

공개 소스 소프트웨어 기반의 VoIP 서비스를 위한 무선단말 개발

서 효 중[†] · 이 병 호^{**} · 김 태 현^{***}

요 약

본 논문에서는 공개 소스 소프트웨어를 기반으로 VoWLAN (Voice over WLAN) 단말 시스템을 개발하였다. 이 시스템은 무선랜에 접속할 수 있는 환경에서 IP-PBX 서버를 통해 VoWLAN 단말들 간에 VoIP 서비스를 제공하며 다음과 같은 특징을 가진다. 첫째, 공개 소스 소프트웨어 기반으로 작성하여 시스템 도입에 따른 초기 투자비용의 부담이 적다. 둘째, 소프트웨어 IP-PBX 시스템과 연동하여 voice mail, conference call, interactive voice response와 같은 다양한 부가 서비스를 제공한다. 셋째, VoWLAN 단말은 임베디드 환경에 적합한 경량 소프트웨어 솔루션들을 이용하여 제한된 시스템 자원 하에서 고수준의 사용자 응용프로그램을 제공한다. 넷째, 공개 소스 소프트웨어 기반으로 작성되어 요구 사항의 변화에 따라 추가 기능의 확장이 용이하다.

키워드 : VoIP, VoWLAN 단말, IP-PBX 서버, 임베디드 시스템, uClinux

Development of a VoWLAN Terminal based on Open Source Software

Hyo-Joong Suh[†] · Byunggho Lee^{**} · Taehyoun Kim^{***}

ABSTRACT

In this paper, we developed a VoWLAN (Voice over WLAN) system based on an open source software. The system aims to provide VoIP service over wireless LAN with an IP-PBX server. The features of system presented in this paper are as follows. First, the initial cost for the development is reduced since the system is developed based on open source software. Second, the system provides various additional services such as Voice Mail, Conference Call, and Interactive Voice Response with a software IP-PBX server. Third, the VoWLAN terminal provides high-level user applications with minimal system resources using lightweight open software solutions. Finally, it is highly scalable since it is based on the open source software.

Key Words : VoIP, VoWLAN terminal, IP-PBX server, Embedded System, uClinux

1. 서 론

최근 인터넷 사용자의 증가와 안정적인 인터넷 인프라가 보급됨에 따라 이를 기반으로 음성이나 영상을 전송할 수 있는 다양한 통신 응용이 소개되고 있으며, 이러한 응용의 대표적인 예로 VoIP (Voice over Internet Protocol)[1] 기술이 있다. VoIP는 인터넷망을 통해 음성신호를 실시간으로 전송하는 기술로, 기존 회선교환 방식의 일반전화와 달리 인터넷의 근간인 IP(Internet Protocol) 네트워크를 통해 음성을 패킷 형태로 전송한다.

VoIP 기술을 음성통화에 이용할 경우의 장점은 다음과 같다. 첫째, 인터넷망을 통해 전화통화 서비스를 이용하므로

국제전화나 장거리전화 같은 고가의 전화서비스를 저렴하게 이용할 수 있다. 둘째, 기존의 전화서비스는 회선교환방식으로 회선 사용에 따른 비용이 높은 반면에 VoIP는 인터넷 패킷 교환 방식에 기반하므로 회선의 독점을 막고 회선을 효율적으로 사용할 수 있다. 셋째, 인터넷망은 기본적으로 데이터 전송을 위해 만들어졌으므로 이를 통해 메신저, 화상통화와 같은 다양한 부가 서비스를 제공할 수 있다. 넷째, 신규 서비스 수요가 발생할 경우 기존의 공중전화망은 물리적인 회선 증설이 불가피하지만 VoIP는 새로운 IP를 할당하거나 네트워크 대역폭을 늘림으로써 시스템의 확장이 용이하다.

VoIP 기술은 1995년 이스라엘의 보칼텍(Vocaltec)에 의해 최초로 서비스가 시작된 후, 최근에 무선랜 (WLAN: Wireless LAN) [2] 기술이 대중화됨에 따라 VoIP 단말에 Wi-Fi (Wireless Fidelity)를 적용한 VoWLAN (Voice over WLAN) 단말의 형태로 발전해가고 있다. VoWLAN 단말은 무선 AP(Access

* 이 논문은 2006년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

† 종신회원: 가톨릭대학교 컴퓨터정보공학부 조교수

** 준 회원: 가톨릭대학교 컴퓨터공학과 석사과정

*** 종신회원: 서울시립대학교 기계정보공학과 조교수 (교신저자)

논문접수: 2007년 5월 23일, 심사완료: 2007년 7월 13일

Point)가 설치된 장소에서 휴대폰과 같은 이동성을 제공하면서 저렴하게 전화 및 데이터 서비스를 이용할 수 있는 장점을 제공한다.

