Journal of Institute of Control, Robotics and Systems (2012) 18(11):1018-1026
http://dx.doi.org/10.5302/J.ICROS.2012.18.11.1018 ISSN:1976-5622 eISSN:2233-4335

안드로이드 플랫폼을 탑재한 스마트 지문인식장치 개발

Development of Smart Fingerprint Recognition System with Android Platform

이 갑 래*
(Kap Rai Lee[1])
[1]Pyeongtaek University

Abstract: This paper presents a developing method of smart fingerprint recognition system. First, we design a hardware configuration circuit using a 32bit Risc CPU, a fingerprint sensor, a LCD, and a WiFi communication chip to realize the smart fingerprint recognition systems. It is necessary to develop a JNI (Java Native Interface) library and a device drive program of fingerprint sense to develop application program of fingerprint recognition system with Android platform. Thus second, we develop a device drive and a JNI program. And we also develop an application program of fingerprint recognition systems using developed JNI library. Finally test results are presented to illustrate the performance of the developed smart fingerprint recognition system.

Keywords: smart fingerprint recognition system, android platform, WiFi, JNI library, device drive, application program

I. 서론

지문인식시스템(FRS: Fingerprint Recognition System)은 전통적으로 출입통제시스템, 근태관리, 범죄수사용 등에 많이 사용되어 왔으며, 최근에는 컴퓨터 데이터 관리, 전자상거래용 인증시스템 등에 응용되기 시작하여 모바일 전용 기기와 결합되어 확장되는 추세이다. 지문인식 시스템 개발 연구 논문으로는 지문입력기 사이의 호환이 가능한 시스템 개발 연구[1]와 무선 환경(IEEE 802.11 및 CDMA 기반)에서 지문인식 신원 조회 시스템 개발 연구 등이 있다[2]. 우리나라는 미국, 프랑스, 일본과 함께 높은 성능의 기술을 보유하고 있으며, 세계적으로 가장 활발하게 개발하는 품목 중의 하나이다. 지문인식 기술은 타 생체보안 시스템보다 사용하기 편리하고 신뢰성이 이미 검증되어있어 바이오인식 전문 업체들은 적극적으로 응용제품을 만들어 해외 또는 국내 시장을 공략하고 있다. Gemplus사는 스마트카드 솔루션을 바탕으로 카드내에서 인식 처리를 수행하는 match-on-card 방식의 기술을 개발하였으며, Obethur Card System사는 스마트카드와 리더로 구성된 지문인증 제품을 개발하였다. 카드리더에 있는 지문입력센서를 통하여 지문을 입력받아 특징을 추출한 후 스마트 카드에 지문 특징정보를 저장하고 스마트카드에서 매칭을 수행하여 인증결과를 출력하는 sensor-on-card 방식이다[2]. 또한 소니사는 지문 정보 저장과 지문 인증 연산을 시스템 내에서 수행하고 USB 방식으로 호스트와 통신하는 지문인증 시스템을 개발하였다[2]. 대체적으로 단순한 지문인식 시스템에서 벗어나, 잡한 인터넷 환경에 접근하는 복합 솔루션을 타진하고 있는 상태이다. 외형에서도 기존의 산업기기 형태에서 벗어나 스타일시한 디자인 또는 수려한 그래픽 정보를 사용자에게 전달할 수 있도록 진화하고 있는 중이다. 기존 출입 보안시스템에서 음성 출력, 화면 디자인은 많은 시간을 요구하는 작업이므로, GUI 디자인에서 우위를 점할 수 있고 빠른 처리속도를 가질 수 있도록 모바일 전용 OS를 채택한 스마트 지문 인식기가 최근 연구 개발되고 있다[3,4].

본 연구에서는 무선 네트워크 환경에서 동작가능하고 안드로이드 플랫폼을 탑재하여 멀티미디어 플레이가 가능한 스마트 지문 인식장치를 개발한다. 32bit RISC 중앙처리장치를 사용하며 지문인식 센서는 가장 물리적으로 안정하다고 알려져 있는 광학식 단말기를 사용한다. 지문 인식은 500 mSec 이내로 구현하며 이 인증자료들은 단말기 플래시 메모리에 저장된다. WiFi 무선 LAN을 장착하여 중앙 서버 컴퓨터와의 연동이 가능하게 한다. 안드로이드 상에서 동작하는 지문 디바이스드라이버를 개발하고, 또한 안드로이드 상에서 그 지문 드라이버를 호출하는 인터페이스 프로그램을 개발한다. 개발한 이 인터페이스 프로그램을 이용하여 지문 인증알고리즘과 연동하는 응용프로그램을 개발한다. 외부 키를 거의 없애고 LCD 터치패드를 사용해 기존의 산업기기 형태에서 벗어나 스타일시한 외형 디자인이 가능하며, WiFi 기능을 탑재하여 무선 네트워크를 사용함으로서 설치되는 곳의 디자인을 세련되게 한다. 또한 채택한 터치 LCD에는 사용자들에게 많은 그래픽 정보를 전달할 수 있다. 중앙 서버컴퓨터와의 연동이 가능하므로 근태 관리 시스템으로 사용가능하며, 단말기는 주로 사무실 정면에 설치되므로 출입이 없는 평상시에는 사내 광고홍보용으로도 사용 가능하다. 안드로이드는 현재 차세대 버전으로 계속 진

* 책임저자(Corresponding Author)
논문접수: 2012. 8. 27., 수정: 2012. 9. 14., 채택확정: 2012. 9. 25.
이갑래: 평택대학교 정보통신학과(krlee@ptu.ac.kr)

화하고 있으며, 음성인식 기능, 메시지 음성 출력 기능들 첨단 기능을 포함시키고 있어 본 제품에 타당한 제품을 채택함으로서 첨단 제품으로 변신하는 것이 가능하다. 안드로이드는 거의 모든 언어를 지원하고 있으며 안드로이드 어플리케이션 작성 시 간단한 폴더 구성으로 거의 모든 국가의 제품을 구성할 수 있다.

II. 지문인식장치 개발

본 연구에서 개발하는 지문인식장치의 지문저장 건수는 10,000건 이상이며 지문처리 속도는 500 msec 이내이다. 또한 와이파이(Wi-Fi, 무선 인터넷) 통신이 가능하며, 터치 패드 기능을 내장하여 화려한 외관 설계가 가능하게 한다. 안드로이드에서 제공하는 화면 처리 기능(surface manager), 미디어 프레임(media framework), 데이터베이스 엔진(SQLite) 등을 이용하여 앱(application)을 개발함으로서 간편성 및 신뢰성을 확보한다. 또한 영어, 한국어 등의 다중 언어 지원과 멀티미디어 기능을 가능하게 하여 광고용으로도 사용가능하게 개발한다.

