

안드로이드 기반 상하좌우 방향의 동작 제스처를 선택할 수 있는 응용 프로그램 구현

전영남*

Implementation of Android-Based Applications that can Select Motion Gestures In Up, Down, Left, and Right Directions

Yeong-Nam Jeon*

요약

본 논문에서는 안드로이드 플랫폼 기반의 GRS칩 구동 JNI 코드 응용 SW 설계 안드로이드 플랫폼 기반 동작 제스처 프레임 모듈 설계 제작하였다. 안드로이드 기반 모듈 설계, 안드로이드 기반 모듈 구현, 안드로이드 기반 기능 모듈 구현 설계로 응용애플리케이션 기반 네트워크 지원 API 기술 제한한 직렬 데이터 수신 모듈 설계를 하였다. 안드로이드 OS의 애플리케이션을 통해 무선 통신 디바이스로부터 데이터를 수신하기 위한 직렬 통신 드라이버, 라이브러리, 프레임워크의 클래스와 같은 안드로이드 애플리케이션을 통해 센서의 데이터 정보를 확인할 수 있었다. 또한 안드로이드에서의 어플리케이션들은 자바를 이용하여 좌우상하 4방향의 동작 제스처를 판단할 수 있는 응용 SW를 구현하였다.

ABSTRACT

In this paper, GRS chip driven JNI code application SW design based on Android platform was designed and fabricated as motion gesture frame module based on Android platform. The serial data reception module design proposed by the application-based network support API technology was designed with Android-based module design, Android-based module implementation, and Android-based function module implementation design. The data information of the sensor could be checked through Android applications such as classes of serial communication drivers, libraries, and frameworks for receiving data from wireless communication devices through Android OS applications. In addition, applications in Android implement application SW that can judge motion gestures in four directions using Java.

키워드

Android Platform, GRS, JNI Code, Network Support API
안드로이드 플랫폼, GRS, JNI 코드, 네트워크 지원 API

* 교신저자 : 동강대학교 시용합기계과
• 접수일 : 2023. 07. 20
• 수정완료일 : 2023. 08. 31
• 게재확정일 : 2023. 10. 17

• Received : Jul. 20, 2023, Revised : Aug. 31, 2023, Accepted : Oct. 17, 2023
• Corresponding Author : Yeong-Nam Jeon
Donggang University, Dept. of AI Mechanical Coverage Eng.
Email : snower24@naver.com

1. 서론

안드로이드는 구글에서 만든 스마트폰용 운영체제이다. 운영체제와 미들웨어, 사용자 인터페이스, 어플리케이션, MMS 서비스 등을 하나로 묶어 서비스를 제공하며 다양한 어플리케이션을 만들어 설치하면 실행될 수 있도록 구성된 어플리케이션 플랫폼이라고도 볼 수 있다. 안드로이드는 오픈소스를 기반으로 하고 있으며 아파치 2.0라이선스를 사용한다. 완벽한 컴포넌트는 안드로이드는 오픈소스임에도 불구하고 스마트폰을 위한 완벽한 컴포넌트들을 제공한다. 핵심적인 커널 부분만 제공하고 다른 부분들은 개발자에게 맡기는 방식이 아니라 제공된 소스를 사용하면 스마트폰이라는 전화가 그대로 만들어지므로 대부분의 개발자들은 그 위에 올라가는 앱만 만들면 된다[1-2]. 안드로이드 플랫폼 기반 연동으로 만든 앱은 다른 앱에서 제공하는 기능을 쉽게 사용할 수 있도록 만들어져 있다. 사진을 찍기 위해 단말의 사진 앱을 연동하면 카메라 미리보기 화면을 직접 만들지 않아도 된다[3-5].

본 논문에서는 안드로이드 플랫폼 기반의 GRS칩 구동 JNI 코드 응용 SW 설계로 안드로이드 플랫폼 기반 공간 터치 Framework 모듈을 설계 제작하였다. 안드로이드 기반 모듈 설계 및 구현, 기능 모듈 구현 설계 및 구현으로 응용어플리케이션 기반 네트워크 지원 API 기술 제한한 직렬 데이터 수신 모듈 설계를 하였다.