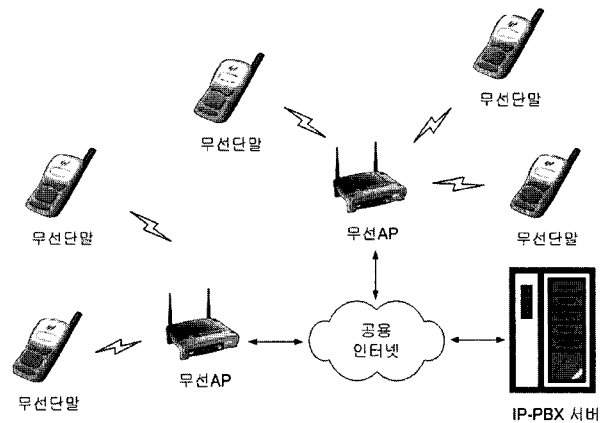
기존의 VoWLAN 단말에 관한 연구들의 경우 단말의 소프트웨어 구조나 기능사항에 대한 내용보다 제한된 채널 용량을 감안한 품질제어[3, 4]나 전력소모제어[5, 6]에 관한 내용을 주로 다루고 있다. 그러나, 현재 VoIP 단말 및 IP-PBX 장비의 도입에 따른 초기 투자비용이 VoIP 시스템 확산에 가장 큰 걸림돌로 작용하고 있는 점을 감안하여 본 논문에서는 공개 소프트웨어를 기반으로 무선랜과 결합하여 VoIP 기능과 다양한 응용 프로그램을 포함하는 VoWLAN 단말을 개발하고, 공개 소스 IP PBX 서버를 통해 다양한 음성 부가서비스를 제공하도록 하였다.

본 논문에서 제안하는 시스템의 특징은 다음과 같다. 첫째, 공개 소프트웨어 기반으로 개발되어 시스템 도입에 따른 초기 비용이 적다. 둘째, 소프트웨어 IP-PBX 시스템과 연동을 통해 다양한 음성 부가서비스를 제공한다. 셋째, 내장형 시스템의 제한된 시스템 자원을 고려한 경량 솔루션을 활용하여 고수준의 응용프로그램을 제공한다. 넷째, 공개 소프트웨어 기반으로 작성되어 추후 확장이 용이하다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 시스템의 전체 구성과 단말의 주요 기능에 대해 설명한다. 3장에서는 시스템의 구현 환경과 내용을 설명하고, 4장에서는 듀얼 코어 SoC 상에서의 성능평가를 수행한다. 마지막으로 5장에서 논문의 결론과 향후 연구 과제를 제시한다.

2. VoWLAN 시스템 구성 및 기능

(그림 1)은 VoWLAN 단말, 무선 AP, IP-PBX서버로 구성된 VoWLAN 시스템의 전체 구성을 보여준다. VoWLAN 단말은 무선 AP를 통해 인터넷 망과 연결되고 이를 통해 사용자는 IP-PBX서버에 접속하여 기본 음성통화와 voice



(그림 1) VoWLAN 시스템 구성

mail, conference call, interactive voice response와 같은 다양한 부가 음성서비스를 이용하게 되며 인터넷망을 통해 다양한 데이터서비스를 이용할 수 있다.

본 논문에서 제안하는 VoWLAN 단말은 기본 VoIP 기능 외에 다양한 부가서비스를 제공하며, 각 서비스의 자세한 내용은 <표 1>과 같다.

3. 시스템 구현

본 논문에서 제안한 VoWLAN 시스템의 개발은 크게 VoWLAN 단말과 IP-PBX 서버로 구분된다.

3.1 VoWLAN 단말

3.1.1 개발 환경

VoWLAN 단말은 임베디드 리눅스의 일종인 uClinux[7]를 GCT사의 Wi-Fi SoC인 GDM5104 프로세서[8]에 탑재하여 구현하였다. 개발보드에서 동작할 커널과 루트파일 시스템 이미지 생성과 응용 프로그램 개발을 위해 GDM5104

<표 1> VoWLAN 단말의 기능

구분	기능
VoIP	음성통화: IAX 시그널링 프로토콜을 통한 VoIP 기능을 제공 Music On Hold: 통화대기시 백그라운드로 발신자의 단말에 음악을 재생 Conference Call: 3자 이상 사용자 동시 통화 Voice Mail: 메일박스에 녹음된 음성 메시지 녹음 및 재생 Interactive Voice Response: 시스템의 안내 음성에 따라 메뉴 선택 후 서비스 제공 Call Parking: 통화를 지정된 위치의 다른 전화로 전달하고 지정된 전화기로 통화
인터넷 응용	MSN 메신저: 무선 단말에 포함된 MSN 메신저 프로그램을 통해 메시지 교환 파일 전송: 무선 단말에 저장된 파일을 무선 랜을 통해 단말과 단말 간에 전송 이메일 수신: POP3 메일서버에 접속해서 이메일 수신 여부 및 내용 확인 Samba: 무선 단말에 Samba 서버를 구성하여 무선 단말을 이동형 스토리지로 사용
게임	Tetris, Invader, iPobble, BlackJack, Chopper, Snake, Light, Mine sweeper, Othello, Pong, Steroids, Tic-Tac-Toe, Tunnel, TuxChess, Vortex
파일 브라우저	Ebook Reader: 텍스트 파일을 LCD화면을 통해 출력 Ebook Voice Reader: 텍스트 파일을 음성합성 엔진을 통해 음성으로 합성하여 재생 Text Editor: 텍스트 파일을 생성 또는 수정 Image Viewer: JPG, BMP 등의 이미지 파일을 LCD화면을 통해 출력

프로세서용 GNU 기반 교차 개발 환경과 임베디드 환경에 적합한 경량 라이브러리인 uClibc[9]를 각각 사용하였다.

본 연구에서는 개발의 효율성을 높이기 위해 개발서버에 크로스 컴파일 환경을 구축하고, SSH와 Samba서버를 통해 다수의 개발자가 원격 접속하여 개발할 수 있는 환경을 구축하였다. 또한, 전체 이미지 생성을 위한 스크립트를 작성하여 복잡한 컴파일 과정을 쉽게 수행할 수 있도록 하고, 컴파일 과정에서 출력되는 모든 메시지에 대해 로그 파일을 생성하여 디버깅에 이용하였다.