또한 지문인증 어플리케이션인 "쥬피터"는 안드로이드 Application Framework 및 Applications을 이용하여 개발하며 지문 인증 알고리즘 등 속도가 요구되는 프로그램은 NDK를 이용해 프로그램을 개발한다. 지문 인증 애플리케이션(지문인증, 지문등록, 사용자리스트 등) 개발 시 안드로이드에서 제공하는 라이브러리 및 애플리케이션 프레임워크만을 이용하여 개발할 수는 없다. 왜냐하면 지문센서로부터 지문 정보를 가져올 시에 지문 디바이스 드라이버 프로그램을 구동 시켜야 하며, 이 디바익스 드라이버를 제어하는 JNI 라이브러리가 없기 때문이다. 따라서 지문 디바이스 드라이버 프로그램과 이를 제어하는 JNI 프로그램 을 또한 개발한다.

1. 하드웨어 구성 및 설계

하드웨어 구성 및 입출력 장치와의 연결 설계는 그림 1과 같다.

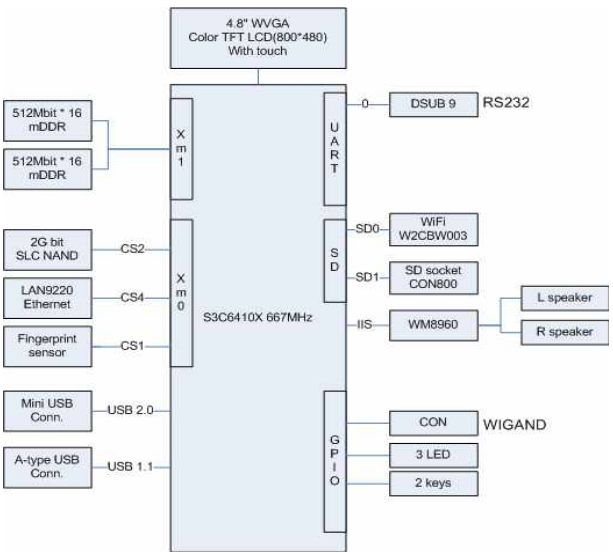

그림 1. 스마트 지문인식장치의 하드웨어 구성회로.

Fig. 1. Hardware configuration circuit of smart FRS.

• 중앙처리장치

중앙처리장치는 현재 모바일 기기에 많이 사용되고 있으며, 모바일 플랫폼 포팅(porting)이 활발하게 진행되고 있는 삼성 S3C6410를 선택한다. S3C6410는 64/32 비트 내부 버스 구조를 가지고 있어 3G 통신 서비스에 최적화된 그래픽 성능을 제공한다. 멀티미디어 지원으로는 비디오, 오디오 처리 기능과 Open GL 3D/2D이 있으며, MSC(Multi Standard Codec)기반의 MPEG4, H.264, H.263, VC1 인코딩과 디코딩 기능을 지원한다[5].

• 플래시 메모리(Flash Memory)

메모리 포트 Xm0의 CS2에는 256메가바이트의 SLC (Signgle Level Cell) 낸드플래시(NAND flash)메모리가 연결되어 있어 운영체제 및 기타 필요한 데이터를 저장할 수 있다. 이 메모리에는 부터로더, 리눅스 커널, 안드로이드 앱, 안드로이드 스택, 안드로이드 앱의 데이터 등이 저장 된다.

• SDRAM 메모리

외부메모리 포트 Xm1의 CS0에 32비트 인터페이스를 사용하는 133 MHz의 모바일 DDR SDRAM을 연결한다. 안드로이드를 안정적으로 구동시킬 수 있는 최소 용량의 RAM 사양으로 128메가바이트로 설계한다. SDRAM 메모리는 앱 및 운영체제가 실행될 때 작업공간으로 사용된다.

• 디스플레이

5인치 WVGA(800x480) 컬러 TFT (Thin Film Transistor) LCD를 선택한다. 터치스크린의 좌표를 입력하기 위한 YM/YP/XM/XP는 중앙처리장치의 ADC단자의 AIN4~7에 연결되어 있다.

• 오디오

Wolfson의 WM8960 오디오코덱은 코덱 기능뿐만 아니라 1W 상당의 D-클래스 스테레오 파워 앰프(stereo power amplifier) 까지 내장되어 있어 외장 스피커로도 좋은 음량의 소리를 들을 수 있다. WM8960 코덱칩은 IIS 인터페이스로 중앙처리장치와 연결되어 음악 또는 소리를 출력한다.

• 이더넷 및 와이파이

이더넷 성능 향상 및 보드 공간 절약을 위해 SMSC사의 LAN9220을 사용한다. 와이파이 구현을 위해 와이파이/블루트스 콤보 모듈을 선택한다. 와이파이칩은 중앙처리장치와 SD 인터페이스로 연결되며 LAN9220는 Xm0의 CS4에 16비트로 인터페이스되어있다.

• SD 메모리

표준 SD 메모리 인터페이스 슬롯을 장착한다. 이 슬롯을 통해 동영상, 음악을 저장 또는 추출할 수 있다. 안드로이드에서 자동적으로 장착/탈착을 인식하는 기능을 제공한다. 또 음악, 동영상을 자동적으로 검색해 리스트 해준다.

• 버튼키

키1은 안드로이드의 필수 키인 "컨텍스트 메뉴(context

menu)"의 기능으로, 키2는 "이전(back)"키로 할당하였다. 이 두 키만으로 훌륭한 안드로이드 환경을 동작시킬 수 있다.

• 위겐드(WIEGAND) 통신 회로

외부 보안 기기들과 통신하기 위해 산업계 표준인 위겐드 통신 회로를 구현한다. Open collector로 중앙처리 장치와 연결되며 충분한 전류를 공급하기 위해 고전류 달링톤 트랜지스터 IC(2003)를 사용한다.

• 지문센서

지문데이터는 Xm0의 CS1에 8비트 인터페이스로 읽어오며, 영상의 동기신호는 GPIO로 연결한다. 지문센서는 CMOS 이미지 센서를 사용하며 이를 최적화하기 위한 칩 설정은 표준 I2C 시리얼 bus를 통해 명령을 전달한다.