II. 관련 연구

2.1 안드로이드 플랫폼 아키텍처

안드로이드 소프트웨어 스택은 소프트웨어 스택(stack) 형태로 구성된다. 마치 레고(Lego) 블록을 조립해 층층이 쌓아놓듯이 어플리케이션, 운영체제, 런타임 환경, 미들웨어, 각종 서비스와 라이브러리 등이 겹겹이 모여 구성된 것이라고 볼 수 있다. 이 아키텍처는 그림 1과 같이 안드로이드 소프트웨어 스택을 시각적으로 표현할 수 있다. 스택의 각 계층과 요소는 긴밀하게 통합되어 있다. 그리고 모바일 장치를 위한 최적의 어플리케이션 개발과 실행 환경을 제공하기 위해 신중하게 조율되어 있다[4-7].

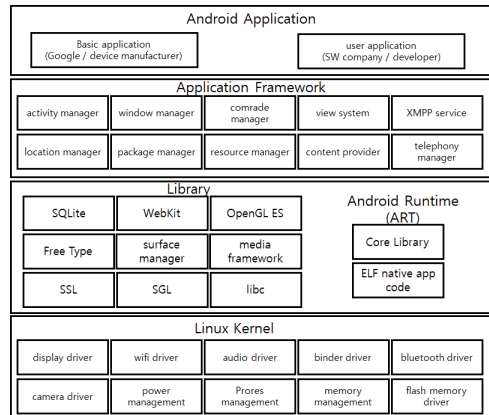


그림 1. 안드로이드 소프트웨어 스택
Fig. 1 Android Software Stack

2.2 안드로이드 라이브러리

문자열 처리, 네트워킹, 파일 처리와 같은 일반적인 작업을 지원하기 위해 제공되는 표준 자바 라이브러리에 추가하여 안드로이드 개발 환경에는 안드로이드 라이브러리도 포함된다. 이것은 안드로이드 어플리케이션 개발에 특화된 다양한 자바 기반 라이브러리이다. 이 부류의 라이브러리 예로는 어플리케이션 프레임워크 라이브러리를 포함하여 사용자 인터페이스 생성, 그래픽 드로잉, 데이터베이스 액세스 등을 가능하게 해주는 라이브러리가 있다. 안드로이드 개발자들이 사용 가능한 핵심 안드로이드 라이브러리를 요약하면 아래와 같다[8-9].

- android.app - 어플리케이션 모델의 액세스를 제공하며, 모든 안드로이드 어플리케이션의 초석이 되는 라이브러리.
- .android.content - 어플리케이션과 어플리케이션 컴포넌트 간의 콘텐츠 액세스와 메시징을 가능
- .android.database - 콘텐츠 제공자가 게시한 데이터를 액세스하는 데 사용되고, SQLite 데이터베이스 관리 클래스를 포함
- android.graphics - 색상, 포인트, 컬러, 사각형, 캔버스를 포함하는 낮은 수준의 2D 그래픽 드로잉 API.
- android.hardware - 가속도 센서와 광 센서 같은 하드웨어의 액세스를 제공하는 API.
- android.opengl - OpenGL ES 3D 그래픽 렌더링

API의 자바 인터페이스.

- android.os - 메시지, 시스템 서비스, 프로세스 간 통신을 포함하는 운영 체제 서비스의 액세스를 애플리케이션에 제공.
- android.media - 오디오와 비디오의 재생을 할 수 있는 클래스를 제공.
- android.net - 네트워크 액세스를 제공하는 API, 장치의 무선 네트워크 액세스를 제공하는 android.net.wifi를 포함.
- android.print - 안드로이드 애플리케이션에서 프린터로 콘텐츠를 전송할 수 있는 클래스들을 포함
- android.provider - 캘린더나 연락처 애플리케이션에서 유지하는 것과 같은 표준 안드로이드 콘텐츠
- android.text - 장치 화면에 텍스트를 나타내고 처리하는 데 사용
- android.util - 문자열과 숫자 변환, XML 처리, 날짜와 시간 처리 등의 일을 수행하는 유틸리티 클래스
- android.view - 애플리케이션의 사용자 인터페이스를 구성하는 뷰 클래스
- android.widget - 버튼, 라벨, 리스트 뷰, 레이아웃 매니저, 라디오 버튼과 같은 사전 제작된 사용자 인터페이스 컴포넌트
- android.webkit - 웹 브라우징 능력을 애플리케이션에 구성할 수 있게 해주는 클래스

2.3 안드로이드 기능

안드로이드는 현재 가장 대표적인 스마트폰 운영체제 중 하나로 구글에서 만들었으며 거의 80%에 가까운 시장 점유율을 가지고 있다. 안드로이드 특징안드로이드는 운영체제(OS: Operating System) 혹은 플랫폼(Platform)등으로 불리웁니다. 의미에 따라서는 같을수도 있고 다를수도 있다. 안드로이드의 주요 기능과 특징은 다음과 같다[9-11].