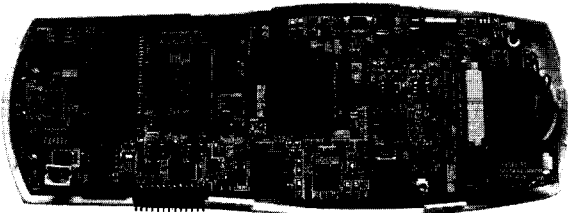
3.1.2 하드웨어 구성 요소

본 연구에서는 (그림 2)와 같이 지씨티 사의 GSK5401 개발보드를 하드웨어 플랫폼으로 사용하였으며, 보드의 세부 구성은 <표 2>와 같다. 본 연구에서 사용한 GDM5104 프로세서는 무선 멀티미디어 통신에 적합한 저전력 임베디드 프로세서로 일반적인 연산을 위한 RISC 코어, 멀티미디어 연산을 위한 DSP 코어, 무선 랜, 스토리지, 오디오, 비디오 인터페이스를 포함하는 SoC이다.

개발을 위한 입력 인터페이스로 JTAG과 UART가 지원이 되며 JTAG 인터페이스를 통해 부트로더 이미지를 플래시 메모리에 쓸 수 있고, UART 인터페이스를 통해 커널 및 루트파일시스템 이미지를 메모리에 다운로드할 수 있다.

3.1.3 소프트웨어 구성 요소

(그림 3)은 본 논문에서 제안한 VoWLAN 단말의 소프트



(그림 2) GSK5401 개발보드

<표 2> GSK5401 하드웨어 구성

구분	사양
System	• GCT GSK5401R4
Processor	• GCT GDM5104 - 32-bit RISC processor (120MHz) - 32-bit DSP (96MHz) - Cache: 16KB I-Cache, 8KB D-Cache
SDRAM	• SAMSUNG K4S561632C (256Mbit)
NOR FLASH	• Intel TE28F640 (64Mbit)
WLAN	• GCT GRF5104 (IEEE 802.11 b/g)
LCD	• D-II28064CB-01(128 × 64)
Audio	Wolfson WM8974 Mono 모노 코덱
Key PAD	• GPIO 기능사용

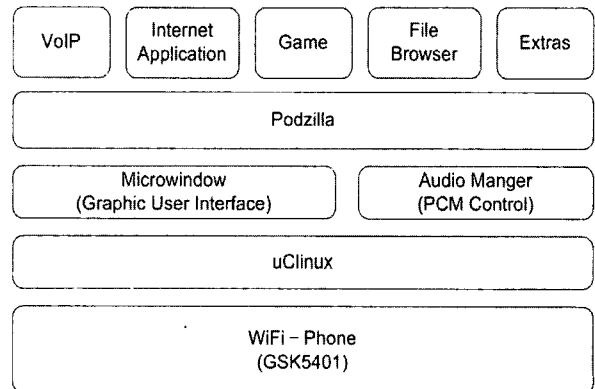
웨어 구조를 보여준다. 최하위 계층에 운영체제로 uClinux가 동작하고 있으며, 이를 통해 GSK5401 개발 보드의 하드웨어 자원을 상위 계층의 사용자 프로그램에게 제공한다. 경량 윈도우 관리기인 Microwindows[10]와 Audio Manager는 각각 운영체제가 제공하는 프레임 버퍼(Frame buffer)와 PCM 디바이스를 추상화하여 고수준의 API를 구성하고 이를 상위 계층의 사용자 인터페이스에게 제공한다. 사용자 인터페이스로는 Apple 사의 iPod[11]에서 리눅스를 사용하기 위한 iPodLinux 프로젝트[12]의 기본 사용자 인터페이스 프로그램인 Podzilla를 이용하였으며, 상위 계층 응용프로그램들은 Podzilla에 각각 모듈로 추가된다. 컴포넌트 계층화를 통해 상위 계층에서 새로운 컴포넌트의 추가가 용이하도록 하였다.

개발과정은 uClinux 상에서 LCD를 통한 사용자 인터페이스 구성과 PCM 디바이스 제어를 위해 Microwindows 윈도우 시스템을 포팅하고 Audio Manager 라이브러리를 작성하는 형태로 진행되었다. 이를 기반으로 Podzilla 인터페이스를 포팅하고 사용자 응용프로그램들을 작성하여 Podzilla 메뉴에 추가하였다. 각 요소별 세부 구현은 다음과 같다.

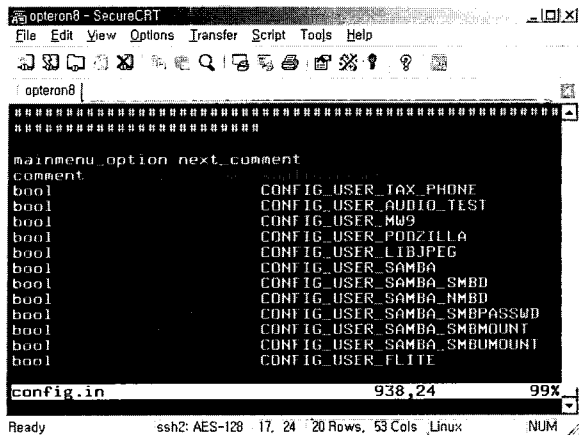
3.1.3.1 uClinux

임베디드 시스템의 구현에 있어 운영체제의 지원 없이 펌웨어 수준의 프로그램을 통해 전체 시스템을 구현하는 경우도 있다. 그러나, 본 응용의 경우 무선네트워크 기능을 이용하기 위해 TCP/IP 스택의 지원이 반드시 필요하고 고수준의 여러 응용프로그램들이 상호작용하여 돌아가기 때문에 운영체제의 사용이 필수적으로 요구된다.