2. 개발환경 및 안드로이드 플랫폼 탑재

지문인식장치에 탑재될 소프트웨어는 부트로더(boot loader), 리눅스 커널(linux kernel), 루트 파일 시스템(root file system)이며, 루트 파일 시스템에 안드로이드 자체 제공 앱(어플리케이션)과 및 본 개발에서 작성한 앱인 "쥬피터"가 탑재된다. 개발 호스트 컴퓨터에는 커널 수정 및 디바이스 드라이버 개발을 위해 리눅스 환경이 필요하다. 안드로이드 개발환경에 가장 적합한 우분투(9.10)을 PC에 설치하고, 그 위에 크로스 컴파일러(ARM eabi 4.2.2)를 설치한다. 안드로이드 루트 파일 시스템 또한 이 우분투에서 컴파일되며 컴파일에 필요한 JDK (Java Development Kit)를 비롯한 여러 가지 라이브러리들을 함께 설치한다. 또한 쥬피터 앱 개발을 위해 윈도우즈 환경에서 안드로이드 SDK (Software Development Kit), 안드로이드 NDK (Native Development Kit)를 설치한다. 안드로이드 SDK는 안드로이드 플랫폼 위에 자바언어를 사용하여 앱을 개발할 수 있는 툴과 API를 제공한다[6,7]. 그와 다르게 NDK는 앱 개발자에게 필요한 물건이 아니라, 하드웨어 개발자가 java에서 하드웨어를 컨트롤하기 위한 도구이다. 본 연구에서도 지문인식 알고리즘이 안드로이드 SDK에 포함되어 있지 않으므로 지문알고리즘을 실행하고, 지문센서에서 영상을 읽어오는 등 하드웨어를 제어하는데 NDK를 사용한다.

부트로더는 낸드플래시에 존재하며 전원이 인가될 때 가정 먼저 실행된다. 커널, 루트 파일 시스템 저장 역할을 하며 사용자의 입력이 없을때 커널로 제어권을 자동으로 넘긴다. TFTP (Trivial File Transfer Protocol), 이더넷, USB otg(on the go), 시리얼 모니터 기능을 사용할 수 있도록 설계한다. 안드로이드 커널는 보안, 메모리 관리, 프로세스 관리, 네트웍 스택, 드라이버 모델 등과 같은 핵심 시스템 서비스를 위해 리눅스 버전 2.6에 기반 한다. 또한 이 커널은 하드웨어와 나머지 소프트웨어 스택간의 추상화된 계층으로서 역할을 수행 한다. 본 개발에 사용한 커널은 ARM linux 2.6.29.x 이다. 작업한 커널은 PC에서 이더넷 또는 USB로 시스템에 전송하며 이 작업은 부터로더가 수행한다.

안드로이드 루트파일시스템은 낸드플래시에 존재하며 안드로이드 스택(stack)을 포함한 루트파일 시스템이다. 낸드플래시에 저장될 때 yaffs저장되어 신뢰성 있는 파일 시스템으로 동작하게 된다. 작업한 루트파일 시스템은 PC에서 이더넷 또는 USB로 시스템으로 전송하여 저장이 가능하며 이 작업은 부터로더가 수행한다. 안드로이드 루트 파일시스템에는 이메일 클라이언트, 문자메시지 프로그램, 달력, 지도, 브라우저, 전화번호부, 그리고 다른 것들을 포함하는 핵심 애플리케이션들이 탑재되어 있다. 또한 안드로이드는 안드로이드 시스템의 다양한 컴포넌트에 의해 사용되는 C/C++ 라이브러리 집합을 포함하고 있으며, 모든 안드로이드 애플리케이션은 Dalvik 가상 머신내의 자신의 인스턴스를 가지고, 자신의 프로세스내에서 동작한다.

3. 디바이스 드라이버 프로그램 개발

개발 응용프로그램인 "쥬피터" 앱과 연동하여 지문인증 센서, 와이파이 무선랜, LCD, 오디오 등의 입출력 장치 들이 연결되어 작동 하도록 디바이스 드라이버 프로그램이 작성되어야 한다. 지문센서 드라이버 프로그램은 본 연구에서 직접 개발하여 사용한다. 지문 센서를 제외한 드라이버 프로그램 중 오디오 코덱 wm8680, LCD, LAN smsc9220은 리눅스 기본 커널(2.6.29)에 있는 것을 사용하였으며, WiFi 칩 드라이버는 기본 커널에 libertas라는 marvel 8686 드라이버을 사용한다. 작성된 센서드라이버를 커널에 포함시켜 컴파일 하지 않고 모듈의 형태로 추후 포함시켜 동작시키는 방법을 선택한다.

지문데이터는 지문센서(HB7122B)로부터 8비트 데이터 버스를 통하여 읽어오며, 이를 위한 제어 신호는 표준 I2C 시리얼 bus를 통한다. 지문센서 드라이버는 파일 입출력 함수 방법을 사용한다. 열기, 닫기, 읽기, 쓰기의 함수를 본 드라이버에서 제공한다.

```
static const struct file_operations hb7122b_fops =
{       .owner          = THIS_MODULE,
        .open           = hb7122b_open,
        .read           = hb7122b_read,
        .write          = hb7122b_write,
        .release        = hb7122b_release, };
```

hb7122b_read는 본 드라이버의 핵심으로 커널 메모리에서 사용자 메모리 영역으로 지문이미지를 복사하여 사용자 영역 프로그램이 사용할 수 있도록 해준다. hb7122b_open에서는 센서 초기화 기능을 하며, hb7122b_release 및 hb7122b_write는 특별한 기능이 필요치 않아 바로 리턴 한다. 디바이스 드라이버가 등록될 때 실행되는 함수인 hb7122b_init은 지문 1 프레임에 해당하는 메모리를 커널로부터 할당받고, 센서를 초기화 하며, 드라이버를 커널에 등록한다. 디바이스 드라이버가 커널로부터 해제되는 함수는 hb7122b_exit이며, 할당받은 메모리를 커널로 돌려주고 드라이버를 커널로부터 제거한다. 본 드라이버에서 이미지를 얻기 위해 필요한 기능은 크게 세가지로 나눌 수 있다. 본 센서를 제어하기위한 I2C 통신하는 부분, 여러 신호 입출력을 제어하는 GPIO 부분, 센서에서 출력되는 8비트 데이터 버스를 읽어들이는 부분으로 나눌 수 있으며 함수를 간략히

설명하면 아래와 같다.

