리모트 컨트롤 앱세서리 설계 및 구현

김성훈, 심혜경, 이종혁
경성대학교

Remote control Appcessories design and implementation

Seong-Hun Kim, Hye-kyeong Sim, Jong-hyeok Lee
Kyung-Sung University
E-mail : jhlee@ks.ac.kr

요 약

최근 스마트폰의 발달로 우리의 생활은 더욱 편리해 졌으며, 이를 이용하려는 이용자 역시 급격하게 증가하였다. 이에 다양한 어플리케이션들이 출시되고 있지만, 스마트폰에 내장된 기능들로는 한계점이 있어, 모바일 단말기와 액세서리를 결합한 앱세서리가 주목받고 있다. 본 논문에서는 블루투스 통신을 이용한 리모트 컨트롤 앱세서리를 설계 및 구현하고, 이를 유아용 전동차 시스템에 적용시켜 모바일 단말기와 시스템 하드웨어의 강인성이나 사용의 편리성에 대해 검증을 하고자 한다.

ABSTRACT

Our lives are more convenient by development of smart phones recently, and users want to use it also increased rapidly. But being released by a variety of such applications, the smart phones features include built-in limitations, so appcessories that combines mobile handsets and accessory is attracting attention. In this paper, we design and implement Remote Control Appcessories using Bluetooth communication, and by applying it to the infant car system, we verify the robustness of mobile terminals and system hardware, and ease of use .

키워드

안드로이드, 스마트폰, 앱세서리, 리모트 콘트롤

Ⅰ. 서 론

최근 스마트폰의 발달로 다양한 어플리케이션(Application, 이하 줄여서 '앱')들이 출현하고 있다. 대표적으로 모바일 뱅킹, 영화나 연극 좌석 예매, GPS를 이용한 내비게이션, 증강 현실 기술을 적용한 건물 찾기 등 여러 앱이 있는데, 이러한 앱들로 편리성이 증가하였고, 아울러 스마트폰의 이용자 역시 급격하게 증가하였다.

문태양 등은 스마트폰의 블루투스 모듈과 기울기 센서간의 데이터를 처리하는 어플리케이션을 구현하였으며 이를 기반으로 GPS를 이용한 위치 추적 어플리케이션을 구현하는 것을 목표로 하고 있다.[1] 아론 링 치 이 등은 스마트 용접기 시스템을 위한 하이브리드 앱 소프트웨어 구조를 설계 개발하였으며,[2] 송성호 등은 스마트폰의 여러 가지 센서들을 이용하여 사물놀이 어플리케이션을 구현하였다.[3] 오승준 등은 자전거 커뮤니티 어플리케이션 설계 및 구현하여 기존의 어플리케이션에서 제공했던 경로, 운동량, 속도의 정보 외에도 사용자간의 경로와 전거정보 공유, 분실된 자전거를 찾을 수 있도록 도움이 되는 기능을 추가하여 기존의 어플리케이션보다 나은 어플리케이션을 제안한다.[4] 조영호 등은 복잡도가 높은 건물 내에 산재되어 있는 비상등, 소화기 및 스프링클러를 유비쿼터스 센서 네트워크 기반의 자율무선통신방식을 이용하여 현재 상태 모니터링, 작동 유무를 관리 할 수 있는 지능형 소방방재 시스템에서의 관리 어플리케이션을 구현하였다.[5]

스마트폰 출시 초기의 이용자 연령층이 20~30대에 집중되어 있었다면, 최근에는 10~60대까지 이용자 연령층이 매우 다양해 졌다. 이렇게 이용자 연령층이 다양해지면서 사용자들은 더욱 많은 기능을 요구하고 있으나, 단말기에서 지원하는 기본적인 기능만으로는 많은 한계가 있어 구현하기 힘든 점이 많다. 이러한 부분을 보완하기 위해 IT 시장에서는 앱세서리(Appcessories)라는 것을 내

놓고 있다.