GSK5401 개발보드에 사용된 GDM5104 프로세서는 MMU (Memory Management Unit)가 없으므로 MMU가 없는 프로세서에서 리눅스 사용을 위해 수정된 uClinux 커널 2.4.19 버전을 포팅하였다. uClinux는 물리 주소를 직접 사용하는 flat 메모리 기법을 사용하고 메모리 보호기능이 없으므로 응용프로그램 작성시 메모리 사용에 주의하여야 한다. 그러나, 일반적인 리눅스에서 제공하는 다양한 응용프로그램, 강력한 TCP/IP 네트워크 기능, 다양한 파일시스템 지원과 같은 장점은 동일하게 제공한다.



(그림 3) VoWLAN 단말 소프트웨어 구조



(그림 4) 개발보드의 config.in 파일

uClinux 배포판은 단순히 uClinux 커널만을 포함하는 것이 아니라 루트파일 시스템 구성에 필요한 응용프로그램 및 라이브러리까지 포함하고 있다. 따라서 새로운 프로그램을 포팅 또는 개발하려면 루트파일시스템의 user 디렉토리에 이를 추가하여야 한다.

본 연구에서도 루트파일시스템에 VoIP 프로그램을 포함한 다양한 응용프로그램을 추가하였으며 추가하는 과정은 다음과 같다. user 디렉토리에 응용 프로그램 소스를 추가하고 make 유틸리티에 의해 추가된 소스가 자동 컴파일될 수 있도록 Makefile을 작성하였다. 컴파일 과정에서 루트파일 시스템에 포함될 응용프로그램을 선택할 수 있도록 (그림 4)와 같이 config 디렉토리의 config.in 파일에 추가된 프로그램 목록을 추가하였다.

3.1.3.2 Microwindows

LCD 디바이스를 통한 그래픽 프로그램을 작성하는 경우 LCD 디바이스를 제어하는 라이브러리를 작성하여 픽셀 단위 폰트 출력, 선 그리기, 박스 그리기 등을 지원할 수 있으나 프로그램의 복잡성 및 코드 중복 등의 문제가 생긴다. 따라서 본 연구에서는 Microwindows 윈도우 시스템을 사용하여 고수준의 그래픽 출력 및 Microwindows 라이브러리를 통한 일관된 프로그래밍 인터페이스를 제공하였다. Microwindows는 임베디드 장치에서 고수준의 윈도우 환경을 제공하기 위해 개발된 공개 소스 윈도우시스템으로 폰트, 텍스트 출력, 비트맵, 이미지 출력, 커서 등의 기능을 지원한다.

Microwindows는 하드웨어 의존적인 드라이버 계층과 하드웨어 독립적인 그래픽 엔진, API 부분으로 계층화되어 있다. 상위 계층 사용자 인터페이스인 Podzilla와의 연동을 위해 Microwindows 0.90 버전을 uClinux 커널의 프레임버퍼 드라이버에 포팅하였다. 따라서 우선 uClinux의 루트파일시스템을 Microwindows를 포함한 바이너리 파일을 생성할 수 있도록 수정한 후 하드웨어에 의존적인 드라이버 층에 해당하는 GSK5401 개발 보드의 키보드 드라이버를 작성하여 포팅 작업을 마무리하였다.

3.1.3.3 Audio Manager

Podzilla를 기반으로 다양한 응용프로그램들이 추가되기 때문에 이러한 응용프로그램에서 PCM 디바이스를 이용하기 위한 일관된 프로그래밍 인터페이스의 지원이 요구된다. 따라서 PCM 디바이스에 대해 wave 형식의 데이터를 쓰고 읽을 수 있는 라이브러리를 작성하였다.

3.1.3.4 Podzilla

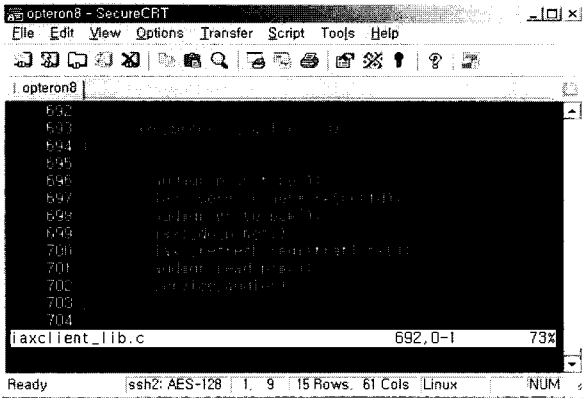
Podzilla는 iPod에 리눅스를 사용하기 위한 iPodLinux 프로젝트의 기본 사용자 인터페이스 프로그램으로 디렉토리 구조의 인터페이스를 통해 직관적이고 사용자 친화적인 사용 환경을 제공한다. 기본적으로 파일 브라우저, 이미지 뷰어, 게임 등 다양한 사용자 응용프로그램을 포함하고 필요에 따라 응용프로그램을 쉽게 추가할 수 있다.

