptation Layer (OAL)**

An OEM adaptation layer (OAL) is a layer of code that logically resides between the Windows CE kernel and the hardware of the target device. Physically, the OAL is linked with the kernel libraries to create the kernel executable file. The OAL facilitates communication between the operating system (OS) and the target device and includes code to handle interrupts, timers, power management, bus abstraction, generic I/O control codes (IOCTLs), and so on (MSDN, 2010).

OEM Adaptation Layer files is shown below in figure 3.15.

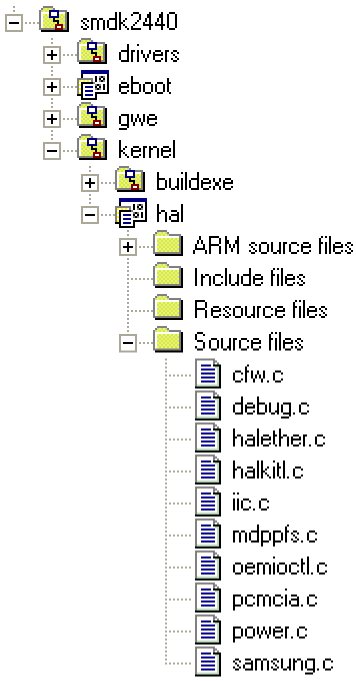

Figure 3.15. OEM Adaptation Layer Files

Creating the OAL is one of the more complex tasks in the process of getting a Windows CE–based OS to run on a new hardware platform. In general, the easiest way to create an OAL is to copy the OAL implementation from a working OS design, and then modify it to suit the specific requirements of the hardware platform (MSDN, 2010).

- **Boot Loader**

The boot loader is a utility that is an integral part of the OEM device development process. In some cases, it is also included in the final OEM product. The general purpose of the boot loader is to place the run-time image into memory, and then jump to the OS startup routine. The boot loader can obtain the run-time image in a number of different ways, including loading it over a cabled connection, such as Ethernet, a universal serial bus (USB), or serial connection (MSDN, 2010).

The Boot Loader files is shown below in figure 3.16.

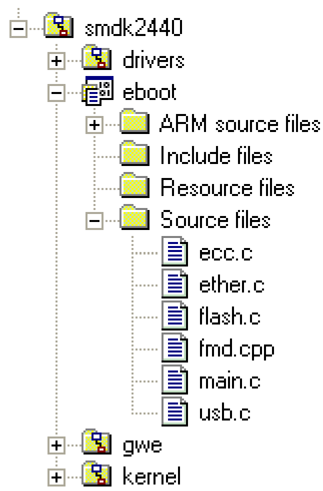

Figure 3.16. Boot Loader Files

The boot loader also loads the OS from a local storage device, such as Compact Flash, or a hard disk. The boot loader might store the run-time image in RAM or in nonvolatile storage, such as flash memory, electrically erasable programmable read only memory (EEPROM), or some other storage device for later use (MSDN, 2010).

- **Configuration Files**

The configuration files of the board using in this study are shown in figure 3.17. For example Config.bib file applies to the OS image. It contains Memory and Config sections for the OS image (MSDN, 2010)

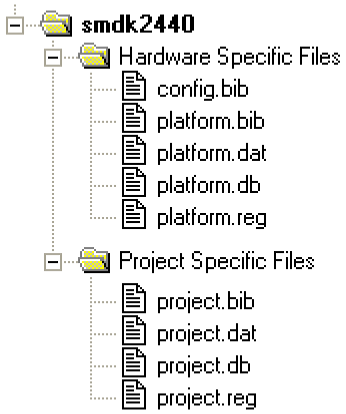

Figure 3.17. Configuration Files

The files that Platform Builder uses during the build process can be separated into two general categories of configuration files as source code configuration files and image configuration files. Source code configuration files are used by the Build Tool to build modules and features and to build the source code for the run-time image. Run-time image configuration files are used by a number of tools that are called by the Make Binary Image Tool to create the run-time imag (MSDN, 2010).

When the Build tool locates a sources file in the current directory, it calls the Nmake tool (Nmake.exe), which compiles the specified C or C++ sources file or links an object module, according to the linking rules contained in the makefile file (MSDN, 2010).