- 지원 네트워크: GSM/EDGE, CDMA, EV-DO,UMTS, Bluetooth, Wi-Fi
- 핸드셋 레이아웃: 중대형 사이즈의 디스플레이,VGA, 기존의 스마트폰 레이아웃 지원
- 그래픽 지원: 2D 그래픽 지원, OpenGL ES 1.0 기반 3D 그래픽 지원

- 저장장치: SQLite 사용
- 멀티터치: 애플의 터치스크린 특허 침해를 방지하기 위해 커널에서 멀티터치기능 제한
- 메시징: SMS, MMS, Threaded 문자메시지 지원
- 웹브라우저: 오픈 소스 웹킷(WebKit) 방식 브라우저
- 자바 지원: Dalvik Virtual Machine을 통해 자바 실행
- 멀티미디어 지원: H.263, H.264, MPEG-4 SP,AMR, AAC, AMR-WB, AAC, HE-AAC,MP3, MIDI, OGG, WAV
- 개발환경: 에뮬레이터, 디버깅 툴, 메모리/성능 프로파일링, Eclipse IDE를 위한 플러그인
- 하드웨어 지원: 카메라, 터치스크린, GPS, 가속계 및 자기계, 3D 그래픽 가속기
- 어플리케이션 스토어: PC 다운로드 는 지원되지않고 OTA를 통한 다운로드 및 설치 가능. 원래는 무료 소프트웨어로만 운영할 예정이었으나 2009년 2월 19일부터 미국 안드로이드 마켓에서 유료 아이템 판매 시작

안드로이드의 여러 구성요소를 포함하면 다양한 환경에 적용 가능한 플랫폼이 된다. 예를 들어 안드로이드를 스마트폰이 아닌 다른 하드웨어나 시스템에서도 활용이 가능하다. 이 경우 완전히 다른 분야의 하드웨어라도 동일한 안드로이드 사용자 경험과 플레이스토어, 앱 등을 공유할 수 있게 된다. 안드로이드의 특징은 다음과 같다[10].

- 운영체제 커널이 리눅스
- 가장 대표적인 프로그램 언어인 자바로 응용 프로그램 개발이 가능
- 오픈소스 지향으로 많은 정보가 공개되어 있고 적용
- 안드로이드 스튜디오라고 하는 전용의 개발 도구를 무료로 제공
- Firebase 등 구글 클라우드 서비스와 손쉽게 연동이 가능
- 플레이스토어를 통해 앱을 공개하고 판매가 가능

2.4 개발환경 비교

표 1에서 대표적인 스마트폰 운영체제의 개발환경 비교이다. 윈도우의 경우 마이크로소프트에서 만들었지만 점차 시장에서 사라지는 추세이며 마이크로소프트

트는 안드로이드 관련 특허들을 통해 많은 라이선스 수입을 가져가고 있으며 최근에는 안드로이드 폰과의 연결성을 확대해 나가고 있다. 또한 개발의 경우에도 Xamarin 등을 통한 멀티 플랫폼 앱 개발쪽으로 방향이 진행되고 있다[11].

표 1. 스마트폰 개발환경 비교
Table 1. Comparison of smartphone development environment

Division	Iphone	Android	windows phone
Development Language	Swift, Objective-C	Java, Kotlin	C#
Development os	Mac OS	Windows, Linux, Mac OS	Windows
Development Tool	XCode	Android Studio	Visual Studio
Version	iOS 13	10.0(Q)	Windows Phone 10
Anufacturer/Store	One manufacturer, one app store	Multi-Manufacturer, Multi-App Store	Multi-manufacturer, single app store

개발 관점에서 안드로이드의 장점은 윈도우, 맥, 리눅스 등에서 동일한 개발도구를 가지고 쉽게 개발할 수 있다는 점이다. 또한 많은 제조사들의 하드웨어와 응용 시스템에서도 활용 가능하므로 개발 수요가 높고 적용 분야가 많다는 점을 들 수 있다. 반면 IOS의 경우 맥 컴퓨터만을 사용해야 하며 Swift 라고 하는 새로운 언어와 XCode 라고 하는 개발도구를 사용해야 되기 때문에 진입장벽이 높은 단점이 있다. 그렇지만 단일 하드웨어로 인해 개발에 고려할 사항이 상대적으로 적은 편이며 지속적이고 체계적인 운영체제 업그레이드 정책으로 인해 대부분의 사용자들이 최신 운영체제를 적용하고 있다는 점은 장점으로 꼽을 수 있다. 또한, 아이폰 사용자들의 높은 앱 구매력으로 인해 개발업의 유료 수익을 추구하는 경우 아이폰을 따라 갈 수 없다[12-15].