현재 시장에 나와 있는 앱세서리의 종류를 살펴보면 스마트폰 사진인화기 'LG 포켓포토'와 스마트폰 프로젝터 'Smart i Beam'[6] 그리고 신용카드 단말기, 바코드 스캐너, 건강관리를 할 수 있는 스마트폰 손목밴드 등으로 다양하다.[7] 앱세서리는 사용법도 간단하여, 스마트폰에 장착 또는 통신으로 연결하여 다운로드 받은 어플리케이션을 이용해 기능을 확장해 이용하기만 하면 된다. 미국 아이디어상품 쇼핑몰에는[8] 원격조종하며 전방에 달린 카메라로 영상을 스트리밍하거나 녹화 및 사진촬영이 가능한 'Rover 2.0 App-Controlled Wireless Spy Tank', 4개의 팬으로 구성된 헬리콥터인 'The future of RC helicopters', 공의 무게 중심을 옮기는 추를 이용한 'Spehro® App-Controlled Wireless Robotic Ball', 손바닥 만한 미니카를 블루투스로 원격조종하는 'CarBots Micro RC Cars' 등이 있다.

본 연구에서는 리모트 컨트롤 앱세서리를 위한 요구분석을 한 후 이를 기반으로 리모트 컨트롤 앱세서리를 설계 및 구현하고, 이를 블루투스 통신과 원격제어를 활용한 시스템에 적용시켜 시스템의 강인성이나 사용의 편리성에 대해 검증을 하고자 한다.

II. 관련 연구

2.1 앱세서리 정의 및 특징

앱세서리란 모바일 앱(app) 형태의 서비스와 일종의 액세서리(Accessories)인 하드웨어가 결합된 방식으로 작동하는 모바일 액세서리를 총칭한다. 이것은 앱과 하드웨어의 결합을 통해 모바일 단말기가 제공하지 못하는 기능을 보완해 주고, 서비스 인터페이스를 확장하는 역할을 한다는 점에서 단순히 모바일 단말기를 보조하는 역할에 머물렀던 기존 액세서리와 차별점을 지닌다. 모바일 업계에서는 모바일 단말기가 갖추지 못한 특정 기능을 제공하는 액세서리와 이와 연계된 모바일 앱을 결합해 작동하는 앱세서리 시장이 두드러진 성장세를 기록하며 새로운 니치마켓으로 주목 받고 있다. 현재 앱세서리 업계는 헤드셋, 스피커, 프린터 등 컴퓨팅 및 커뮤니케이션 주변기기, 리모컨으로 조종하는 탁상용 미니카 등 완구, 디지털 카메라, 리모컨과 같은 소비자 가전, 시계, 혈압 등을 측정하는 신체 모니터링 기기, 로봇, 악기, 자동차, 조명 및 전열 기구, 보안용 단말 등 광범위한 유형의 하드웨어를 모바일 앱과 연계해 제공하고 있는 것으로 파악되고 있다. 그 중에서도 블루투스 통신 기반의 앱세서리가 주목받고 있다. 과거 블루투스 통신은 전송속도가 느려 대용량 데이터를 전송하기에 부적합하였으나, 현재 24Mbps의 속도를 유지하면서 수년간 사용할 수 있는 저전력 블루투스 v4.0까지 개발되었다.[9]

2.2 블루투스 통신

기기 목록 검색 및 연결, 상호 데이터 전달 등은 블루투스 기반으로 통신하고 페어링 요청 및 수신을 통해 등록기기 정보를 확인한다. 이를 그림 1에 나타내었다.