. U8 osSetReg(U8 RegisterAddress, U8 WriteData)

I2C방식으로 레지스터 어드레스에 데이터를 출력한다. 통신 방식은 I2C 표준을 따른다.

. U8 osGetReg(U8 RegisterAddress, U8 *ReadData)

I2C방식으로 레지스터 어드레스의 데이터를 읽어 ReadData로 전달한다.

. int osReadFrame(U8 *pcImage)

I2C명령어로 센서의 soft reset을 해제한 후 Vsync 신호가 H가 될 때까지 기다린 후 Vsync 신호가 L이 되면 1 프레임 이미지가 전송되기 시작한다. 이미지의 1행을 연결된 데이터 버스로 보터 읽어낸다. 이때 센서에 할당된 virtual address로 접근해야한다. 256번 반복하여 256행의 이미지를 만든다. 256행 이미지를 받고 난 후 센서를 다시 reset상태로 만든다.

. void FingerLED(int i)

지문 영상을 받기 전 LED를 켜고, 영상을 받고 난 후 LED를 끄기 위한 함수이다.

. osInit

센서 초기화 함수로서 I2C 포트 설정 및 다른 GPIO 포트 설정을 하고 센서를 하드웨어 리셋한다. 그 후 영상의 밝기 조절을 위하여 Ingegration time을 조정한다. 마지막으로 지문을 받는 CMOS 셀의 영역을 설정한다.

• 커널 수정

하드웨어 설계 시 지문센서 HB7122B는 중앙처리장치 CS1에 연결하였다. 먼저 CS1를 읽기위하여 물리적주소인 0x38000000을 S3C6410_PA_HB7122B로 선언한다.

./arch/arm/mach-s3c6400/include/mach/map.h

```
#define S3C64XX_VA_HB7122B      S3C_VA_HB7122B
#define S3C64XX_PA_HB7122B      (0x38000000)
#define S3C64XX_SZ_HB7122B      SZ_4K
```

또한 사용자 프로그램에서는 물리적 주소를 사용하지 못하므로 할당이 가능한 가상주소 0x04000000에 매핑한다.

./arch/arm/plat-s3c/include/plat/map-base.h:

```
#define S3C_VA_HB7122B      S3C_ADDR(0x04000000)
```

둘을 연결시키는 다음과 같이 실행된다.

./arch/arm/mach-s3c6410/mach-mango6410.c:

```
static struct map_desc mango6410_iodesc[] = {
                IODESC_ENT(SMSC911X),
                IODESC_ENT(HB7122B), /* HB7122B*/ };
```

4. FingerLibary JNI 프로그램 작성

com_jupiter_main은 안드로이드 앱인 쥬피터의 클래스이다. 지문인증 화면 JNI 함수는 다음과 같다,

. Java_com_jupiter_main_FingerMain_devOpen

"/dev/hb7122b"를 open 한다.

. Java_com_jupiter_main_FingerMain_devRead

224*256 바이트의 메모리를 할당받고, "/dev/hb7122b" 드라이버에게 read명령어를 실행한다. read된 결과는 지문 1 프레임의 영상이며, read한 결과를 Java와 연결되는 공간에 리턴한다.

. Java_com_jupiter_main_FingerMain_getMinutiae(*)

함수 파라메터인 pbImage는 224*256의 지문 이미지 1 프레임이며, 이를 지문의 특징점을 추출하는 함수를 호출하여 400 바이트의 특징점으로 바꾼다. 이렇게 바꾼 특징점 400 바이트를 Java가 인식하는 공간에 리턴한다.

. Java_com_jupiter_main_FingerMain_matchFingerAuth(*,*)

함수 파라메터인 pbMinutiae1, pbMinutiae2는 각각의 지문 특징점이며 이를 지문을 비교하는 함수를 호출하여 비교한다. 그 결과는 1또는 0이며 이를 jint로 리턴한다. pbMinutiae1, pbMinutiae2중 하나는 현재 센서에서 읽어오는 영상에서 특징점을 추출한 것이며(_FingerMain_getMinutiae를 호출하여 변환한 것임), 다른 하나는 데이터베이스에 저장되어있는 것을 불러온 것이다.

. Java_com_jupiter_main_FingerMain_raw2BMP

이 함수는 센서에서 읽어온 영상을 bmp 구조로 변환하는 기능을 한다. 먼저 bmp 헤더의 규정에 따라 헤더를 추가하였다. 이때 bmp는 상하 반전이 기본이므로 높이를 음수값으로 설정하여야한다. 또 본 센서의 이미지는 1바이트 그레이 레벨이므로 RGB로 변환하는 게 필요하다. 아래처럼 RGB에 동일한 값을 저장하므로써 변환이 가능하다.

```
for(j=0;j<IMAGEX*IMAGEY;j++) {
        cstrBMP[i++] = nativeImage[j]; //R
        cstrBMP[i++] = nativeImage[j]; //G
        cstrBMP[i++] = nativeImage[j]; //B
        cstrBMP[i++]=0;        }
```

지문 인증화면이 아닌 지문 등록화면에도 위와 유사한 기능이 필요하다. JNI에서는 함수명에 반드시 화면이름(activity)를 추가하여야 인식하므로 동일한 기능을 가진 함수를 추가하였다. 지문 특징점 추출, 지문 특징점을 비교하는 함수 역시 NDK에서 작성하지만, 함수 형태는 일반 C와 동일하다. 왜냐하면 Java와 연결할 필요가 없기 때문이다. FingerLibary 모든 함수들의 소스를 compile 하면 *.so 형태로 앱에 추가된다.

5. 지문인식 알고리즘

지문 인식 알고리즘은 일반적으로 특징 추출과 지문매칭의 단계를 거친다. 특징 추출 단계에서는 특징점 데이터를 구성하기 위하여 전처리를 통한 지문 융선의 방향성을 사전 정보로 이용하여 특징점을 추출하고 후처리를 한다[8-10]. 지문 매칭 방식은 크게 특징점 정보를 이용한 정합과 전역 정합 방식이 있다. 특징점 매칭방식은 모든 특징점을 연결하는 거리의 합이 최소가 되도록 MST (Minimal Spanning Tree)를 이용하여 유사성을 찾는 방법이 있으며, 또는 특징점의 방향을 기준으로 좌표계를 설정한 다음 좌표계내에 존재하는 가장 가까운 특징점과의 거리, 각도, 등을 정합에 이용하는 방법이 있다[10-12].

본 연구개발에서는 특징점을 추출하고 국부좌표계를 이용한 지문매칭 방법[8-10]을 이용하여 지문인식 알고리즘을 설계한다.