- **Drivers**

A device driver is software that abstracts the functionality of a physical or virtual device. A device driver manages the operation of these devices. Examples of physical devices are network adapters, timers, and universal asynchronous receiver-transmitters (UARTs). An example of a virtual device is a file system. Implementing a device driver allows the functionality of the device to be exposed to applications and other parts of the operating system (MSDN, 2010)

While developing a device driver, take advantage of the services provided by the OS. Although Windows CE device drivers are trusted modules, they do not have to run in kernel mode (MSDN, 2010)

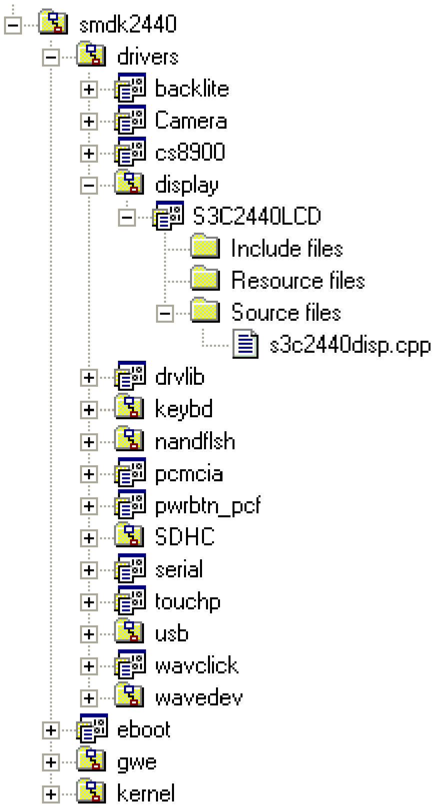

Figure 3.18. Hardware Drivers

The all drivers used in this study are shown above in figure 3.18.

3.2.2. Operating System Layer

The part of an Operation System Layer is shown in figure 3.19.

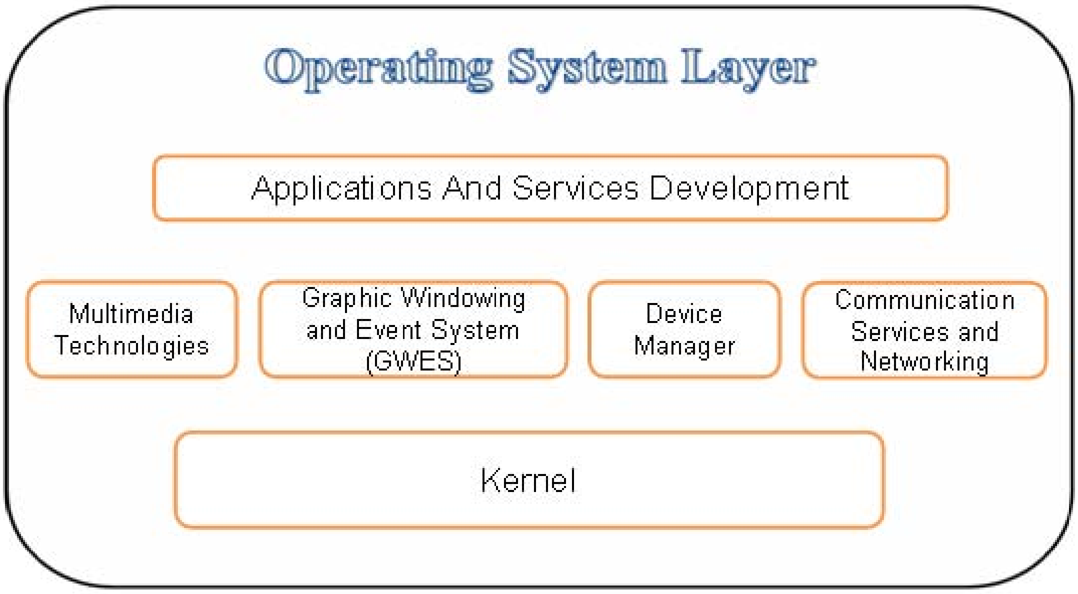

Figure 3.19. Operating System Layer

- **Application and Services Development**

The application and services using in this study is shown in figure 3.20. The .NET Compact Framework is a hardware-independent program execution environment for secure downloadable applications optimized for resource-constrained computing target devices MSDN, 2010).

It offers a choice of languages, initially Microsoft Visual Basic and Microsoft Visual C#, and eliminates the common problems faced with language interoperability (MSDN, 2010).