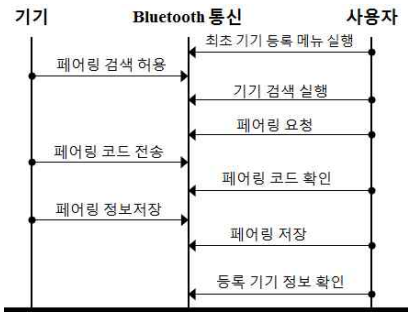


그림 1. 블루투스 통신

안드로이드에서 블루투스 통신을 하기 위해서 우선 블루투스 통신 모듈인 H/W를 사용하기 위해 AndroidManifest.xml에 허가를 요청한다. 그리고 블루투스 어댑터 객체를 통해 API객체를 호출하며, 블루투스 API를 통해 연결할 디바이스를 검색한다. 디바이스 검색이 시작되면, 관련된 디바이스들의 정보를 브로드캐스트를 통해 받는다. 받아온 정보들 중 해당 액션을 잡아주기 위해 onCreate()메소드에서 Reciver를 등록하고, 그에 해당하는 값을 등록한다. 리시버에 등록되었다면 디바이스 검색에 대해 성공 한 것이다. 연결할 디바이스가 검색되면, 검색을 중지하여 메모리 소모를 줄인다. 연결을 위한 검색된 디바이스의 MAC Address를 통해 BluetoothDevice를 생성한다. Android Bluetooth 구조에 따르면 Rfcomm Socket을 생성한다. 블루투스도 통신 모듈이기에, 이에 해당하는 소켓을 생성한다. 그리고 생성된 소켓에 connect() 메소드를 통해 디바이스를 연결한다. 여기서 보여지는 UUID는 연결하고자 하는 모듈통신이 가능한 UUID를 사용한다.[10]

2.3 통신 프로토콜

전동차와 App간의 통신을 할 때 데이터 전송이 제대로 되었는지를 확인하기 위한 방법으로 오류 검출 방식 중에 Check Sum을 이용하였다. 이것은 1번째 데이터, 2번째 데이터, 3번째 데이터를 모두 합을 4번째 데이터로 전송한다. 설정, 진단 요청은 App에서 차량으로 신호를 보내는 경우이고, 경고 메시지(과부하 상태, 제어기 내부 온도 상승)은 차량에서 App으로 신호를 보내는 경우이다. Check Sum 값이 일치 할 경우 일치 신호를 전송하고, 일치하지 않을 경우 Data 재전송 요구 신호를 전송한다. 이를 그림 2에 나타내었다.

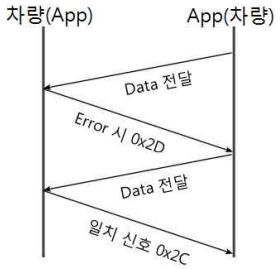


그림 2. Check Sum Error시 Data 재전송 요구

III. 설계

3.1 요구사항 분석

모니터링에서는 전동차의 제어기로부터 블루투스 무선데이터 통신을 통해 전송된 차량의 주행속도, 주행거리, 위험주행상황, 배터리 잔량, 자동안전제어 운전 진행 상황 등을 사용자가 스마트폰 디스플레이 화면 및 음성안내를 통해 모니터링 할 수 있도록 하며, 원격 운전 제어에서는 차량과의 양방향 무선데이터 통신을 통해 차량의 가속, 감속, 제동, 방향 전환 등을 원격 조종이 가능하도록 하며, 운전 모드 설정에서는 무선데이터 통신을 통해 차량의 최고속도, 주행모드, 조향감도, 제동모드, 수동/원격운전 전환 등 차량에 설치된 다양한 모드를 원격으로 설정할 수 있도록 하며, UI는 세로 방향으로 표시하며, 상단에 메뉴를 두도록 하고, 하단부에 현재 동작하는 내용을 표시하며, 좌우 혹은 속도 제어는 드래그 혹은 자이로 인터페이스를 선택하여 조절할 수 있도록 한다. 표시부는 하단부에 기존 자동차 패널 이미지를 두고, 현재 주행속도, 잔여주행가능거리, 배터리 잔량, 현재 설정된 모드 표시, 고장상태, 비상상태 등을 표시하도록 한다.

또한 블루투스 무선데이터 통신을 사용함으로써 기존 공용 대역의 주파수를 사용한 Radio Control 방식에서 발생하는 주파수 혼선 문제를 근본적으로 차단할 수 있도록 한다. 그리고 조작중에 전화가 올 경우 임의로 처리할 수 있도록 하기 위하여 루팅을 하게 될 경우 폰의 보안성이 많이 떨어지게 되어서 이용하는데 많은 제약이 있으므로 이의 처리는 당분간 보류하도록 한다.