GSK5401 개발보드와 iPod 모두 uClinux를 운영체제로 사용하고 Podzilla가 Microwindows 9.0버전을 기반으로 작성되었으므로 Podzilla를 본 연구에서 개발한 시스템에 포팅하는 데 큰 어려움은 없었다. 그러나, GSK5401 개발보드와 iPod는 LCD디바이스의 해상도와 키패드 디바이스가 다르다는 점을 고려하여 해당 부분을 수정하였다. LCD 해상도와 관련해서는 GSK5401 개발보드에서 사용된 LCD 해상도는 128×64 로 iPod LCD의 해상도에 비해 낮으므로 각각의 응용프로그램에 대해 128×64 해상도에 맞게 출력을 수행하도록 수정하였다. 또한, GSK5401 개발보드의 키패드 입력을 Podzilla에서 받아들일 수 있도록 이벤트 입력 부분을 수정하였다.

3.1.3.5 VoIP

VoIP 기능을 구현하기 위해서는 시그널링이 가능한 프로토콜이 필요하며 본 구현에서는 IAX 프로토콜[13]을 사용하였다. IAX는 Asterisk IP PBX 서버에서 지원하는 VoIP 프로토콜로 기존 VoIP 시스템에서 많이 사용되고 있는 SIP (Session Initiation Protocol)[14], H.323[15] 등을 대체하여 사용할 수 있으며, SIP에 비해 구조가 단순하기 때문에 구현이 용이하고 제어와 데이터 전송을 위해 UDP port 4569 번 하나만을 사용하기 때문에 NAT 환경에서도 잘 작동하는 장점이 있다.

VoIP 프로그램은 iaxcomm 소프트웨어[16]에서 사용된 IAX 라이브러리를 기반으로 작성하였다. IAX 라이브러리는 PC 환경을 고려하여 개발되었으므로 다음과 같이 개발보드에 맞게 수정하였다. 우선 호환성 제공을 위해 작성된 코드 부분을 제거하고 GSK5401 보드에 최적화하여 동작할 수 있도록 재구성하였다. 그리고, GDM5104 프로세서가 가진 4byte 단위 strict alignment 문제의 해결을 위해 IAX 라이브러리의 패킷 처리 코드 데이터 처리 단위를 모두 byte 단위로 수정하였다. 또한 네트워크 패킷을 처리하는 프로세스와 음성 데이터를 처리하는 쓰레드 간에 제한된 시간 안에 스케줄링이 이루어지지 않아 통화시 음성이 끊어지는 문제가 발생하여 쓰레드로 돌아가는 음성 데이터 처리 코드를 기존의 네트워크 패킷을 처리하는 부분과 하나의 루프 내에서 처리



(그림 5) VoIP 프로그램의 콜 프로세스 처리

하도록 수정하여 문제를 해결하였다.

VoIP와 같은 실시간 응용의 경우 제한된 시간 안에 특정한 작업이 이루어져야 하기 때문에 특정한 코드 부분이 제한된 시간 안에 수행되는지를 판단 할 수 있어야 한다. 이러한 경우 기본적으로 시스템 라이브러리를 사용하여 시스템 시간 값을 읽어올 수 있는데 대부분의 시스템에서 10msec 이하의 정밀한 시간 측정이 어렵고 최악의 경우 약 5msec의 오차가 발생할 수 있기 때문에 실제 개발에서 이러한 방법은 사용하기 어렵다. 본 구현에서는 GSK5401 개발 보드의 RTC (Real-Time Clock) 레지스터의 내용을 읽어 개별 코드 부분의 수행 시간을 정밀하게 측정할 수 있도록 하였다.

(그림 5)는 VoIP 프로그램의 핵심인 콜 프로세스를 처리하는 iaxc_process_calls 함수의 소스 코드이다. iaxc_process_calls 함수는 음성 패킷을 받아 이를 디코딩하여 PCM 디바이스에 쓰고, PCM 디바이스를 읽어 이를 인코딩하여 네트워크를 통해 전송하는 과정을 수행한다. 오디오 데이터의 인코딩, 디코딩을 위해 G.711, GSM 음성 코덱을 사용하였다.

3.1.36 응용 프로그램

VoWLAN 단말을 통해 다양한 부가기능을 제공하기 위해 본 연구에서 추가된 주요 응용 프로그램들은 다음과 같다.

- mini-httpd

VoWLAN 단말이 개인화되어 단말의 시스템 설정 및 개인 정보에 대한 관리가 요구됨에 따라 범용 웹 브라우저를 통한 시스템 및 개인정보 관리기능을 구현하였다. 이를 통해 사용자는 편리하게 개인 PC의 웹 브라우저를 통해 자신의 단말에 접속하여 개인정보를 수정할 수 있다. 이러한 기능은 임베디드 시스템을 위한 경량 웹 서버인 mini-httpd[17]와 C 언어를 사용한 CGI(Common Gateway Interface) 프로그래밍을 통해 구현하였다.

- MSN 메신저

VoWLAN 단말은 기본적으로 무선 AP를 통해 인터넷 망에 연결이 되므로 메신저와 같은 응용 프로그램은 핵심 기능이 될 수 있다. 메신저 구현은 개발 시간 단축 및 안

정성 확보를 위해 이미 많은 사용자들로부터 기능 및 안정성을 인정받은 공개 소스 MSN 메신저 프로그램을 사용하기로 하고, 이 중 C 언어로 작성되어 이식성이 뛰어나고 터미널 환경 기반으로 일반 GUI 기반 메신저보다 포팅이 용이한 Gtmess를 선택하였다.

