

# 개방형 STB기반의 U-Health 서비스 연구

## A Study on the Open STB based U-Health Services

강민구\* 김인기\*\*

### ◆ 목 차 ◆

1. U-Health 서비스 동향분석
2. 안드로이드기반 오픈 STB 플랫폼
3. 모바일/고정형 오픈 STB 플랫폼
4. 바이오 STB형 U-Health서비스
5. 결론

## 1.U-Health 서비스 동향분석

U-Health(유비쿼터스 헬스)는 보건의료에 유무선 네트워크 기술을 접목시켜 병원을 찾지 않더라도 언제, 어디서나 건강상태의 평가, 진단 및 치료, 건강관리를 받을 수 있도록 하는 의료서비스를 지칭한다. RFID 같은 U-Health 장비나 서비스를 일부 도입해 U-Health 병원으로 변신 중이다. 일부 지자체에서도 시범사업으로 유무선 네트워크를 이용해 지역 보건소와 대학병원을 연결하는 서비스를 실시하고 있다. 휴대폰 업체나 전자회사 등 기업에서도 U-Health 사업진출에 열을 올리고 있으며 병원 내 IT시스템 도입이 확대되는 등 병원의 원격진료 시스템이 점차 확산된다.

홈네트워크형 U-Health도 확산일로 환자의 혈압, 맥박, 혈당 등 생체신호를 병원 밖에서 측정한 후 운동과 식이, 투약 등 원격 서비스를 제공하는 이른바 ‘홈네트워크형’ U-Health 서비스도 급증하고 있다. 휴대폰을 이용한 만성질환자 관리 서비스는 이미 시행되거나 시행 중이다.

가톨릭의료원은 당뇨병 환자가 집에서 측정한 혈당을 인터넷 상의 웹-차트에 입력하면 담당 의사가 기록된 혈당이나 환자의 다양한 정보를 해석해 진단을 전송할 수 있도록 하는 ‘인터넷 혈당 관리 시스템’을 시범 운영했다.

서울대병원 강남센터는 포스코건설과 ‘u-헬스케어 서비스’ 협약을 맺고 송도시에 재택진료를 제공할 계획이다. 가정마다 체성분 분석기와 전자혈압계, 혈당측정계 등을 설치해 거주자가 집적 비만도나 혈압, 혈당 등을 측정하면 서울대병원 강남검진센터에 있는 첨단 의료기기가 이를 분석해 그에 필요한 건강식단이나 운동지침을 알려주거나 병원 진료 예약 등을 해주는 서비스다.



(사진 1) U-Health기반의 의료서비스 모델

\* 한신대학교 정보통신학과  
\*\* 이노디지털(주)

휴대전화 단말기에 건강관리 기능을 추가하는 것도 트렌드다. 삼성전자는 스포츠용품 업체 아디다스와 손 잡고 스포츠 특화 전화인 ‘마이코치(micoach)’를 출시했다. 휴대전화에 압센드, 심장박동 모니터, 보폭센서 등을 부착해 운동 시 자신의 심장박동수, 달리기 속도, 칼로리 소비량 등 각종 정보를 실시간으로 저장하고 보여준다. 뿐만 아니라, 저장된 정보를 마이코치 사이트(www.micoach.com)에 올리면 상세한 분석을 통해 신체 상태에 맞는 맞춤형 운동 프로그램 정보를 보여준다.

LG전자는 외장형 혈당 측정기기를 휴대폰에 연결해 채혈침으로 손가락에 피를 낸 뒤 일회용 스크립에 묻혀 측정기에 꽂으면 혈당수치가 단말기 화면에 곧바로 뜨는 ‘당뇨폰’에 이어 혈압측정 등이 가능한 휴대폰을 곧 선보일 예정이다.

LG전자는 캐나다 앨버타의대 등 해외 병원들과 함께 휴대폰으로 혈압, 체온 등과 같은 환자의 건강수치를 측정해 병원이나 주치의에게 전송하는 시스템 구축작업을 추진하고 있다.