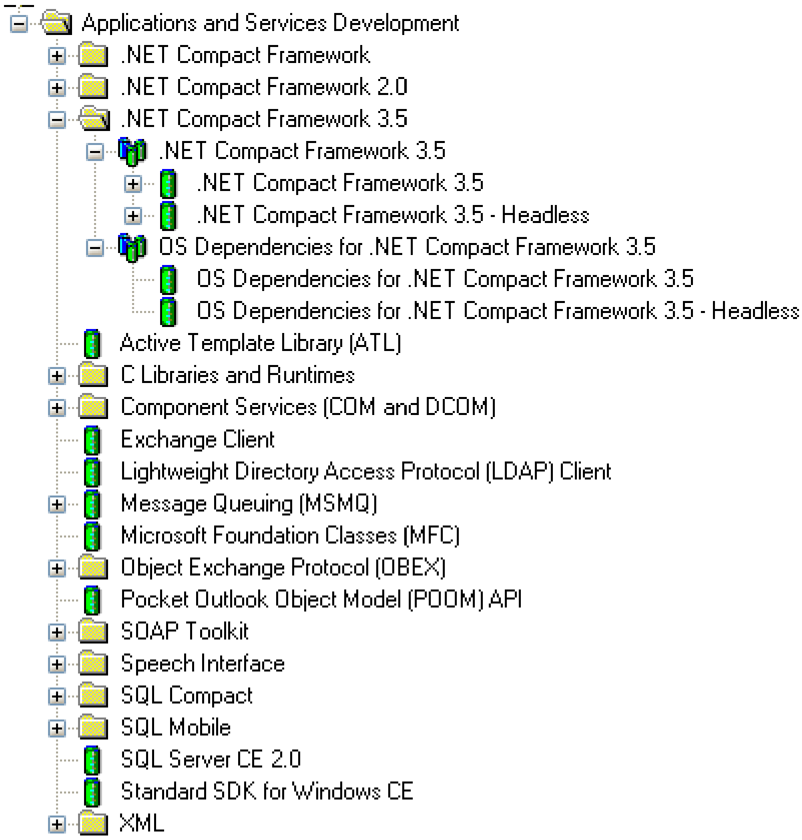

Figure 3.20. Application and Services Development

- **Multimedia Technologies**

In this section provides information about Windows CE audio technologies. These technologies provide support for waveform audio playback and capture. The Graphics provides information about Windows CE graphics technologies. These technologies provide levels of performance beyond that of the Graphics Device Interface (GDI) by providing low-level access to audio and video hardware in a device-independent manner. The Media provides information about the Windows CE technologies used to support playback for a variety of encoded audio and video data formats. Subfolders is shown in figure 3.21 (MSDN, 2010)

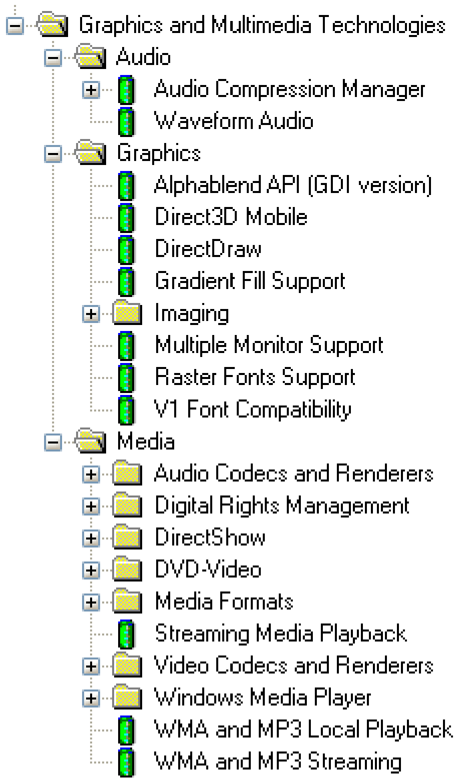

Figure 3.21. Multimedia Technologies

- **Graphic Windowing and Event System (GWES)**

Microsoft Windows CE combines the Microsoft Win32 application programming interface (API), user interface (UI), and graphics device interface (GDI) libraries into the Graphics, Windowing, and Events Subsystem (GWES) module (Gwes.exe). GWES is the interface between the user, the application, and the operating system (MSDN, 2010).

GWES supports all the windows, dialog boxes, controls, menus, and resources that make up the Windows CE user interface (UI), which enables users to control applications. GWES also provides information to the user in the form of bitmaps, carets, cursors, text, and icons (MSDN, 2010).

Graphics, Windowing and Events are shown in figure 3.22.