6. 지문인증 어플리케이션 개발

개발 응용프로그램인 쥬피터는 지문인증 관련 메뉴, 사용자리스트 화면, 지문 등록 메뉴, 기기 설정 메뉴(근태모드, 도어모드 등)등으로 구성하며 메뉴키 및 복귀키로 이동 가능하게 한다. 안드로이드에서 제공하는 화면 처리 기능(surface manager), 미디어 프레임(media framework), 데이터베이스 엔진(SQLite) 등[13-17]을 이용하여 앱(application)을 개발함으로서 간편성 및 신뢰성을 확보한다.

• 어플리케이션 개발 시 사용 안드로이드 스택

본 연구에서 개발한 어플리케이션 프로그램인 "쥬피터"는 지문인증, 지문등록, 사용자리스트 등으로 구성 되어 있으며 개발에 사용한 안드로이드 스택 구성은 그림 2와 같다. 여기서 지문센서 드라이버 프로그램인 FingerPrint Driver 및 지문 인증에 관련된 JNI 프로그램인 FingerPrint 프로그램은 본 연구에서 개발한 프로그램을 이용한다.

• 어플리케이션 동작 알고리즘

지문 인식 어플리케이션의 동작 알고리즘은 그림 3과 같다.

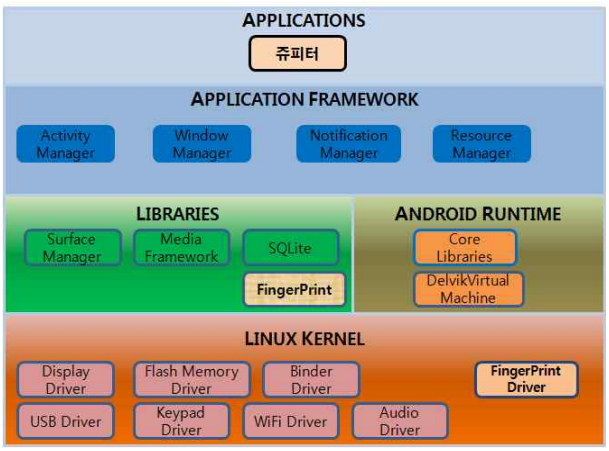

그림 2. 응용프로그램 개발시 사용한 안드로이드 스택.

Fig. 2. Android stacks used for application program.

그림 3. 응용프로그램의 동작 알고리즘.

Fig. 3. Execution algorithm of application program.

(i) 지문센서에 지문 접촉시 라이브러리 스택의 FingerLibary(*.so)를 통하여 지문 디바이스 드라이버 프로그램인 FingerPrint의 devRead(*,*) 함수를 호출하여 지문프레임의 영상을 가지고 온다.

(ii) FingerLibary 클래스 getMinutiae 함수를 이용하여 지문센서로부터 읽어온 지문정보의 특징점을 추출하고, 데이터베이스에 저장된 지문정보를 읽어온다. 이때 라이브러리 스택의 SQLite에 있는 클래스 DataBaseHelper를 이용한다.

(iii) 지문센서로부터 읽어온 입력지문정보와 데이터베이스에 저장된 지문정보를 비교하여 지문인증을 판단한다. 이때 라이브러리 스택의 FingerLibary의 지문인증 함수 matchFingerAuth()를 호출한다.

(iv) 인증된 결과의 문자 메시지는 지문이미지와 함께 LCD에 출력 한다. 이 때 Application Framework 스택의 Activity Manger에 있는 TextView를 이용하여 LCD에 출력한다.

(v) 인증된 결과의 음성 메시지를 오디오에 출력 한다. 이때 라이브러리 스택의 Media Framework에 있는 MediaPlayer를 이용하여 오디오로 출력한다.

(vi) 인증결과는 중앙 서버컴퓨터에 전달한다. 이를 위해 TCP/IP 소켓통신을 위한 server 및 client 동작 프로그램을 작성한다.

• 지문인증 메인화면처리 프로그램

지문 인증 메인 Activity 화면 프로그램의 개발 방법은 다음의 과정으로 이루어진다.

- 날짜, 시간, 로고 구성 등은 xml로 프로그래밍 하며 setContView()를 호출하여 액티비티 화면의 콘텐츠를 안드로이드 뷰 위젯으로 채운다. 이때 부요소에 대하여 findViewById()를 호출하여 주어진 리소스 id의 안드로이드 뷰를 찾는다.
- 사용자 리스트, 리스트 삭제, 환경설정 등의 메뉴를 추가하기 위하여 Menu의 menu.add()로 메뉴들을 추가한다. 또한 MenuItem의 item.getItemId()로 메뉴를 선택하게 한다.
- showDialog()를 호출하여 각 메뉴에 대한 대화 상자를 활

성화 하게 하며, AlertDialog.Builder를 이용하여 다이얼로그를 만든다.
- 지문센서 접촉시 지문센서 정보를 읽어오기 위하여 FingerLibary의 devRead() 함수를 호출하여 지문프레임 영상 mScanImage를 가지고 온다. 또한 FingerLibary의 getMinutiae() 함수를 호출하여 특정점 추출 이미지 mScanMinutiae를 만든다.
- DatabaseHelper의 mDbHelper.fetchAllNotes()를 이용하여 데이터베이스 검색해서 DB에 저장된 지문이미지 mRegMinutiae를 가지고 온다.
- FingerLibary 의 matchFingerAuth()를 호출하여 지문센서의 추출이미지 mScanMinutiae와 DB저장 이미지 mRegMinutiae를 비교하여 출입 인증을 여부를 판단한다.
- 출입인증시 인증되었다는 메시지 및 등록자가 없다는 메시지는 TextView의 mresult.setText()를 이용하여 출력한다.
- 인증되었다는 sound 출력은 MediaPlayer의 audio_play.setDataSource()로 play할 mp3 파일을 정하는 것으로 sound 출력 작업을 시작하며, audio_play.prepare()한 다음 audio_play.start()로 음악이 시작된다.
- 인증시 음성 출력 및 서버로 출력하는 부분은 음성출력 프로그램밍 및 TCP/IP 통신 프로그래밍 작성에서 상세히 나타낸다.

• 지문 사용자 리스트 프로그램

지문 사용자 리스트 Activity 화면 프로그램의 개발 방법은 다음의 과정으로 이루어진다.