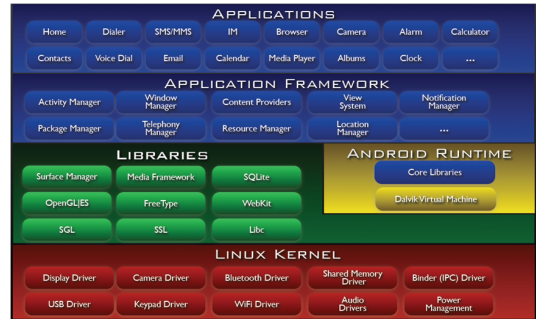
이러한 U-Health는 신성장동력 산업으로 정부 차원에서 많은 관심을 두고 있다. 보건복지가족부는 올해 u-의약품 종합관리 시스템, 원격 건강모니터링 시스템등을 추진할 예정이다. 또, 2009년부터 U-Health 산업의 연구개발에 예산을 투입할 계획이다.

## 2. 안드로이드기반 오픈 STB 플랫폼

### 2.1 안드로이드 플랫폼의 구조

안드로이드는 Google OHA(Open Handset Alliance)에서 발표한 모바일 오픈 솔루션이다. 안드로이드는 기존의 모바일 솔루션의 공개 범위와 매우 다르게 최하단의 OS부터 최상단의 킬러 어플리케이션까지 모든 범위를 소스레벨로 공개했다. 더욱이 공개된 솔루션의 안정성이 하드웨어만 있으면 누구나 핸드폰을 만들 수 있을 정도로 높고 다양한 어플리케이션이 포함되어 있으며 그것의 완성도 또한 높아 이미 대만의 HTC에서는 안드로이드 솔루션을 이용하여 핸드폰을 제조 판매하고 있다.

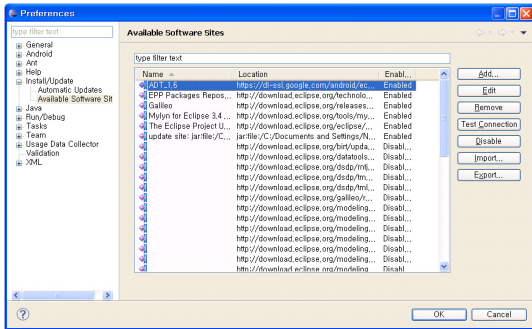
다른 핸드폰 제조업체인 삼성, LG, 모토로라 또한 안드로이드를 이용한 핸드폰개발에 대한 계획을 이미 세우고 있다.



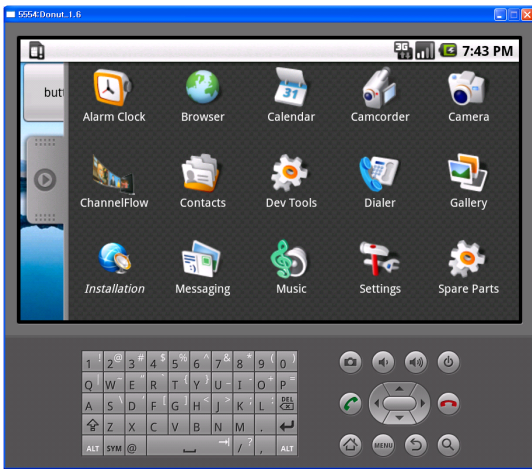
(그림 1) 안드로이드 플랫폼 구조도

그림 1은 안드로이드 플랫폼의 구조이다. 그림에서 볼 수 있듯이 OS는 Linux Kernel을 사용하였고 Library는 많은 오픈 솔루션을 이용하였다. Android Runtime은 구글이 자체 개발한 Dalvik Virtual Machine과 Core Library로 구성되어 있고 이것 또한 소스 레벨로 공개된 상태이다. Application Framework은 다양한 기능의 프레임워크가 이미 완성되었고, Application은 전화 송,수신, SMS 송,수신, 인터넷, 이메일, 구글맵스 등의 킬러 어플리케이션이 안드로이드 솔루션에 이미 포함되어 있다.