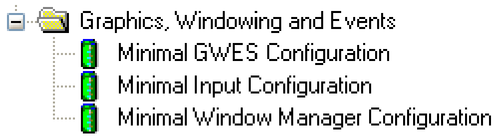

Figure 3.22. Graphic Windowing and Event System

- **Device Management**

Device Management is one of the key requirements for deploying target devices in an enterprise scenario. Managing target devices involve being able to remotely distribute software, keep track of software and hardware, manage inventory, and configure target devices. Methods of installing files or applications on Microsoft Windows CE-based devices that do not use Device Management functionality often require user interaction and offer no way of keeping a target device updated over time, nor do they offer an administrator an easy way to centrally distribute software, and manage hardware and software MSDN, 2010).

The Device Management is shown in figure 3.23

Figure 3.23. Device Management

- **Communication Services and Networking**

Microsoft Windows CE provides networking and communications capabilities that enable devices to connect and communicate with other devices and people over both wireless and wired networks. Windows CE also includes wireless features such as Bluetooth, 802.11 (802.1x, Extensible Authentication Protocol and 802.11 automatic configuration), and Media Sense; server features such as Remote Access Service (RAS)/ Point-to-Point Tunneling Protocol (PPTP) and File Transfer

Protocol (FTP) Servers, and services and APIs such as the RTC Client API, Winsock 2.2, and Object Exchange Protocol (OBEX) as shown in figure 3.24 (MSDN, 2010).

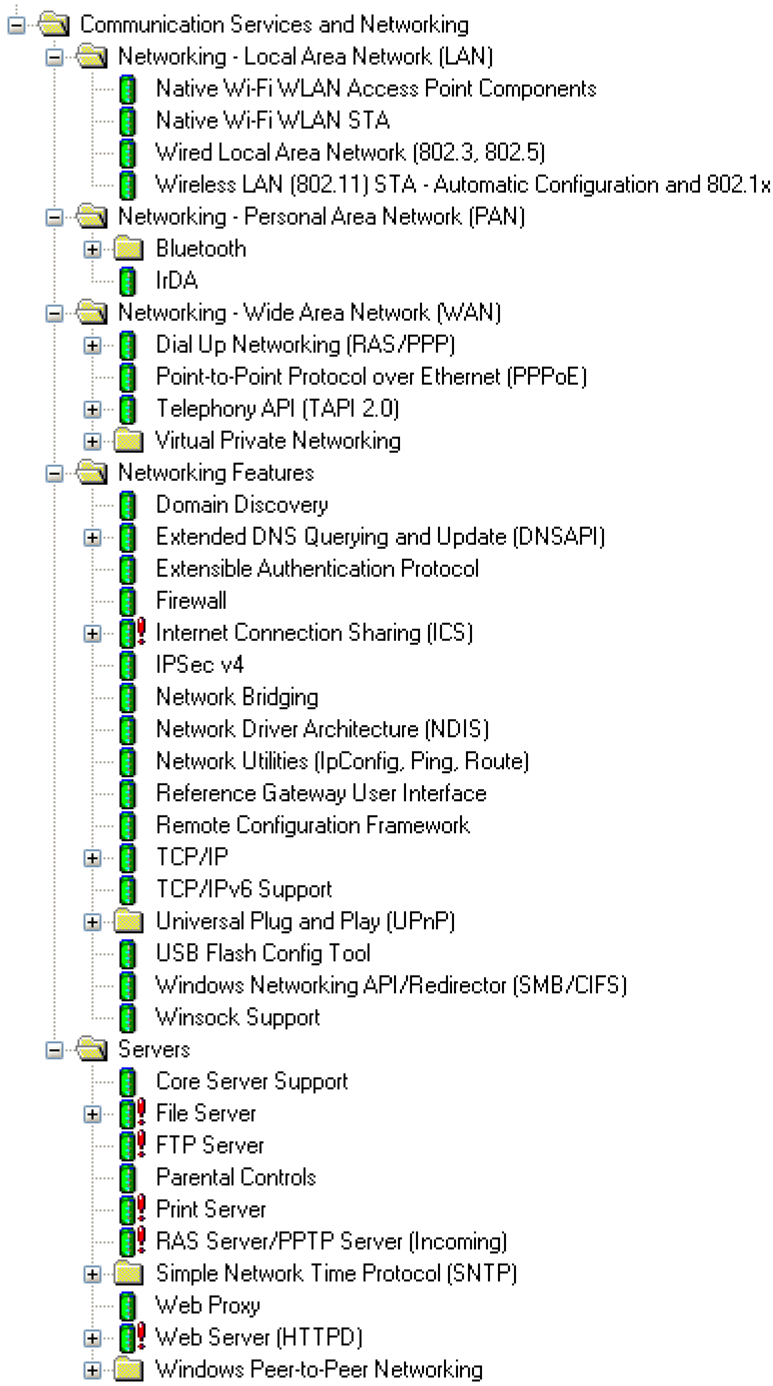

Figure 3.24. Communication Services and Networking

- **Kernel**

The kernel, which is represented by the Nk.exe module, is the core of the Microsoft® Windows® CE operating system (OS). The kernel provides the base OS functionality for any Windows CE–based device. This functionality includes process, thread, and memory management. The kernel also provides some file management functionality. Kernel services enable applications to use this core functionality (MSDN, 2010).