Gtmess를 VoWLAN 단말 내에서 사용하기 위해 다음과 같이 크게 두 부분을 수정하였다. 첫째, Gtmess의 사용자 인터페이스는 ncurses 라이브러리[18]를 사용하지만 본 개발에서 사용된 플랫폼은 ncurses 라이브러리에 대한 지원이 없으므로 해당 부분을 기능적으로 대응되는 Microwindows 라이브러리 함수로 대체하였다. 둘째, Gtmess는 PC환경에서 사용되는 프로그램이므로 사용자 입력이 키보드를 통해 이루어지만, 개발보드에는 키패드만이 존재하므로 일반적인 모바일 단말에서 사용하는 키패드를 통한 입력방식을 지원하는 기능을 추가하였다.

- Samba

윈도우즈가 설치된 PC에서 VoWLAN 단말에 저장된 파일을 편리하게 관리할 수 있도록 Samba를 포팅하였다. 이를 통해 단말에 저장된 파일을 마치 PC의 로컬 하드디스크에 저장된 파일처럼 이용할 수 있는 휴대용 무선스토리지 기능을 구현하였다.

- 음성합성 엔진

시각장애인을 위해 VoWLAN 단말의 메뉴를 음성으로 재생해서 들려주는 Voice Menu 기능과 단말에 저장된 텍스트 파일을 음성으로 합성, 재생해 주는 기능을 구현하기 위해 음성합성기능을 구현하였다. 다양한 음성합성 엔진이 공개되어 있으나 임베디드 환경에서 사용하기에 적합한 것으로는 카네기 멜론 대학의 Flite 음성 합성 엔진[19]이 있다. Flite는 이미 다양한 플랫폼에 포팅되어 endianness, stricter alignment와 같은 문제들이 해결되어 있으므로 리눅스의 OSS (Open Sound System) 형식의 사운드 출력 부분만 단말의 Audio Manager에서 제공하는 라이브러리로 대체하였다.

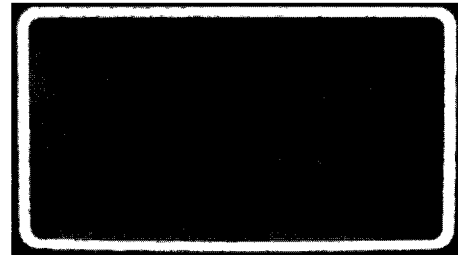
3.2 IP-PBX 서버

IP-PBX 서버로는 Asterisk IP-PBX 서버[20]를 사용하였다. Asterisk는 Digium 사에 의해 개발되고 있는 공개 소프트웨어 IP-PBX 시스템으로 별도의 하드웨어 장비 없이 conference Call, voice mail, interactive voice response와 같이 상용 IP-PBX 시스템에서 제공하는 다양한 음성 부가서비스를 제공한다. 또한, Asterisk는 IAX를 포함하여 SIP, MGCP[21], H.323 등의 주요 VoIP 프로토콜과, G.711, G.723.1, G.726, GSM, iLBC, Speex 등 다양한 음성 코덱을 지원하여 높은 호환성을 제공한다.

본 연구에서는 IP-PBX 서버를 구축하기 위해 레드햇 리눅스 9.0이 설치된 노트북에 Asterisk IP PBX 서버를 설치

<표 3> IP PBX 서버 구성

구분	내용
CPU	• Intel Centrino 1.6 Ghz
Memory	• 512 MB
HDD	• 40GB
OS	• RedHat 9.0 Linux
IP PBX Software	• Asterisk Ver 1.0.9



(그림 7) VoWLAN 단말의 초기화면

```

esca_note_canine - SecureCRT
File Edit View Options Transfer Script Tools Help
esca_note_canine
;Call Account
exten => 111,1,Dial(IAX2/Steve@Steve,10,Ttr)
exten => 111,2,VoiceMail(771@default)
exten => 111,3,Hangup()

exten => 112,1,Dial(IAX2/Bred@Bred,10,Ttr)
exten => 112,2,VoiceMail(772@default)
exten => 112,3,Hangup()

exten => 113,1,Dial(IAX2/Peter@Peter,10,Ttr)
exten => 113,2,VoiceMail(773@default)
exten => 113,3,Hangup()

exten => 114,1,Dial(IAX2/Ricky@Ricky,10,Ttr)
exten => 114,2,VoiceMail(774@default)
exten => 114,3,Hangup()

exten => 115,1,Dial(IAX2/James@James,10,Ttr)
                                     549,27 82%
Ready      ssh2: AES-128 18. 27 19 Rows, 48 Cols Linux  NUI
    
```

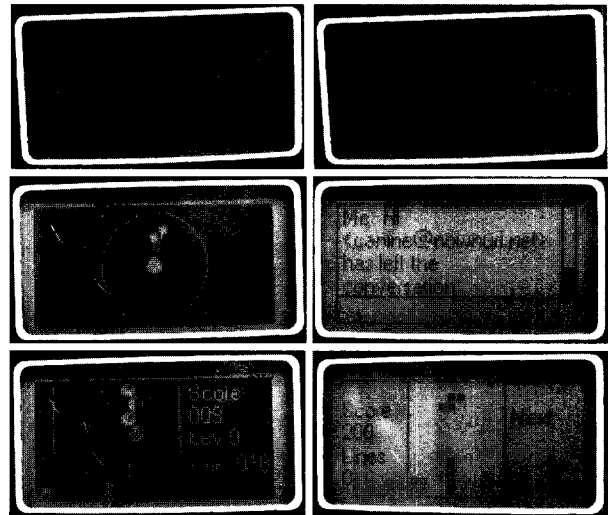
(그림 6) extensions.conf 파일

하고 사용자 계정 및 음성부가서비스 제공을 위한 각 기능에 해당하는 설정 파일을 작성하였다. Asterisk 서버는 이를 통해 VoIP, Music On Hold, Conference Call, Voice Mail, Interactive Voice Response, Call Parking과 같은 기능을 제공한다. <표 3>은 IP-PBX 서버의 세부 사양이며, (그림 6)은 사용자 계정을 등록하기 위하여 Asterisk 서버의 extensions.conf 파일을 수정한 화면이다.