III. 개발 모듈 설계

3.1 안드로이드 플랫폼 GRS칩 구동 JNI 코드 응용 SW 모듈

본 논문에서는 안드로이드 플랫폼 GRS칩 구동 JNI 코드응용 SW 모듈 설계하였다. 안드로이드 기반

모듈 설계로는 응용애플리케이션 기반 네트워크 지원 API 기술 제안한 직렬 데이터 수신 모듈 설계를 하였다. 응용애플리케이션 기반 네트워크 지원 API 기술 설계는 API는 리소스에 대한 액세스 범위를 넓히는 동시에 보안과 제어를 유지 및 액세스 권한을 어떻게, 누구에게 제공 여부만 결정한다. 모든 시스템 환경에 연결하는 분산형 통합 플랫폼을 통해 API에 연결하고 API에 의해 노출된 데이터 사용이 가능하다.

소프트웨어 프레임워크 리눅스 커널, init, 컨텍스트 매니저, 미디어 서버, Zygote, 시스템 서버와 같이 시스템 서비스를 실행하는 역할을 할 수 있도록 설계한다. 직렬 통신용 클래스가 정의는 직렬 클래스의 직렬 통신 기능을 정의한다. 직렬 통신 한 번에 한 비트씩 전송한다. 클래스를 사용하여 직렬 통신이 가능하고 연결하고 데이터 전송이 이루어진다. 제어 흐름이 프레임워크의 코드에 상주한다. 태그 컴파일, 코드 작성, 전체 애플리케이션 통하여 컴파일한다.

안드로이드 기반 기능 모듈 구현을 자바 프로그램을 이용하여 상하좌우 방향의 동작 제스처를 선택할 수 있는 응용 프로그램 구현하였다. 안드로이드 상하좌우 동작 제스처 방향 판단 구현 기능 모듈 구현, 안드로이드 플랫폼 기반 기기의 상하좌우 방향 및 속도 동작 제스처 방향 판단 구현 기능 모듈 구현, 안드로이드 플랫폼 기반 GRS의 근접 동작 제스처 방향 판단 기능 모듈 구현, 안드로이드 플랫폼 환경 기능 모듈 구현 상하좌우 방향의 이동 동작 제스처를 판단 할 수 있는 응용 프로그램을 구현하였다. 또한, 안드로이드 기반 스마트 기기의 상하좌우 속도별 단계 동작 제스처 방향 판단 구현으로 상하좌우 이동 동작 제스처 방향 판단 구현 및 스마트 기기에서 상하좌우 동작 제스처 단계별 속도 막대 바 판단 구현 UI 구현하였다.

3.2 제안한 직렬 데이터 수신 모듈 설계

제안한 직렬 데이터 수신 모듈 설계안드로이드 기반 응용애플리케이션 네트워크 지원 API는 리소스에 대한 액세스 범위를 넓히는 동시에 보안과 제어를 유지, 액세스 권한을 누구에게 제공 여부만 결정한다. 응용 애플리케이션 기반 네트워크 지원 API 기술 설계는 안드로이드 기반 모듈 설계로 응용애플리케이션을 위한 네트워크 지원 API 직렬 데이터 수신 모듈 설계는 제안한 그림 2와 같다.

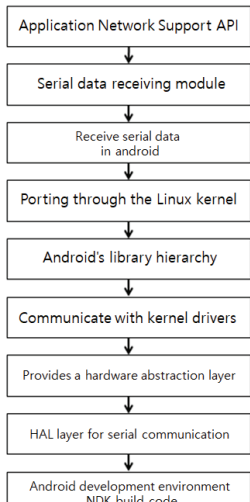


그림 2. 응용애플리케이션을 위한 네트워크 지원 API 직렬 데이터 수신 모듈 설계 절차
 Fig. 2 Network Support API Serial Data Receiving Module Design Procedure for Applications

가) 게이트웨이 사용 레거시 시스템과 사물 인터넷은 시스템 환경에서 연결하는 분산형 통합 플랫폼을 API에 연결하고 API에 의해 노출된 데이터 또는 기능을 사용할 수 있다.

나) 안드로이드 기반 응용애플리케이션 직렬 데이터 수신 모듈은 직렬 통신으로 수신되는 데이터 직렬 통신은 PC와 PCB 보드 간의 통신한다.