The Kernel Features are shown in figure 3.25.

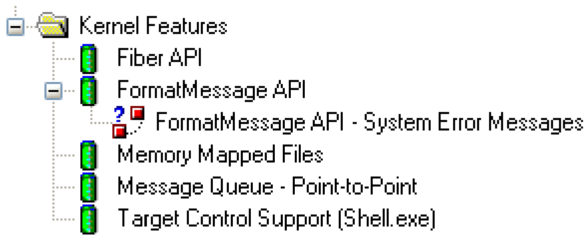

Figure 3.25. Kernel

A fiber API is a unit of execution that must be manually scheduled by the application. Fibers run in the context of the threads that schedule them. Each thread can schedule multiple fibers. In general, fibers do not provide advantages over a well-designed multithreaded application. However, using fibers can make it easier to port applications that were designed to schedule their own threads (MSDN, 2010).

A memory-mapped file, or file mapping, is the result of associating file contents with a portion of the virtual address space of a process. It can be used to share a file or memory between two or more processes. Windows Embedded CE supports named and unnamed file-mapping objects. Unnamed files provide a method for inter process communication and a way to allocate virtual memory regions larger than the 2 GB process size limit (MSDN, 2010).

Target Control service is used to transfer files to a target device, to access files on the development workstation from a target device, and to debug applications.

When Target Control is enabled, Platform Builder includes the shell tool, shell.exe, in the run-time image (MSDN, 2010).

3.2.3. Application Layer

The part of an Application Layer is shown in figure 3.26.

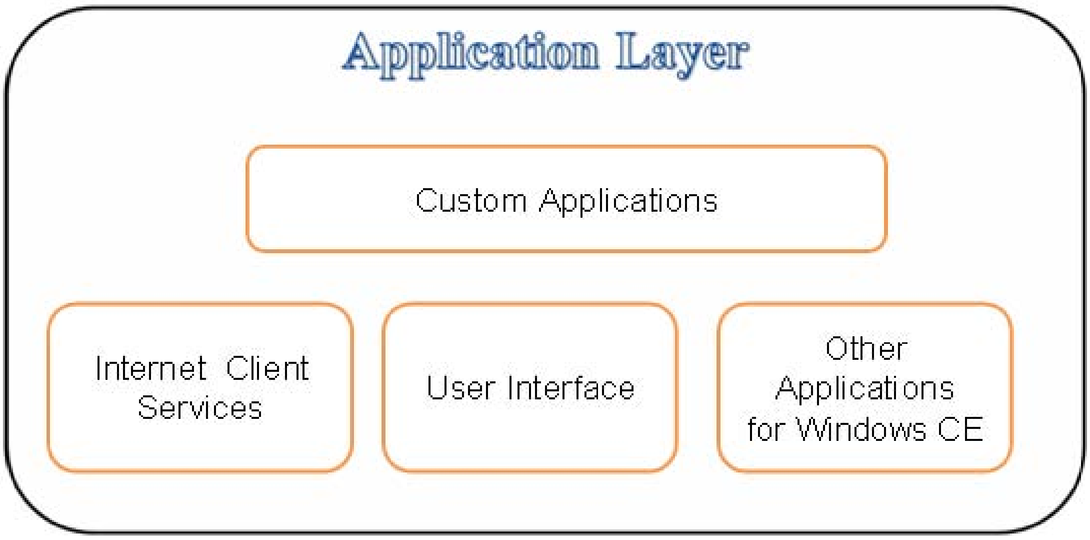

Figure 3.26. Application Layer

- **Custom Application**

This section describes the operating system functionality that is available for developing end user applications. For example ActiveSync provides support for synchronizing data between a Windows-based desktop computer and Microsoft Windows CE-based devices (MSDN, 2010).

All application of the Platform Builder updated until January 2010 are shown in figure 3.27.