본 논문에서도 이 솔루션을 모바일 핸드폰이 아닌 고정형 셋톱박스에 올려 플랫폼을 통일하고 서비스와 콘텐츠를 통일하여 비용절감 및 효율성을 꾀하는데 있다. 또 하나 안드로이드 특이할 사항은 바로 개발 환경이다. 안드로이드는 자바의 개발 IDE로 유명한 Eclipse에 그림 2와 같은 Plug-in을 제공하여 개발을 용이하게 할 수 있고, 개발 응용 어플리케이션을 PC에서 쉽게 확인해 볼 수 있는 그림 3과 같은 Goldfish ARM Emulator환경을 핸드폰의 형상과 동일한 형태로 제공하고 있어 어플리케이션 개발 및 검증을 쉽고 빠르게 할 수 있다는 큰 장점을 가지고 있다. 더불어 안드로이드 솔루션은 ADB(Android Debugger Bridge)라는 디버깅 환경을 제공하여 개발자에게 개발 중 문제해결을 쉽게 할 수 있다.



(그림 2) Eclipse 안드로이드 Plug-in



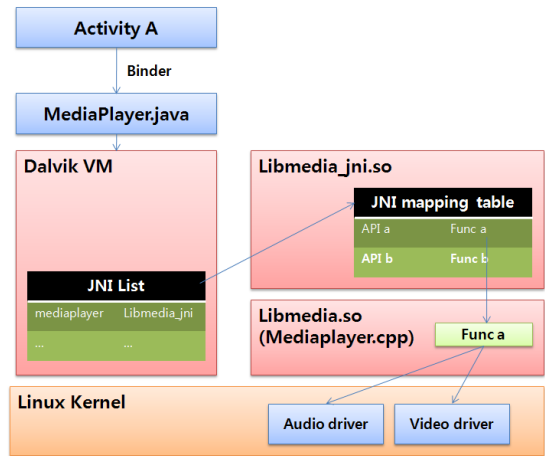
(그림 3) Goldfish 안드로이드 에뮬레이터

표 1은 안드로이드 플랫폼의 주요 특징을 보여주고 있다. 안드로이드는 모든 소스 및 컴포넌트가 오픈 소스 기반이다. 즉 제조사에서 제품을 만들 때 가장 부담이 되는 라이선스 정책을 용이하게 할 수 있다.

(표 1) 안드로이드 플랫폼의 주요 특징

Feature	Description
응용 프레임워크	컴포넌트를 교체하고 재사용할 수 있음
Dalvik VM	현재는 모바일 디바이스에 최적화
통합 브라우저	Webkit 기반 브라우저
최적화된 Graphics	2D 라이브러리와 OpenGL ES 1.0 라이브러리 제공
SQLite	데이터 베이스
Media 지원	이미지, 오디오, 비디오(MPEG4, H.264,

Feature	Description
	MP3, AAC, AMR, JPG, PNG, GIF)
GSM Telephony	하드웨어에 맞게 포팅 가능
Bluetooth, EDGE, 3G, WiFi	하드웨어에 맞게 포팅 가능
카메라, GPS, 나침반, 가속기	하드웨어에 맞게 포팅 가능
풍부한 개발환경	에뮬레이터, 디버거, 프로파일링, Eclipse 플러그인



(그림 4) 미디어 플레이어 실행과 JNI

안드로이드 플랫폼의 장치 드라이버는 일반 리눅스 커널의 장치 드라이버 형태를 그대로 사용하기 때문에 특별한 설계가 필요하지 않다. 단 장치 드라이버를 접근하는 경로가 모두 JNI 함수로 구현되어야 하기 때문에 이를 고려하여 장치 드라이버를 작성할 필요가 있다. 그리고 일부 장치 드라이버는 일반 리눅스에서는 필수적인 사항이 아니지만 안드로이드 플랫폼에서는 필수적으로 지켜져야 하는 사양이 정해져 있는 경우도 있다.