다) 안드로이드 운영체는 직렬(USB) 데이터 수신되는 텍스트 데이터는 Hex(16진수) 포맷으로 변경된다. PC에서 안드로이드 데이터 전송한다. 안드로이드 기기와 PC를 직렬 통신 연결해주고 GRS칩에서 각각 직렬 통신 어플리케이션을 실행한다.

라) 리눅스 커널을 통해 포팅은 호스트 PC와 타겟 보드 연결 후 여기서 기억하고 넘어갈 내용은 호스트 PC와 GRS칩 구동에서는 컴파일만 실행하고 동작은 PCB 보드에서 진행한다. UART 직렬 통신, TFTP 파일 시스템 통신, NFS 네트워크 파일 시스템을 연결한다. 타겟보드 확인 정보를 통하여 환경 구성 및 커널 컴파일한다.

마) 안드로이드의 라이브러리 계층으로 리눅스 커널의 총 5개 계층으로 분류한다. 안드로이드의 라이브러리는 시스템 전체의 중심 역할을 한다.

바) 커널의 드라이버 통신은 프로그램 커널과 레벨

드라이버와 상호간에 데이터 통신한다. GRS칩 구동을 위하여 I/O Control(IOCTL) 명령으로 애플리케이션과 커널 드라이버간의 통신 데이터를 정의한다.

사) 하드웨어 추상화 계층 제공은 하드웨어가 변경
 야) 프로그램 소프트웨어 작동 플랫폼을 제공한다.

호출 프로그램은 GRS칩 장치와 일반적인 방식으로 상호 작용할 수 있다.

자) 직렬 통신을 위한 HAL 계층은 운영체계가 하드웨어 계층이 아닌 하드웨어 서비스와 상호 작용할 수 있다. 마지막으로 안드로이드 개발환경 NDK 빌드 코드 Android에서 NDK를 이용하여 JNI 코드 앱이 컴파일한다.

IV. 안드로이드 플랫폼 기반 응용 SW 구현

4.1 제안한 직렬 데이터 수신 모듈 구현

본 논문에서는 안드로이드 기반 모듈 구현으로 안드로이드 운영체는 직렬(USB) 데이터 수신을 위해 직렬 디바이스 드라이버로 리눅스 커널 적재 포팅한다. GRS칩 구동을 위하여 안드로이드의 라이브러리 계층과 커널 드라이버 통신하는 하드웨어는 HAL 추상화 계층 프로토콜 프로그램을 개발하였다. GRS칩 구동을 하기 위하여 안드로이드 플랫폼에서 직렬 통신을 위한 HAL 계층에 대하여 라이브러리는 안드로이드 개발환경에서 제공하는 NDK를 사용하여 빌드 통하여 프로그램을 실행하였다. 그림 3과 같이 직렬 데이터 수신 모듈 소스 프로그램 구현은 아래와 같은 형태의 코드로 작성하였다.

```

    package com.koti.serial;

    import android.os.Parcel;
    import android.os.Parcelable;
    import android.util.Log;

    import java.io.IOException;
    import java.lang.String;

    public class SerialPrimitive implements Parcelable {
        private int mFd;

        public SerialPrimitive(int fd) {
            mFd = fd;
        }

        public int getFd() {
            return mFd;
        }

        public void setFd(int fd) {
            mFd = fd;
        }

        @Override
        public int describeContents() {
            return 0;
        }

        @Override
        public void writeToParcel(Parcel dest, int flags) {
            dest.writeInt(mFd);
        }

        public static final Parcelable.Creator<SerialPrimitive> CREATOR = new Creator<SerialPrimitive>() {
            @Override
            public SerialPrimitive createFromParcel(Parcel in) {
                return new SerialPrimitive(in.readInt());
            }

            @Override
            public SerialPrimitive[] newArray(int size) {
                return new SerialPrimitive[size];
            }
        };
    }

    public class SerialPrimitive {
        public SerialPrimitive(int fd) {
            mFd = fd;
        }

        public int getFd() {
            return mFd;
        }

        public void setFd(int fd) {
            mFd = fd;
        }

        @Override
        public int describeContents() {
            return 0;
        }

        @Override
        public void writeToParcel(Parcel dest, int flags) {
            dest.writeInt(mFd);
        }

        public static final Parcelable.Creator<SerialPrimitive> CREATOR = new Creator<SerialPrimitive>() {
            @Override
            public SerialPrimitive createFromParcel(Parcel in) {
                return new SerialPrimitive(in.readInt());
            }

            @Override
            public SerialPrimitive[] newArray(int size) {
                return new SerialPrimitive[size];
            }
        };
    }
    
```

그림 3. 직렬 데이터 수신 모듈 소스 프로그램 구현
 Fig. 3 Implementation of serial data receiving module source program

4.2 호스트와 데이터 연동을 위한 호스트 인터페이스 프로토콜 구현

제안한 안드로이드 플랫폼 GRS칩 구동 JNI 코드로 호스트 단말과 센서 간 직렬 통신을 전송하기 위해서는 안드로이드 애플리케이션에 통해서 동작 센서 디바이스와 데이터 교환 위해 명령어 세트와 패킷 구조로 정의한다.