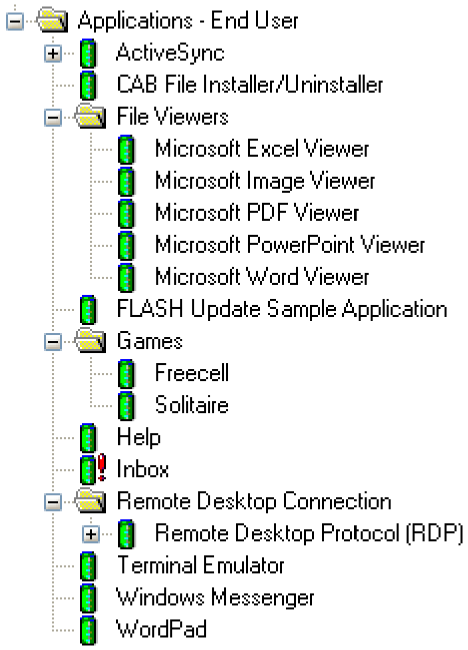

Figure 3.27. Custom Application

- **Internet Client Services**

Internet Client Services provide support for browser applications, technologies that enable to create custom browsers, and run-time engines for parsing and translating scripting languages (MSDN, 2010). The Internet Client Services are shown in figure 3.28.

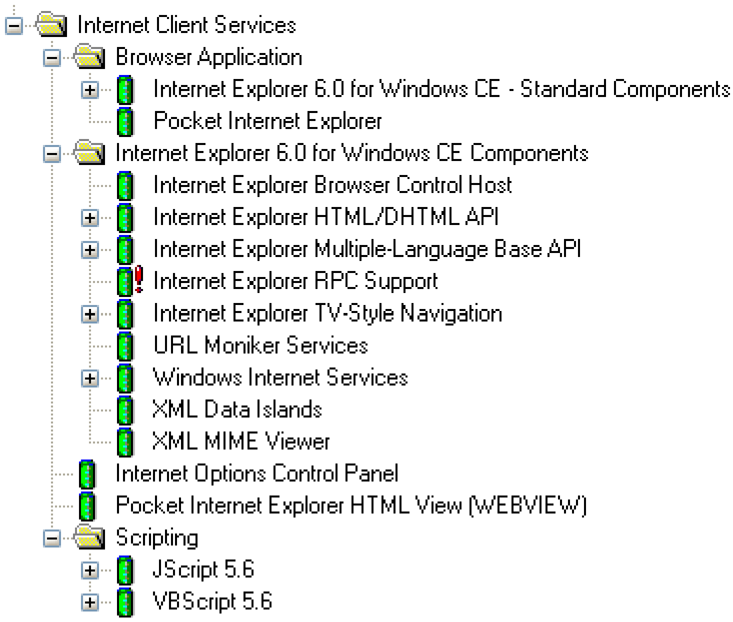

Figure 3.28. Internet Client Services

- **User Interface**

The User Interface in Microsoft Windows CE 5.0 consists of ways that a user can interact with a Windows CE-based device and its applications. For example the Stylus provides information on interacting with a Windows CE-based application by using a stylus and a touch screen. The stylus and the touch screen provide a direct and intuitive alternative to a Mouse (MSDN, 2010).

The User Interface is shown in figure 3.29.

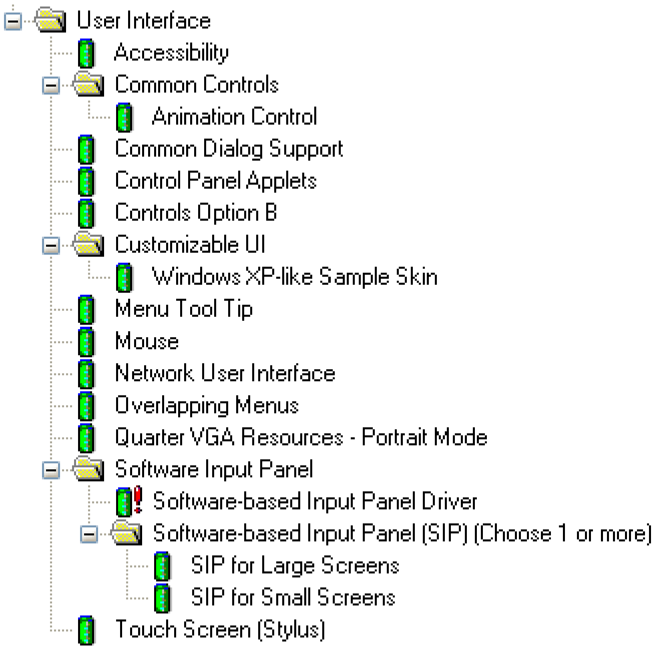

Figure 3.29. User Interface

- **Other Applications for Windows CE**

These applications can be developed using embedded Visual C++ together with Platform Builder. Firstly the user must use Platform Builder to create an OS design, build a run-time image, and then download the run-time image to the target device. When the run-time image runs, Platform Builder communicates with the target device over a kernel transport. To develop an application, it is required to keep Platform Builder connected to the target device, and then run embedded Visual C++ (MSDN, 2010).