### 3. 모바일/고정형 오픈 STB 플랫폼

#### 3.1. 프레임 버퍼의 성능 개선방안

안드로이드는 프레임버퍼 두 개를 할당하여 서로 번갈아가면서 화면을 구성하는 더블 버퍼링 구조가

포함되어 있다. 모바일 안드로이드 프레임 버퍼를 셋톱박스의 프레임 버퍼로 기능을 개선위해 셋톱박스의 화면 크기는 모바일 제품의 화면 크기보다 훨씬 크다. 최근 방송시장에서는 HD가 기술 트렌드가 되고 있어 그 크기는 더욱더 커지게 되었다.

이에 안드로이드 프레임 버퍼 장치 드라이버를 셋톱박스의 일반적인 화면 사이즈인 720 x 576(576i), 1280 x 720(720p), 1920 x 1080(1080i) 중 하나를 지원할 수 있도록 수정해야 한다. 요즘 디스플레이의 상당 부분이 LCD로 바뀌면서 고해상도인 1080i를 지원하는 것이 일반화 되었기에 1080로 프레임 버퍼를 수정한다.

1080i의 해상도를 지원하게 되면 안드로이드의 킬러 어플리케이션 중 하나인 웹 브라우저를 LCD의 큰 화면에서 편리하게 사용할 수 있다. 그리고 화면 resolution은 24bit RGB모드로 수정한다. 24bit RGB모드를 사용하면 자연색 그대로를 표현 할 수 있기에 좀더 자연스러운 HD급 메뉴가 가능해 진다. 26bit RGB 모드의 1080i를 지원하기 위해서는 리눅스 커널의 drivers/video/에 있는 프레임 버퍼 driver를 셋톱박스의 프레임 버퍼에 맞게 수정한다. 이때 커널 파라미터의 CONFIG\_FB=y, CONFIG\_FB\_BPP\_24=y를 설정하여 driver에서 사용하도록 편리하게 bpp를 바꿀 수 있다. 그림 5는 커널 로그가 부팅 시 출력되면 더블 버퍼링 프레임 버퍼가 정상적으로 설정된다.

```
Window[0] - FB1 : map_video_memory: clear ff600000:00177000
FB1 : map_video_memory: dma=57c00000 cpu=ff600000 size=00177000
Window[0] - FB2 : map_video_memory: clear ff6bb800:000bb800
FB2 : map_video_memory: dma=57cbb800 cpu=ff6bb800 size=00177000
```

(그림 5) 안드로이드 프레임 버퍼 관련 커널 로그

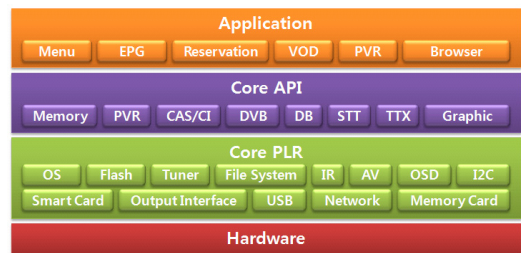
### 3.2 고정형 플랫폼 미들웨어 개발

방송 수신기인 셋톱박스는 많은 멀티미디어 기능과 다양한 출력, 그리고 칩셋에 대한 솔루션도 많고, 기능에 따라 하드웨어의 종류도 적지 않다. 특히 나라마다 지역마다 스펙 및 기능을 조금씩 바꾸어 차별화를 하고 있고 같은 나라, 같은 지역이라 하더라도 바이어마다 조금씩 커스텀마이징을 해야 되는 상황이다. 이에 가장 빠르게 시스템을 설계하고 가장 빠르게 바이

어를 지원하는 타임 투 마켓이 매우 중요한 기술 분야이다.

이에 타임 투 마켓의 경쟁력을 확보하기 위해 제조사는 자체적으로 이를 가장 효율적으로 하기 위한 미들웨어 솔루션을 확보하거나 또는 미들웨어 전문 업체의 솔루션을 가져다가 포팅 하는 방법을 사용하고 있다.