(1) 제안한 호스트 안드로이드 단말과 센서 간 직렬 통신 구현

안드로이드의 최상위 애플리케이션이 GRS칩 구동 JNI 코드로 라이브러리를 통해 직렬 통신을 수행한다. 안드로이드의 애플리케이션 프레임워크를 통해 직렬 통신용 클래스가 정의한다. 해당 코드는 아래와 같은 구조로 작성하고, 안드로이드 플랫폼 GRS칩 구동에서 사용한 JNI 구현은 본 논문에서는 안드로이드 기반 모듈 구현으로 안드로이드 플랫폼 GRS칩 구동에서 사용한 JNI 코드 구현은 그림 4와 같다.

```

public class deviceJNI {
    static {
        System.loadLibrary("12cdev");
    }
    private native int openDev();
    private native int closeDev();
    private native byte readDev();

    // JNI 코드에서 GRS에 인식되는 최소채널의 각인의 EVENT를 정의
    public final int LEFT_EVENT = 0x01;
    public final int RIGHT_EVENT = 0x02;
    public final int UP_EVENT = 0x03;
    public final int DOWN_EVENT = 0x04;

    // 좌측 센서의 FIELD를 정의
    private static final byte LEFT_FIELD = (byte)0x01;
    private static final byte RIGHT_FIELD = (byte)0x02;
    private static final byte UP_FIELD = (byte)0x03;
    private static final byte DOWN_FIELD = (byte)0x04;
    private static final byte VALID_FIELD = (byte)0x05;
    private static final byte DOWN_UP_FIELD = (byte)0x06;
    private static final byte DOWN_UP_FIELD = (byte)0x06;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    Message msg;
    EventHandler handler;
    Thread thread;
    byte vector;

    public deviceJNI(EventHandler aHandler) {
        handler = aHandler;
        thread = new Thread(new Runnable() {
            public void run() {
                try {
                    while(true) {
                        Thread.sleep(10);
                    }
                } catch (InterruptedException e) {}
            }
        });
        // vector에서 GRS의 값을 읽음
        vector = readDev();
        vector = (byte)vector;

    // Handler 함수를 이용한 좌측에서 FIELD02의 범위가 일치하고, 조건이 일치하면 출력 가능하게 함
        if ((vector & VALID_FIELD) == VALID_FIELD) {
            if ((vector & DOWN_UP_FIELD) == DOWN_UP_FIELD) {
                ((vector & RIGHT_LEFT_FIELD) == RIGHT_LEFT_FIELD)
            }
            if ((vector & 0x7) == 0x7) {
                ((vector & 0x7) == 0x7)
            }
            ((vector & 0x7) == 0x7)
        }
    }
}
    
```