It is also possible using Visual Studio 2005 or 2008 to operate on Windows CE for improving applications.

4. RESULT AND DISCUSSIONS

In this thesis Single Board Computer designed is shown in Figure 4.1. Both the internet connection is planned as the wireless and the power unit is planned with the battery because of usage as the VoIP phone.

Figure 4.1. Single Board Computer with S3C2440 Microcontroller

The internet connection can be obtained adding the driver of the USB Wireless Adapter to Board Suppose Package. As a result of adding this driver, no extra cabling nor RJ45 connector is required for the internet and IP network connection.

Our SBC is also used as the wireless such as PDA or other hand devices as shown on the photos above. The power unit is designed using Lithium-Polymer Battery. In the normal condition the power dissipation of the board is approximately 250 mA on 11V. A Li-Po battery 2000mAh, 11.1 V, was the sensible solution for using a long duration wireless connection.

In figure 4.2. the web site of Cukurova University is shown on the board above. Here the board is connected to the internet using an USB Wireless Adapter.

A media player application on the playing is shown in figure 4.3.

Figure 4.2. Internet Connection Using an USB Wireless Adapter

Figure 4.3. Media Player Using

Since the VoIP communications as well as the required of Multimedia features are considered in the planning, a 400 MHz processor is selected. This speed is enough for both VoIP and other multimedia feature such as media playing or on the internet surfing.

On the board a USB Flash Memory Drive and a SD Memory Card Unit is also used to increase multimedia capabilities. Therefore, 64 MB Flash ROM and 64 MB SDRAM that is designed as a data storage capabilities of the board has been expanded with SD card and flash disk. In figure 4.4.,USB Flash Memory Disk as Hard Disk, also SD Memory Card as Storage Card is shown.

Figure 4.4. USB Flash Memory Disk and SD Card

SJPhone named the VoIP application is used as VoIP phone. Our rationale in choosing this application is the required support for different platforms such as the Windows Desktop, Windows Mobile, Pocket PC's and Linux.

There are numerous versions of the VoIP test application as open source. These applications in shares over the internet is still being developed by the programmers. The SJPhone applications which are running on SBC, on the PC and on the LG Mobile Phone. After network adjustments all of the aplications receive an IP address from DHCP server. Then the lists seen by each application is shown Figure 4.5, 4.6 and 4.7.

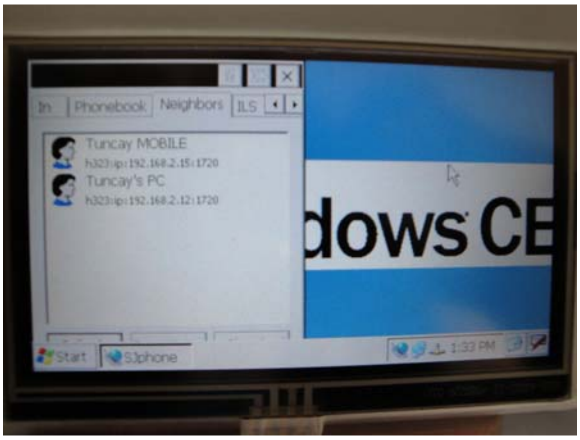

Figure 4.5. SJ is ready on our SBC

Figure 4.6. SJ is ready on PC

Figure 4.7. SJ is ready on Mobile Phone

Selecting the application from the Neighbours list or typing the selected applicant's IP number starts a SIP meeting as shown figure 4.8 and figure 4.9.