- 사용자 LIST를 보여 주는 부분으로서 SetContView()를 호출하여 작성한다.
- 화면 아래 부분에 사용자추가, 모든 사용자 삭제 등의 메뉴를 추가하기 위하여 Menu의 menu.add()를 사용하여 작성한다.
- 메뉴의 사용자 추가를 클릭하여 사용자를 등록하는 부분은 cursor의 c.moveToPosition() 및 c.getColumnIndexOrThrow() 등을 이용하여 작성한다.

• 지문 사용자 등록 프로그램

지문 등록 화면을 보여주는부분으로 setContView()를 호출하여 액티비티 화면의 콘텐츠를 안드로이드 뷰 위젯으로 채운다.

- 지문 등록을 유도하는 sound 출력은 mp3 file의 위치를 MediaPlayer의 audio_play.setDataSource()로 play할 파일을 정한 후 audio_play.start()로 지문 등록을 유도하는 sound 출력이 시작된다.
- 지문센서 접촉시 지문센서 정보를 읽어오기 위하여 FingerLibary의 devRead() 함수를 호출하여 지문프레임 영상 mScanImage를 가지고 온 후 ImageView의 setImageBitmap() 함수를 이용하여 화면에 출력한다. 또한 FingerLibary 의 getMinutiae() 함수를 호출하여 특정점 추출 이미지 mMinutiae를 만든다.
- 추출된 이미지 mMinutiae 및 User id는 SQLiteOpenHelper를 이용하여 DB에 등록 한다.
- 지문 등록이 완료되었다는 sound 출력은 MediaPlayer의 audio_play.setDataSource() 및 audio_play.start()로 sound를 출력한다.

• 디바이스 설정 UI 프로그램

디바이스 설정 Activity 화면 프로그램의 개발 방법은 다음의 과정으로 이루어진다.

- 근태시간 설정, 출입문 모드, 위겐드 입출력 설정, 문영릴시간 설정 등의 메뉴는 Xml로 작성하여 setContView()를 호출하여 디바이스 설정 Activity화면을 뷰 위젯으로 채운다.
- 뷰 요소에 findViewById()를 호출하여 주어진 리소스 id의 뷰를 찾는다. 또한 View의 onClickListener()를 이용하여 뷰를 건드리거나 클릭했을 경우 해당하는 객체를 작동시킨다.
- Editor의 editor.putInt() 및 editor.putString()를 이용하여 근태시간 및 출입문 모드 등의 데이터를 설정한다.

• 음성 출력 프로그램 작성

Sound 출력은 두 개의 activity(UI를 가지는 화면)에서 사용하고 있다. 지문 인증후의 결과와 지문을 등록하는 화면에서 각각 사용하고 있다. 지문 등록의 경우 지문 등록을 유도하는 사운드와 등록이 완료됨을 알리는 사운드를 출력한다. 사운드는 mp3 파일이며 MediaPlayer를 사용하여 play한다. 사운드 출력의 경우 mp3 file을 위치를 MediaPlayer의 audio_play.setDataSource()로 play할 파일을 정하는 것으로 sound 출력 작업을 시작하며, audio_play.prepare()한 다음 audio_play.start()로 음악이 시작된다. 이미 음성이 출력되고 있는 경우 audio_play.stop() 및 audio_play.release() 함수를 이용하여, 그 음성을 중단하고 새로운 음성을 시작해야 한다. 메인화면이 종료될 때 사운드 자원을 반환해야한다.

• 언어 다중화 프로그램

본 제품에서는 영어와 한글을 기본적으로 지원한다. 언어 전환은 기기 설정 프로그램으로 간단하게 할 수 있다. 그러나 안드로이드 앱의 경우 미리 리소스를 각 언어에 대응하여야만 언어 전환이 가능하다. 다중언어를 지원하기 위해서 폴더에 각국 언어의 문자열을 정의해야 한다. 언어 전환은 먼저 select local을 선택하고, 그 후 전환하고자 하는 언어를 선택한다.

• TCP/IP 통신 프로그램 작성

본 연구개발 제품에서는 서버컴퓨터와의 통신으로TCP/IP socket통신을 사용한다. 인터넷통신을 통신을 하기 위해서 먼저 안드로이드 Manifest.xml에서 INTERNET pemission을 아래와 같이 추가한다.

```
<uses-permission android:name = "android.permission
.INTERNET"> </uses-permission>
```

Client 동작 프로그램은 지문인증 결과를 server로 보내는 동작을 구현하는 것이다. 이에 반응하는 server 컴퓨터에서 처리하는 프로그램은 JupiterAccess이다. Client 동작 프로그램은 다음과 같다. 먼저 socket 통신을 하기 위하여 Server의 IP 및 port 번호를 안드로이드 code에 추가하고 thread를 발생시켜 socket통신이 가능하게 한다. 다음, 인증 결과에 따라 사운드를 출력하고, 사용자가 등록되지 않은 경우, 인증이 실패한 경우, 인증에 성공한 경우 각각 해당 문자열을 byte형태로 만들어 서버로 전송한다. 전송하는 코드는 out.write(w), out.flush() 함수를 이용하여 실제 문자열 또는 packet으로 서버에 전송한다.

III. 성능평가

그림 4는 본 연구에서 개발한 개발 시제품이며, 개발 시제품에 대한 성능평가는 다음과 같다.

• 지문인증 결과

개발 프로그램에 대한 지문인식장치의 LCD 출력 화면은 그림 5와 같으며, 등록된 4번 사용자가 센서 접촉 시 지문 인증에 성공 한 경우에 대한 결과이다.

그림 6은 등록된 사용자가 없는 경우와 등록되지 않은 지문을 접촉한 경우 인증 하지 않은 결과를 나타내고 있다.

• 지문 사용자 리스트 출력 화면

지문 사용자리스트 프로그램에 대한 출력화면은 그림 7과 같다.

• 지문 사용자 등록 화면

지문 등록 프로그램에 대한 출력화면은 그림 8과 같다. 지문센서에 지문 접촉 후 사용자 ID를 등록하는 화면이다.

• 디바이스 설정 화면

디바이스 설정에 대한 개발 프로그램의 출력 화면은 그림 9와 같다. 도어 모드 등을 선택함으로서 근태관리기 등으로 다양하게 사용할 수 있다.

• 언어 다중화

그림 10은 한글 및 영어에 대한 쥬피터 화면을 나타내고 있다. 해당 언어 선택만으로 디바이스 설정 화면이 영어에서 한글로 바뀌어서 출력됨을 알 수 있다.

그림 4. 개발보드.

Fig. 4. Development board.