그림 4. 안드로이드 플랫폼 GRS칩 구동에서 사용한 JNI 코드 구현

Fig. 4 Implementation of JNI code used in Android platform GRS chip operation

(2) 안드로이드 스마트폰에서 사용한 GRS 구동 UART CLASS 코드 구현

안드로이드 기반 모듈 구현으로 제안한 안드로이드 스마트폰에서 사용한 GRS 구동 UART CLASS 코드 구현은 그림 5와 같다.

```

public class deviceJNI {
    static {
        System.loadLibrary("12cdev");
    }
    private native int openDev();
    private native int closeDev();
    private native byte readDev();

    // JNI 코드에서 GRS에 인식되는 최소채널의 각인의 EVENT를 정의
    public final int LEFT_EVENT = 0x01;
    public final int RIGHT_EVENT = 0x02;
    public final int UP_EVENT = 0x03;
    public final int DOWN_EVENT = 0x04;

    // 좌측 센서의 FIELD를 정의
    private static final byte LEFT_FIELD = (byte)0x01;
    private static final byte RIGHT_FIELD = (byte)0x02;
    private static final byte UP_FIELD = (byte)0x03;
    private static final byte DOWN_FIELD = (byte)0x04;
    private static final byte VALID_FIELD = (byte)0x05;
    private static final byte DOWN_UP_FIELD = (byte)0x06;
    private static final byte DOWN_UP_FIELD = (byte)0x06;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    private static final byte RIGHT_LEFT_FIELD = (byte)0x07;
    Message msg;
    EventHandler handler;
    Thread thread;
    byte vector;

    public deviceJNI(EventHandler aHandler) {
        handler = aHandler;
        thread = new Thread(new Runnable() {
            public void run() {
                try {
                    while(true) {
                        Thread.sleep(10);
                    }
                } catch (InterruptedException e) {}
            }
        });
        // vector에서 GRS의 값을 읽음
        vector = readDev();
        vector = (byte)vector;

    // Handler 함수를 이용한 좌측에서 FIELD02의 범위가 일치하고, 조건이 일치하면 출력 가능하게 함
        if ((vector & VALID_FIELD) == VALID_FIELD) {
            if ((vector & DOWN_UP_FIELD) == DOWN_UP_FIELD) {
                ((vector & RIGHT_LEFT_FIELD) == RIGHT_LEFT_FIELD)
            }
            if ((vector & 0x7) == 0x7) {
                ((vector & 0x7) == 0x7)
            }
            ((vector & 0x7) == 0x7)
        }
    }
}
    
```

그림 5. 안드로이드 스마트폰에서 사용한 GRS 구동 UART CLASS 코드 구현

Fig. 5 Implementation of GRS driven UART CLASS code used in Android smartphone

본 논문에서는 안드로이드 GRS칩 구동 JNI 코드 기반 모듈 구현으로 안드로이드 운영체제의 애플리케이션을 통하여 무선 통신 디바이스에서 데이터를 수신으로 직렬 통신 드라이버, 라이브러리, 프레임워크의 클래스는 다음과 같은 ART Setting 안드로이드 애플리케이션을 이용해서 센서의 데이터를 확인할 수 있다. 그림 6과같이 안드로이드 애플리케이션 터미널을 통해 무선 통신 모듈은 마스터와 스마트폰을 1대 1로 직렬 통신한다. 데이터 마스터 부팅 네트워크 통하여 동기신호 비콘(beacon) 송·수신한다.

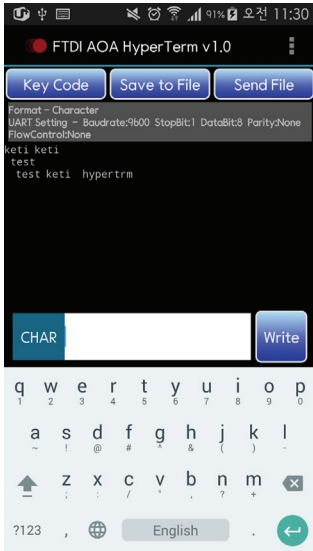


그림 6. 안드로이드 터미널 애플리케이션 구현
Fig. 6 Android terminal application implementation

그림 7과 같이 안드로이드 디바이스 정보는 안드로이드 운영체제 상에서 직렬 통신을 구현한다. 본 논문에는 범용 안드로이드 디바이스 버전 5.0.1의 갤럭시 노트4를 안드로이드 기반 디바이스 정보 활용하여 실험하였다.