3.3 시스템 구성화면

(그림 7)은 본 논문에서 제안한 VoWLAN 단말의 초기 실행화면이다. 각각의 메뉴는 단말의 키패드의 선택을 통해 지정한 프로그램을 실행하거나 서버메뉴로 들어갈 수 있도록 구현하였다.

(그림 8)은 (그림 7)의 각 메뉴를 선택하여 VoIP, 시계, MSN 메신저, 게임 기능을 실행한 모습이다. 첫 번째 행의 왼쪽 그림은 1234 번의 전화번호에 해당하는 사용자에게 전화를 거는 화면을 보여주고 있다. 첫 번째 행의 오른쪽 그림은 사용자 이름이 cbc이고 1237번의 번호에 해당하는 사용자로부터 전화가 걸려온 화면을 보여준다. 두 번째 행의 왼쪽 그림은 시계 프로그램을 실행한 화면이고 오른쪽 그림은 MSN 메신저 메뉴를 선택하여 MSN 메신저를 실행한 화면이다. 세 번째 행의 왼쪽과 오른쪽 그림은 게임 메뉴를 선택하고 서버메뉴로 들어가서 iPobble 및 Tetris 게임을 실행한 화면이다.



(그림 8) 무선단말의 어플리케이션 실행화면

4. 성능평가

본 논문에서 사용한 GDM5104 임베디드 프로세서는 TI의 OMAP [22] 등과 유사하게 고성능 신호처리와 일반적인 프로토콜 처리를 DSP와 RISC 프로세서를 이용하여 분할, 병렬로 처리할 수 있도록 되어 있다. VoIP의 구현은 크게 데이터의 송수신에 관련되는 프로토콜 처리 부분과 전송하는 음성 데이터의 효율적인 압축 및 압축 해제에 해당하는 코덱 부분으로 분할해 볼 수 있다. RISC와 DSP를 병렬로 처리하는 경우에 있어서 데이터 전송 프로토콜 처리는 RISC 프로세서에서 담당하고, 고성능의 신호 처리를 필요로 하는 코덱은 DSP에서 실행하는 것이 일반적이다. 이는 일반적으로 DSP프로세서가 RISC프로세서에 비해 단위 시간당 높은 전력을 소모하지만 RISC 프로세서에 비해 단위 전력당 높은 처리율을 보이기 때문이며[23], 본 논문에서 제시한 바와 같이 배터리를 이용하는 모바일 시스템의 경우 전력 사용량에 제한을 받기 때문에 동작 시간의 최대화를 위해서는 DSP 프로세서의 적절한 활용이 중요하다고 볼 수 있다.

<표 4>는 이와 같은 결과를 뒷받침하는 것으로써, 본 논문에서 적용한 G.723.1 인코더와 디코더를 RISC와 DSP 프로세서에 적용했을 때의 요구 성능 측정 결과이다. 초당 33 프레임이 발생하는 G.723.1 코덱을 처리하기 위하여 RISC 프로세서의 경우 DSP 프로세서부에 비하여 두 배 이상의

〈표 4〉 코덱의 RISC와 DSP 프로세서별 요구 성능

기능		RISC	DSP
G.723.1 인코더	MP-MLQ(6.4kbps)	67 MIPS	29.1 MIPS
	ACELP(5.3kbps)	45 MIPS	20.1 MIPS
G.723.1 디코더		6.4 MIPS	2.3 MIPS

〈표 5〉 프로토콜과 코덱의 구현에 따른 동작 시간

처리형태		동작시간 (분, 5회 반복 평균값)
프로토콜	코덱	
RISC	RISC	185
RISC	DSP	212

처리량이 필요함을 알 수 있다.

전력소모량은 전체 시스템 상에서 전류량의 동적 변화에 따라 가변적으로 나타나므로 본 연구에서는 전력 소모 측정을 위하여 시스템에 적용한 600mAh 용량의 NiMH 전지를 이용하여 RISC 프로세서에서 코덱과 프로토콜 처리를 이용한 경우의 동작 시간과 RISC 프로세서에서 프로토콜 처리만을 담당하고 DSP 프로세서에서 코덱의 처리를 담당한 경우의 동작 시간을 비교하였다. <표 5>는 무선랜을 이용한 VoIP 연결을 통하여 연속으로 음성을 양방향 전송했을 때 배터리 방전으로 동작이 불가능 할 때까지 측정된 시간 결과이다.

위와 같은 측정 결과에 따라 DSP 프로세서를 이용한 경우 RISC 프로세서만을 이용한 경우에 비하여 14.6%의 동작 시간 향상을 나타냈다고 할 수 있다. 이와 같은 결과는 RISC와 DSP의 전력효율성에 대비하여 적은 차이를 나타낸 것으로 판단되나, 이는 무선랜 등 전체 시스템에서 소모되는 전력에 대비하여 프로세서가 차지하는 전력 소모의 비율에 따른 것이라 할 수 있다.