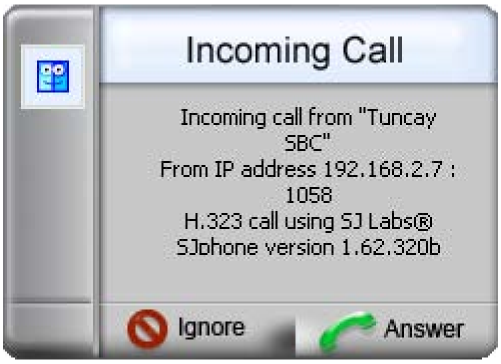

Figure 4.8. From SBC to PC SIP request

Figure 4.9. VoIP started

On the result of this thesis:

- Designed a Single Board Computer hardware with 32-bit ARM Microprocessor

- Prepared OEM Layer softwares using Embedded Visual C++

- Worked the Hardwares
    - 4,3” LCD with touch screen
    - GL850A USB Hub Controller
    - CS8900A LAN Controller
    - Wireless Adapter via USB ports
    - UDA1341TS Audio Codec

- Worked the Windows CE 5.0 Operating System
    - Internet Explorer
    - Media Player
    - File Viewers etc

- Applied VoIP (SIP) communication using an open-source VoIP application
    - Between our SBC and PC
    - Between our SBC and a Mobile Phone

5. CONCLUSIONS

In this study, it is pointed out the voice communication over IP using a SBC with 32 bit-microprocessor. At the end of the study, the desirable aim will be achieved. In the present day, there have been some LAN and IP network in the job environment. It will be reasonable to utilize these networks not only to exchange data but also communicate , by via of this study.

Espacially, it will be cost effective for the organizations to communicate over the IP by using a kind of mobile phone we handle in the tests, especially rather than the SBC used in this study. Furthermore, in the developing countries, taking most of the new LAN and IP networks into consideration, infrastrucure will be ready to communicate over the IP in wireless if designed for the main communication.

REFERENCES

ARM Architecture Reference Manual, 2005.

ANDREWS, J. R., 2005. Co-verification of hardware and software for ARM SoC design. Elsevier Press, Amsterdam:287.

AXELSON, J., 2003. Embedded ethernet and internet complete – Designing and programming, small devices for networking. Lakeview Research LLV, Madison, USA:498.

BALL, S.R., 2002. Embedded microprocessor systems real world design. Newnes Press, Australia:432.

CAMARILLO, G., 2002. SIP Demystified. The McGraw-Hill Companies, New York, USA:285.

DALGIC, I., BORELLA, M., DEAN, R., GRABIEC, J., MAHLER, J., SCHUSTER, G., SIDHU, I., 1999. True number portability and advanced call screening ana SIP-based IP phone system. IEEE Communications Magazin, 7:96-101.

HARDY, W.C., 2003. Voip service quality. The McGraw-Hill Companies, New York, USA:338.

IVOV, E., NOEL, T., 2004. Optimizing SIP application layer mobility over IPv6 using layer 2 triggers. IEEE Electronics and Communication Engineering Journal, 7:1-5.

NARVAEZ, L., PEREZ, J., GARCIA, C., CHI, V., 2007. Designing 802.11 WLANs for VoIP and data. IJCSNS International Journal of Computer Science and Network Security, 7(7):249-256.

NOERGAARD, T.,2005. Embedded systems architecture – A comprehensive guide for engineers and programmers. Elsevier, USA:657.

SAMSUNG Electronics, 2004. S3C2440A 32-bit RISC microprocessor user’s manual:560.

SHEPARD, S., 2005. Voice over IP crash course. The McGraw-Hill Companies, New York, USA:408.

SLOSS, A. N., SYMES, D., WRIGHT, C., 2004. ARM system developer's guide, designing and optimizing system software. Elsevier, USA:704.

VAHID, F., GIVARGIS, T., 1999. Embedded system design: A unified hardware and software approach. University of California, USA:103.

WARD, M., 2005. Developing video phones with ARM processor-based solutions. Technology in-depth 4(4):16-18

ZANG, Y., 2002. SIP based Voip network and its interworking with PSTN. IEEE Electronics and Communication Engineering Journal, 12:274-282.

ZHOU, J., LI, J., XIA, Y., B., CAI, B., YING, C., 2008. SIP network discovery by using SIP message probing. IEEE Electronics and Communication Engineering Journal, 7:791-794.

ZURAWSKI, R., 2006. Embedded systems handbook. Taylor and Francis Group, London:1089.

CURRICULUM VITAE

The Author was born in Antakya / Turkey, on October 22, 1973. He was graduated from Antakya High School, Antakya, Turkey, in 1990. He has received BS diploma in Electrical & Electronic Engineering from the Department of Electrical and Electronics Engineering, University of Selcuk, Konya, Turkey, in 1993-1994 Spring Semester. He is married and has a son who is seven years old. He is currently working as a lecturer for Osmaniye Korkut Ata University, Kadirli Technical High School. His main research interests are microprocessors, microcontrollers, programmable logic control systems, electronic circuits and system modelling.