3.2 리모트 콘트롤 앱세서리 설계

유아동용 전동차를 원격 조종할 수 있는 앱을 개발하기 위하여, 요구사항 분석을 토대로 전동차의 상태를 파악 할 수 있는 모니터링 기능과 블루투스 통신을 이용한 전동차 원격 운전 제어기능, 여러 가지 방법으로 작동 할 수 있게 하는 운전 모드 설정 등의 기능을 지원하도록 앱세서리를 설계하였으며 이의 DFD(Data Flow Diagram)를 그림 3에 나타내었다.

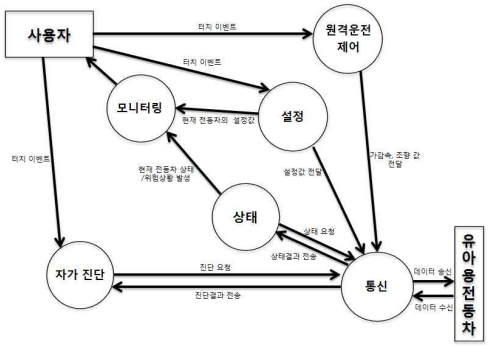


그림 3. App의 Data flow diagram

IV. 구현 결과

4.1 메인화면

전동차의 전·후진과 좌우조작은 드래그 조작으로 하되, 각각의 조작에 5단계를 두어 강도를 조절한다. 그리고 블루투스 연결 상태와 전동차 상태에 문제점이 있는 경우에 경고등을 표시하는 부분이 있으며 가운데 상단 쪽에는 제한속도와 운행모드, 수동/원격 조종 상태를 표시하는 창이 있고, 좌우에는 현재 속도와 배터리 잔량 표시부가 있다. 그리고 가운데 하단에는 비상 정지 버튼이 있다. 왼쪽 상단에는 부가 메뉴를 불러오는 버튼으로 구성하였다. 이를 그림 4에 나타내었다.

그림 4. 메인화면

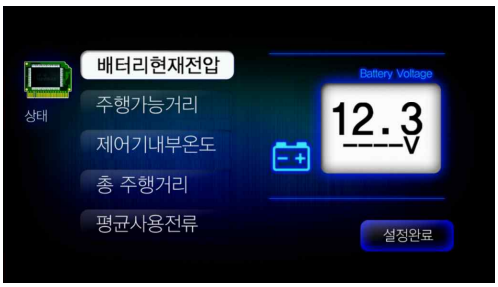
4.2 메뉴화면

전동차의 운행에 관련된 제한속도, 운행모드, 원격/수동설정, 수신방식을 설정할 수 있는 설정메뉴, 전동차의 현재 상태를 확인할 수 있는 상태메뉴, 전동차의 고장여부를 진단할 수 있는 진단메뉴, 음악을 지원할 수 있는 MP3 메뉴 및 현장정보를 기록 저장 할 수 있는 다이어리 메뉴로 구성하였으며 이를 그림 5에 나타내었다.

그림 5. 메뉴 화면

4.2.1 상태

전동차의 현재 상태를 나타내주는 메뉴이다. 배터리 현재전압과 주행 가능 거리, 제어기 내부 온도, 총 주행 거리, 평균 사용 전류가 있으며, 단순히 확인만 가능하도록 하였으며, 배터리 현재 전압 화면과 소스코드를 그림 6에 나타내었다.