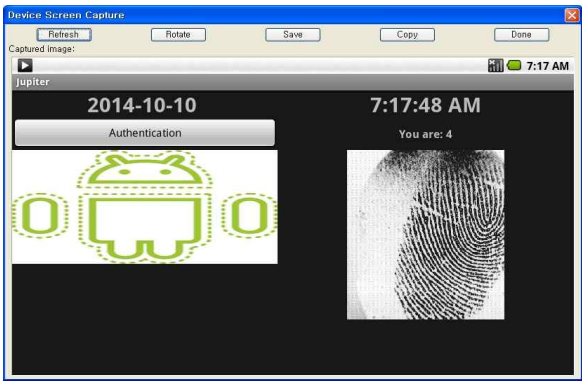

그림 5. 지문인증 성공시의 인증화면.

Fig. 5. Output picture for pass of FRS.

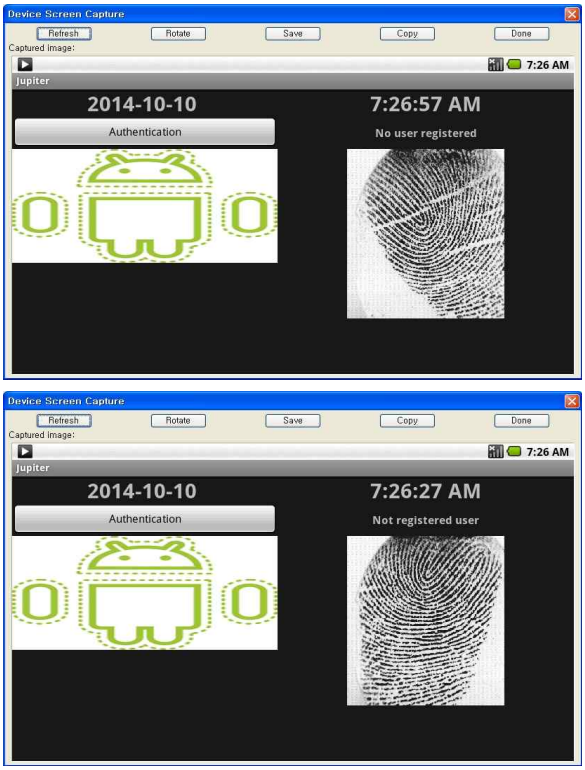

그림 6. 지문인증 실패시의 인증화면.

Fig. 6. Output picture for failure of FRS.

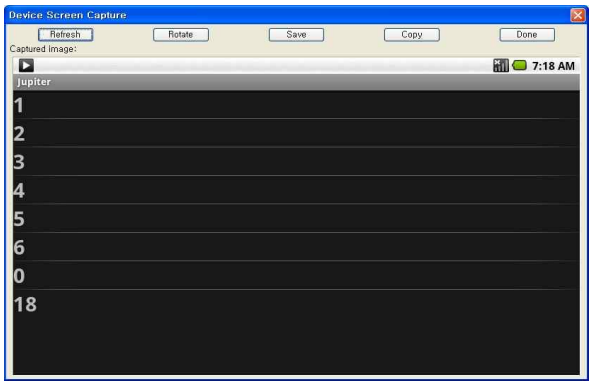

그림 7. 사용자 리스트 출력화면.

Fig. 7. Output picture for list of users.

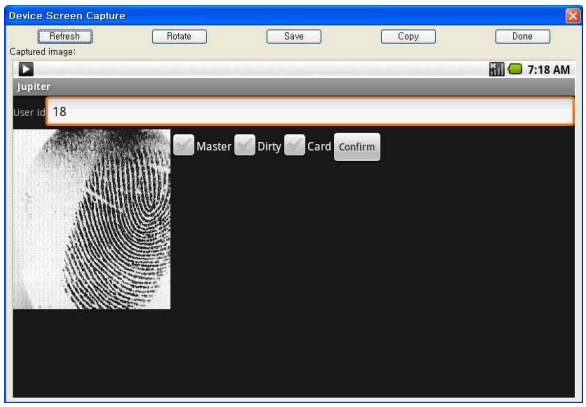

그림 8. 지문사용자 등록화면.

Fig. 8. Registration picture for list of users.

그림 9. 디바이스 설정화면.

Fig. 9. Setting picture for devices.

그림 10. 언어다중 화면.

Fig. 10. Multi-lingual picture.

• TCP/IP 통신 성능

서버 컴퓨터와 지문 인증 단말기를 같은 네트워크 그룹에 연결하고 지문 인증 시험을 하였다. 그림 11은 등록 사용자가 인증에 성공했을 경우의 본 제품인 지문 인증 장치에 나타난 결과와 서버컴퓨터에서 나타난 결과를 보여 주고 있다.

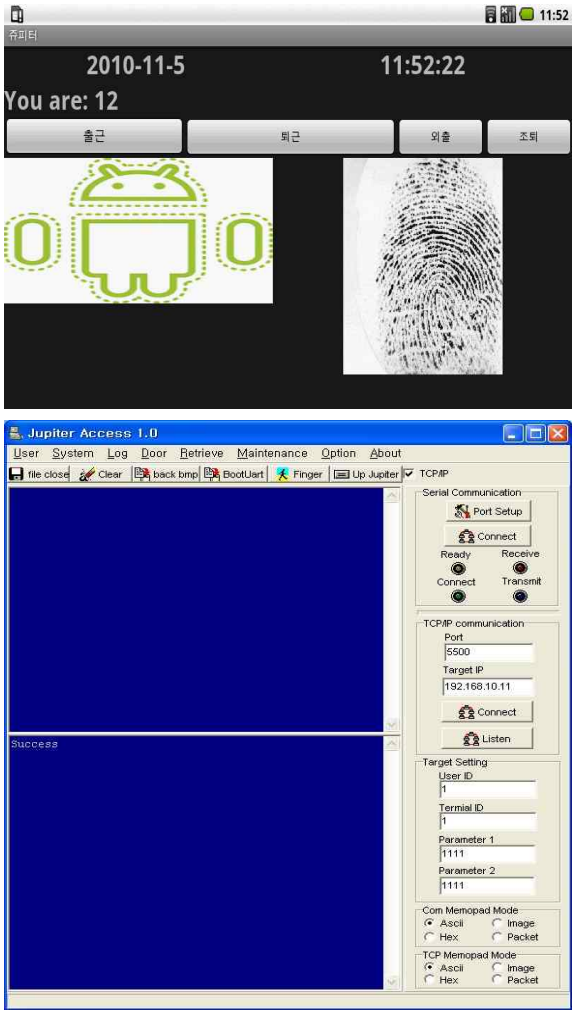

그림 11. TCP/IP 통신 출력화면.