5. 결 론

본 논문에서는 무선랜을 이용하여 기본 VoIP 서비스와 Voice Mail, Conference Call, Interactive Voice Response 등의 다양한 부가 서비스를 제공하는 VoWLAN 단말을 공개 소스 소프트웨어를 기반으로 임베디드 개발보드에 구현하였다. 본 논문에서 구현한 VoWLAN 단말은 공개 소프트웨어에 기반을 두어 추후 확장성을 보장하며, 내장형 시스템에 적합한 경량 소프트웨어 패키지들을 활용하여 제한된 시스템 자원으로 고수준의 기능을 구현할 수 있다. 본 연구에서 구현된 무선단말의 기능은 IP-PBX 소프트웨어를 이용한 IP-PBX서버를 구성하여 검증하였으며, 듀얼 코어 SoC 상에서의 코덱 처리 관련 성능평가 결과도 제시하였다.

본 논문의 결과를 바탕으로 진행되어야 할 차후 연구과제는 다음과 같다. 첫째, 무선단말과 센서 네트워크를 활용한 무선단말 기반의 홈 네트워크 기능의 확장에 대한 연구를 생각해 볼 수 있다. 둘째, 통화중 상대방의 단말로부터

스피커의 출력이 반사되어 되돌아오는 반향 문제 해결을 위해 반향제거(Echo canceler) 기능이 구현되어야 할 것이다. 마지막으로 무선단말은 무선 랜에 접속할 수 있는 공간에서만 통화가 가능하기 때문에 현재 통화 가능한 사용자의 목록을 보여주는 프리젠스(Presense) 기능이 구현되어야 할 것이다.

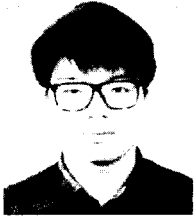
참 고 문 헌

- [1] VoIP Forum, <http://www.voip-forum.or.kr>.
- [2] IEEE 802.11 Standard, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, 1999.
- [3] H. Pan and S. Yamano, "An analysis of the channel utilization for call admission control in voice over WLAN," in Proceedings of the 2006 International Conference on Communications and Mobile Computing, pp.1229-1234, 2006.
- [4] H. Koga et. al., "A Quality-Aware VoWLAN Architecture and Its Quantitative Evaluations," IEEE Wireless Communications, pp.52-59, Feb., 2006.
- [5] S-L. Tsao and E-C. Cheng, "Reducing Idle Mode Power Consumption of Cellular/VoWLAN Dual Mode Mobiles," in the Proceedings of IEEE Globecom 2005, pp.2902-2906, 2005.
- [6] 권성수, 이종철, "통화 중 적응적 Power Save와 QoS 지원이 가능한 IEEE 802.11g VoWLAN 단말기 구현," 한국통신학회 논문지 제31권 제10A호, pp.1003-1013, 2006. 10.
- [7] uClinux™ - Embedded Linux Microcontroller Project, <http://www.uclinux.org>.
- [8] GCT Semiconductor, GDM5104 Developer's Manual, 2005, <http://www.gctsemi.com>.
- [9] uClibc: A C library for embedded Linux, <http://www.uclibc.org>.
- [10] Microwindows Project (currently, Nano-X Window System), <http://www.microwindow.org>.
- [11] <http://www.apple.com/ipod>.
- [12] iPod Linux, http://ipodlinux.org/Main_Page.
- [13] IAX (Inter-Asterisk eXchange) Protocol, <http://www.voip-info.org/wiki-IAX>.
- [14] J. Roseberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Scholler, "SIP: session initiation protocol," Request for Comments 3261 Internet Engineering Task Force, June, 2002.
- [15] ITU, ITU-T Recommendation H.323: Packet-based multimedia communications, Nov., 2000
- [16] <http://iaxclient.sourceforge.net/iaxcomm>.
- [17] http://www.acme.com/software/mini_httpd.
- [18] GNU ncurses library, <http://www.gnu.org/software/ncurses>.
- [19] Flite (Festival-lite) run-time synthesis engine, <http://www.speech.cs.cmu.edu/flite>.
- [20] Asterisk: An Open Source PBX and telephony toolkit, <http://www.asterisk.org>.
- [21] IETF, IETF RFC 3435, Media Gateway Control Protocol

(MGCP) Version 1.0, IETF, Jan., 2003.

[22] <http://www.omap.com>.

[23] M. Olivieri, M. Scarana, and S. Smorfa, "Circuit-level power efficiency investigation of advanced DSP architectures based on a specialized power modeling technique," IEEE Intl. Symp. on Circuits and Systems, Vol.5, pp.5266-5269, 2005.



서 호 중

e-mail : hjsuh@catholic.ac.kr

1991년 서울대학교 이학사

1994년 서울대학교 컴퓨터공학과
(공학석사)

2000년 서울대학교 컴퓨터공학과
(공학박사)

2002년 (주) 지씨티 리써치 책임연구원

2003년~현재 서울대학교 컴퓨터신기술공동연구소
직원연구원

2003년~현재 가톨릭대학교 컴퓨터정보공학부 조교수
관심분야: 컴퓨터구조, 컴퓨터시스템, 내장형시스템



이 병 호

e-mail : canine@catholic.ac.kr

2004년 가톨릭대학교 컴퓨터정보공학부
(학사)

2004년~현재 가톨릭대학교 컴퓨터공학과
석사과정

관심분야: 내장형시스템, 임베디드 프로세서,

운영체제



김 태 현

e-mail : thkim@uos.ac.kr

1994년 서울대학교 컴퓨터공학과 (학사)

1996년 서울대학교 컴퓨터공학과 (공학석사)

2001년 서울대학교 전기 컴퓨터공학부
(공학박사)

2001년~2005년 (주) 지씨티 리써치

책임연구원

2005년~현재 서울시립대학교 기계정보공학과 조교수

관심분야: Embedded Systems, Real-Time Systems, Wireless
Network