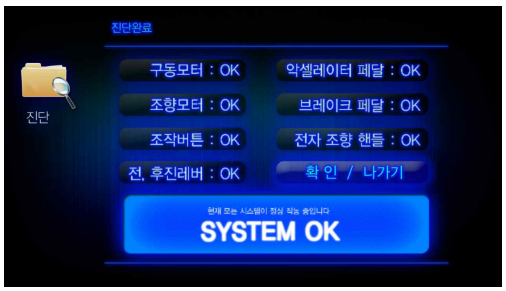



```
public void onClick(View v) {
    switch (v.getId()) {
    case R.id.state_ok://상태확인 버튼
        finish();
        break;
    case R.id.state_voltage://배터리 현재 전압
        ADC0_btn.setSelected(true);
        Source_Distance.setSelected(false);
        ADC8.setSelected(false);
        Total_Distance.setSelected(false);
        AVRG_Cur.setSelected(false);
        display.setImageResource(R.drawable.state_voltage_display);
        break;
```

그림 6. 배터리 현재 전압 화면과 소스 코드

4.2.2 진단

사용자의 지령에도 작동하지 않거나 비정상 작동하는 부위를 감지하고 이를 메모리에 저장된 사용자의 주행이력과 비교함으로써 차량의 고장 부위를 진단한다. 전동차 진단 결과 중 한 예로 모든 시스템이 정상인 경우의 결과와 소스코드를 그림 7에 나타내었다.




```
if(check_param.substring(0, 1).equals("0")){
    motor.setImageResource(R.drawable.motor_ok);
} else {
    motor.setImageResource(R.drawable.motor_error);
    count += 1;
    mc = " 구동모터";
}

if(check_param.substring(1, 2).equals("0")){
    johyang.setImageResource(R.drawable.johyang_ok);
} else {
    johyang.setImageResource(R.drawable.johyang_error);
    count += 1;
    jc = " 조향모터";
}
```

그림 7. 전동차 진단 결과 화면과 소스코드

V. 결 론

모바일 기기의 빠른 발전과 보급으로 인해 모바일 단말기의 사용용도가 다양해지고 있다. 이에 따라 모바일 단말기의 기능들만으로는 한정되어 있는 기술 및 시스템을 외부 디바이스를 활용하는 앱세서리가 등장하였고, 그것 또한 빠른 발전과 보급이 이루어지고 있다. 이를 바탕으로 아동용 전동차를 운전이 미숙한 아이들을 위해 원격 조종이 가능한 앱으로 언제 어디서든 리모컨을 따로 휴대하지 않고도 아이들의 전동차를 원격 조종 할 수 있도록 시스템을 구현하였다. 그리고 원격 조종 뿐만 아니라 자동차의 상태 까지 파악 할 수 있도록 모니터링 시스템을 구축하였고, 전동차 내부 진단까지 받아와 차량으로 직접 확인하지 않고도 전동차의 고장 유무를 파악 할 수 있도록 구현하였다.

참고문헌

[1] 문태양, 이호상, 진민수, 김윤호, 강희조, "기울기 센서와 블루투스 모듈을 이용한 iLog 어플리케이션 구현," Sustainable ICT Conference 2012, pp. 673-676, 2012.

[2] 아론 링 치 이, 강대기, "스마트 용접기 시스템을 위한 하이브리드 앱 소프트웨어 구조 설계," 2012 한국컴퓨터종합학술대회 논문집, Vol.39, No.1(B), pp. 48-50, 2012.

[3] 송성호, 김현기, "스마트폰을 이용한 사물놀이 어플리케이션 구현," 2010년도 한국멀티미디어학회 추계학술발표대회 논문집, 제13권, 2호, pp.365-368, 2010.

[4] 오승준, 윤근, 이승철, 김석훈, 조진성, "Bike-Friend : 자전거 커뮤니티 어플리케이션 설계 및 구현," 2012 한국컴퓨터종합학술대회 논문집, Vol. 39, No. 1(D), pp. 158-160, 2012.

[5] 조영호, 여상수, 강희조, "안드로이드 기반의 지능형 소방방재 관리 어플리케이션 구현," 2011 한국정보기술학회 하계학술대회 논문집, pp. 208-210, 2011.

[6] http://news.kbs.co.kr/news/NewsView.do?SEARCH_NEWS_CODE=2599930

[7] http://www.zdnet.co.kr/news/news_view.asp?artice_id=20130413103049&type=xml

[8] http://www.brookstone.com/

[9] blog.naver.com/strabase/120175537297

[10] http://developer.android.com/guide/topics/connectivity/bluetooth.html