Fig. 11. Output picture for TCP/IP communication.

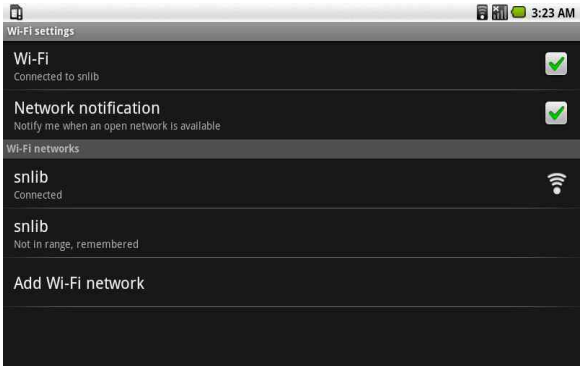

그림 12. 무선 와이파이 연결 화면.

Fig. 12. Output picture for WiFi Communication.

• 와이파이 연결

와이파이 통신 칩이 적절하게 연결되어 있으며 구동 드라이브 프로그램이 잘 동작하고 있으면 와이파이 설정은 기기 설정 프로그램으로 활성화할 수 있다. 와이파이를 클릭하여 범위내에 있는 와이파이를 검색하면 그림 12와 같이 검색이 되며 가장 신호가 강한 무선 AP에 자동으로 연결된다. 와이파이를 클릭하여 동작시켜 강한 무선 AP (Access Point)에 연결한 화면이다. 와이파이연결 후에 안드로이드 기본 앱인 브라우저로 인터넷 검색을 할 수 있다.

• 기타 성능 시험

본 연구개발 시스템 설계에 포함된 하드웨어 장치 이내에서 구동 될 수 있는 각종 안드로이드 응용 프로그램은 본 개발시제품에서 사용 할 수 있다. SD 메모리에 있는 사진의 배경화면 설정 및 SD 메모리에 있는 mp3 파일 재생 등은 안드로이드에 기본적으로 포함되어 있는 응용 프로그램으로 사용 가능하다.

IV. 결론

본 개발에서는 안드로이드 플랫폼을 탑재한 스마트 지문인식장치를 개발하였다. 지문 인식을 500 mSec 이내로 구현하기 위하여 32bit RISC 중앙처리장치(CPU)를 사용하며, 지문인식 센서는 물리적으로 안정하다고 알려져 있는 광학식 단말기를 사용하였다. 또한 터치패드를 사용해 입력키의 개수를 줄여 수려한 디자인이 가능케 하였으며, WiFi 기능을 탑재하여 무선 네트워크가 가능한 지문인식기를 개발하였다. 인증자료들은 단말기 플래시 메모리에 저장된다. 지문 인증 소프트웨어 개발을 위해 지문 디바이스 드라이버 및 이를 제어하는 JNI 프로그램을 개발하였다. 또한 개발한 이 지문 JNI 프로그램과 안드로이드 라이브러리를 이용하여 지문 인증 애플리케이션 프로그램을 개발하고 개발한 제품의 성능시험 평가를 하였다. 안드로이드 플랫폼은 멀티미디어 플레이(동영상 및 mp3 파일 재생)기능을 포함하고 있어 본 개발 제품은 출입이 없는 평상시에는 광고홍보용으로 사용 가능하며, 무선 네트워크가 가능하므로 중앙 서버 컴퓨터와의 연동으로 근태관리기 등의 복합장치로도 사용가능하다.

참고문헌

[1] E. B. Yi, S. W. Jun, C. W. Ryu and H. I. Kim, “Development of a Fingerprint Recognition System for Various Fingerprint Image,” *Journal of Electronics Engineers of Korea,* vol. 40, SP, no. 6, pp. 10-19, 2003.

[2] D. Ryu, S. J. Shin and B. H. Kim, “Implementation of a Fingerprint Identification System in Wireless Environments,” *Journal of Electronics Engineers of Korea,* vol. 42, TE, no. 1, pp. 53-59, 2005.

[3] K. R. Lee, “The study on design of smart embedded system with WiFi for industrial information instrument,” *Journal of Natual Science in The Pyeongtaek University*, 2010.

[4] K. R. Lee, “Development of Device Drive and Application for Android Fingerprint Recognition Apparatus,” *Proc. of Electronics Engineers of Korea*, Nov. 2011.

[5] Samsung, "S3C6410 Application Processor User's manual Rev.1.2," Samsung, 2009.

[6] Ed Burnette, "Introducing Google's Mobile Development Platform," Android 2.1, Edition. 2010.

[7] Ed Burnette. *Eclipse IDE Pocket Guide,* O'Reilly & Associates, Inc, Sebastopol, CA, 2005.

[8] A. K. Jain, L. Hong, and R. Bolle, "On-line Fingerprint Verification," Research Report, Michigan State University, Department of Computer Science. 1996.

[9] A. K. Jain, L. Hong, and Y. Wan, “Fingerprint image enhancement: algorithm and performance evaluation,” *IEEE Transaction on Pattern Analysis and Matching Intelligence*, vol. 20, no. 8, pp. 777-789, 1998.

[10] D. Maitoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd Ed., Springer, 2009.

[11] H. Motulsky and A. Christopoulos, *Fitting Models to Biological Data Using Linear and Nonlinear Regression*, Oxford University Press, 2003.

[12] J. H. Chang and K. C. Fan, “A new model for fingerprint classification by ridge distribution sequences,” *Pattern Recognition*, vol. 35, no. 6, pp. 1209-1223, Jun. 2002.

[13] J. Gennick, *SQL Pocket Guide,* O'Reilly & Associates, Inc, Sebastopol, CA, second edition, 2006.

[14] M. Owens, *The Definitive Guide to SQLite. Berkeley*, CA: Apress, 2006.

[15] http://d.android.com/guide/index.html.

[16] http://d.android.com/reference/android/graphics/package-summary.html.

[17] http://d.android.com/guide/topiocs/media/index.html.

이 갑 래

1987년 경북대학교 전자공학과 학사. 1990년 동 대학원 전자공학과 공학석사. 1999년 동 대학원 전자공학과 공학박사. 1990년~1996년 국방과학연구소 연구원. 2001년~평택대 정보통신학과 부교수. 관심분야는 지능시스템, 지능제어, 산업용 네트워크, 안드로이드 탑재 임베디드 시스템 설계.