셋톱박스에 미들웨어는 필수적이다. 미들웨어 설계 시 크게 고려해야 할 부분은 다양한 하드웨어를 지원하는 포팅 레이어를 어떻게 설계할 것인지, 그리고 포팅 레이어를 이용한 코어 API를 얼마나 효율적으로 설계하는지에 따라 시스템의 성능 및 안정성이 결정된다. 코어 API설계 시 고려해야 할 부분은 셋톱박스에서 지원해야 할 어플리케이션이 어떤 것인지 먼저 그것에 대한 사양을 정하여 그것에 맞는 코어 API는 개발이 되어져야 한다. 그림 6과 같이 셋톱박스 소프트웨어 구조도와 같이 미들웨어가 레이어 구조로 설계 되어야 한다.



(그림 6) 고정형 셋톱박스 소프트웨어 구조도

Core PLR은 하드웨어별로 포팅을 용이하기 위한 API셋으로 구성되어 있다. 하드웨어에 컴포넌트가 새롭게 추가되면 추가된 하드웨어에 맞는 포팅 레이어 API를 제작하면 된다. 그림 6은 셋톱박스에 맞추어져 있는 포팅 레이어이고 하드웨어가 셋톱박스가 아닌 것이라면 해당 하드웨어에 맞는 컴포넌트들로 구성하면 된다. 포팅 레이어를 얼마나 잘 만드느냐에 따라 하드웨어 의존성 여부가 결정된다. 어플리케이션 및 코어 API이상에서는 하드웨어와 연관된 어떠한 API도 호출하지 않는 구조가 최상의 구조로 하겠다. 포팅 레이어의 안정성 정도에 따라 전체 시스템의 안정성이 결정된다. 어플리케이션의 종류에 따라 Core API의 기



능 또한 막강해 지고 API수 또한 많이 필요하게 된다. 포팅 레이어 API와 다르게 Core API는 미들웨어 개발 시 단 한 번만 개발하면 되는 모듈이기에 매우 치밀하게 설계를 할 필요가 있고 향후의 호환성까지도 고려가 되어야 한다.

그렇지 못하면 향후 Core API수정 또는 보완뿐만 아니라 어플리케이션의 수정도 불가피 하기 때문이다. 어플리케이션까지 수정을 하게 되면 작업량은 상상을 초월할 정도로 많아지고 시스템의 안정성은 사라지고 제품의 QA, QC시간이 많이 소요되기 때문이다.

### 3.3 자바에서 방송API를 호출위한 JNI개발

JNI는 Java Native Interface이다. 즉 Java Code에서 시스템에 Dependant한 Native API를 호출하여 사용하기 위한 인터페이스이다. 많은 임베디드 시스템의 Low-Level Code들은 아직까지 Java로 구성되어 있지 않다. 대부분 C 또는 C++로 작성되어 있고 이것을 모두 Java로 바꾸기란 거의 불가능한 수준이다. 하지만 어플리케이션은 자바로 되어 있고 그 어플리케이션에서 Native API를 사용하고자 한다면 그때는 JNI를 사용하여 구현하면 된다. 여기에서 주의할 점은 JNI를 사용하는 순간 Java의 플랫폼 독립성이 사라진다는 것이다. 즉 JNI API가 포함된 어플리케이션들은 다른 Java 플랫폼에서 동작하지 않는다는 것이다. 이에 JNI를 설계할 때 어플리케이션에서 사용하는 API만을 잘 정리하여 인터페이스화 시킬 필요가 있다. 불필요하게 많은 Native API를 JNI 인터페이스로 만들 필요가 없다. 플랫폼의 독립성은 높으면 높을수록 유리하기 때문이다.

## 4. 바이오 STB형 U-Health서비스

손목에 찬 상태로 다양한 건강정보를 측정 할 수 있는 소형 헬스케어 장치가 개발되어 고령화 시대를 맞아 증가하고 있는 만성질환을 움직이면서 손쉽게 관리 할수 있는 길이 열린 것이다. 현재 선진국에서도 소목형 다측정기능을 가진 헬스케어장치로 혈압측정기와 혈당측정기만 손목시계형으로 제품화된 상황이다.