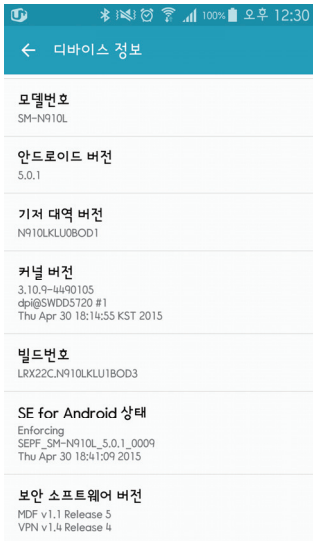


그림 7. 안드로이드 기반 디바이스 정보
Fig. 7 Android based device information

V. 결론

본 논문에서는 GRS칩 구동 JNI 코드 응용 SW 프로그램 설계하여 안드로이드 플랫폼 기반 동작 제스처 프레임워크 모듈 설계 구현하였다. 안드로이드 기반 모듈 설계, 안드로이드 기반 모듈 구현, 안드로이드 기반 기능 모듈 구현 설계로 응용애플리케이션 기반 네트워크 지원 API 기술의 직렬 데이터 수신 모듈 설계를 하였다. 안드로이드 기반 플랫폼 GRS칩 구동 JNI 코드 설계는 JNI 코드 구동을 하기 위해서 JNI 코드는 안드로이드에서의 JNI 장치 드라이버 제어하였다. GRS칩 구동 안드로이드 기반 플랫폼은 동작 제스처 값들의 필드를 정의한다. 이동체 동작 제스처를 인식하는 방법은 동작 제스처에 대한 벡터 값을 추출하는 동작과 추출된 벡터 값에 벡터의 패턴을 생성하는 동작과 저장된 데이터 벡터의 패턴과 새로이 생성된 벡터의 패턴을 비교하는 동작과 비교 동작에 기반을 통하여 이동체 동작 제스처의 유형을 결정한다. 안드로이드 플랫폼 기반 동작 제스처 터치 판단 프레임워크 안정화 기반 응용 SW 개발을 통하여 안드로이드 운영체제의 애플리케이션 전송되는 무선 통신 디바이스로부터 수신된 데이터를 직렬 통신 드라이버 및 라이브러리 프레임워크의 클래스와 같이 안드로이드 애플리케이션을 통해 센서의 데이터 정보를 확인하였다.

References

- [1] S. Youn and K. Seok, "Design and implementation of TDMA-based wireless IP video transmission system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 17, no. 6, 2022, pp. 1025-1032.
- [2] B. Jung and K. Seok, "A study on 2.4/5GHz dual-band RF design technology," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 18, no. 2, 2023, pp. 259-268.
- [3] S. Youn, M. Kim, and K. Seok, "Inter-module interworking evaluation of TDMA-based wireless IP video transmission system," *J. of the Korea Institute of Electronic Communication Sciences*, vol.

- 18, no. 1, 2023, pp. 1-9.
- [4] J. Cho and K. Jeong, "Remote Controlled Robots Used for the Mitigation of the Fukushima Nuclear Power Plant Accident," *J. of Institute of Control, Robotics and Systems*, vol. 2011, no. 12, 2011, pp. 148-151.
- [5] X. Hao and C. Kim, "Design and Implementation of a Smart Home Cloud Control System Using Bridge based on IoT," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 12, no. 5, 2017, pp. 865-872.
- [6] S. Park, W. Choi, B. Chung, J. N. Kim, and J. M. Kim, "The Study on the Cyber Security Requirements of Cyber-Physical Systems for Cyber Security Frameworks," *J. of Institute of Embedded Engineering of Korea*, vol. 7, no. 5, 2012, pp. 255-265.
- [7] Y. Shin and J. Ryu, "Study on Adoption of Suitable Encryption Scheme According to Data Properties on MySQL Database," *Proc. Korea Computer Congress 2010*, Jeju island, South Korea, June 2010.
- [8] R. D. Silverman, "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths," *Bulletin of RSA Laboratories*, no. 13, Apr. 2000.
- [9] B. S. Dhillon, *Engineering Safety*. Singapore: World Scientific, 2003.
- [10] C. Kong and L. Semyeong, "A Study of Inverse Modeling from Micro Gas Turbine Experimental Test Data," *J. of Korean Society of Propulsion Engineers*, vol. 13, no. 6, 2009, pp. 1-7.
- [11] A. Selby and C. Mitchel, "Algorithms for Software Implementations of RSA," *IEE Proc. E - Computer and Digital Techniques*, vol. 136, no. 3, 1989, pp. 166-170.
- [12] T. Fujita, K. Kogiso, K. Sawada, and S. Shin, "Security Enhancement of Networked Control Systems Using RSA Public-key Cryptosystem," *Proc. 10th Asian Control Conf.*, Kota Kinabalu, Malaysia, May 2015.
- [13] W. Choi and K. Seok, "Survey on ways to improve the system in preparation for changes in the radio management system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 13, no. 6, 2018, pp. 1145-1154.
- [14] J. Yang, K. Seok, and H. Sin, "Technological and Social Significance of the Revision of the Radio Law," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 14, no. 4, 2019, pp. 627-636.
- [15] C. Yeon and K. Seok, "A study on radio wave management regulations in the United States to improve the domestic radio wave management system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 15, no. 3, 2020, pp. 379-388.

저자 소개



전영남(Yeong-Nam Jeon)

2004년 조선대학교 경영학과 졸업(경영학사)

2010년 호남대학교 대학원 컴퓨터공학과 졸업(공학석사)

2023년 동강대학교 AI융합기계과 교수

※ 관심분야 : 안드로이드 통신, 데이터 통신