한국전기연구원은 혈압, 혈당, 심전도, 맥박수, 피부온도, 체지방,움직임(활동) 등 7가지의 건강정보를 생활하면서 측정,실시간 컴퓨터로 무선 전송해 의료진에게 전달할수 있는 손목 시계형 헬스케어 시스템을 개발했다.



(사진 2) 손목시계형 헬스케어장치

특히, 혈압측정기능은 절대 압력과 맥박의 펄스와 형을 도식화해 표시함으로써 눈으로 상태를 직접 확인, 혈압측정이 가능하다. 측정시간의 단축과 더불어 혈압측정시 맥박수를 동시에 측정해 표시해준다. 심전도 측정은 건성 전극을 사용, 간편하게 심전도 파형을 측정하면서 기록 및 전송이 가능하도록 해부정맥 현상 등의 건강이상을 체크할 수 있다. 이 시스템의 크기는 110×75×30mm로 측정된 건강관리 데이터를 저장할 수 있는 512Mb의 플래시 메모리와 전원공급을 위한 Li-ion 배터리를 포함한 무게는 210g으로 충전은 휴대전화 충전기를 사용하여 충전할 수 있다.

또한, U헬스케어 서비스가 휴대전화를 매개체로 다양하게 출시되어 건강관리부터 여가생활, 스포츠 기록 관리 및 금연관리, 임신관리 등을 할 수 있는 서비스도 등장하였다. IT융합서비스로 총 7종의 헬스케어 애플리케이션을 서비스가 가능한 대표적인 애플리케이션은 △테마별 건강안내 △금연관리 △물 섭취량관리 △바이오리듬/운세 △임신관리 및 월경주기 관리 등이다.

유럽에서 오픈한 삼성 앱스토어에는 △사운드박스 △바이오리듬△골프 △임신관리 등이 있다. 이러한 손목형 헬스케어 시스템과 이동형 헬스케어 시스템 등

이 개방형 STB 기반의 바이오 셋탑박스와 결합한다면 고령사회에 대비한 헬스케어 시스템의 개발에 박차를 가하고, 정부가 추진하고 있는 신성장동력 산업의 하나인 헬스케어 분야에 활용될 것이다.

## 5. 결론

본 논문에서는 개방형 STB기반의 U-Health 및 BM 개발 연구와 함께 U-Health의 핵심 기술인 센싱의 정확도, 의료정보에 대한 표준화 작업, 원격의료의 허용 범위나 건강보험 같은 의료법 제도가 정비되고 네트워크를 통해 병원 밖에서 생체정보를 주고 받기 때문에 정보보호가 해결되고, 인터넷 등 병원 밖에서 이용하는 각종 의료향정보를 홈 게이트웨이 역할을 하는 STB가 U-Health를 위한 원격 모니터링이 가능할 경우 노인환자나 만성 질환자들에게 많은 도움을 주는 것은 물론 질병예방 차원에서도 효과가 클 것이다.

## 참고 문헌

- [1] 구글 안드로이드 플랫폼, 아키텍처, <http://code.google.com/android>
- [2] 구글 애플리케이션 아키텍처: Android & OSGi, <http://www.osgi.org/blog/2007/11/android-and-osgi.html>
- [3] 마이크로소프트, “임베디드 플랫폼 구글 안드로이드 전략,” 2008.05
- [4] 남상엽, 강민구, 김인기, “구글 안드로이드 분석과 실습,” 상학당, 2009.09

## ● 저 자 소 개 ●



### 강 민 구

1986 연세대학교 전자공학과(공학사)  
 1989 연세대학교 전자공학과(공학석사)  
 1994 연세대학교 전자공학과(공학박사)  
 1985~1987 삼성전자 연구원  
 1997~1998 일본 오사카대학 Post Doc.  
 1994~2000 호남대학교 정보통신공학부교수  
 2000~현재 한신대학교 정보통신학과 교수



### 김 인 기

2005 군산대학교 제어계측공학과(공학사)  
 2009 한신대학교 정보통신학과(석사수료)  
 2005-2001 삼성전자 연구원  
 2001-2008 가온미디어(주) 연구소장  
 2008~현재 이노디지털(주) 대표이사