

# 아두이노를 이용한 위치정보 기반 안전한 자전거 애플리케이션

김동건, 이호세, 김소영, 김태우, 이형우<sup>1</sup>

<sup>1</sup>한신대학교 컴퓨터공학부

## LBS/GPS based Bicycle Safety Application with Arduino

**Dong-Gun Kim, Ho-Se Lee, So-Young Kim, Tae-Woo Kim, Hyung-Woo Lee<sup>1</sup>**

<sup>1</sup>Division of Computer Engineering, Hanshin University

요 약 최근 자전거 이용자가 천만 명을 돌파하면서 자전거 라이딩은 일상생활의 요소로 자리 잡고 있다. 하지만, 이용자가 늘어날수록 사고의 위험에 무방비로 노출되면서 보다 안전하게 라이딩을 할 수 있는 예방 대책 수립이 절실하다. 이에 본 논문에서는 GPS 정보를 통하여 획득된 위치정보를 활용하여 사용자에게 주행 상황에 맞는 다양한 서비스를 제공할 수 있는 애플리케이션을 개발하였다. 이를 위해 안드로이드와 아두이노 기반 디바이스간 블루투스 송수신을 통해 자전거 전용 내비게이션 기능을 제공하여 안전성을 향상시킬 수 있었다. 향후 다양한 형태의 아두이노 센서로 확장할 경우 개인 맞춤형 모바일 IoT 서비스로 발전시킬 수 있을 것이라 기대한다.

주제어 : 아두이노, 위치정보, IoT, 모바일 어플리케이션, 안전성

**Abstract** Bicycle users were surpassed a million recently as bicycle riding will be a key element of daily life. As users increase, we can expect that the user is more exposed to the risk of an unexpected accident. Therefore, we need to establish several preventive measures for a safe riding environment. In this paper, we developed mobile application by using the acquired GPS-based location information that provide safety services for the bike riding as a LBS service. In detail, we have been able to provide a bicycle safety navigation service by using the bluetooth transmission and reception method between Android and Arduino-based devices. As a future work, the safety could be improved if we expand various forms of Arduino sensors. As a result, a service developed in this study will be able to converge ahead to connect into several personalized mobile IoT services.

**Key Words** : Arduino, LBS/GPS, IoT, Mobile App, Safety Service

### 1. 서론

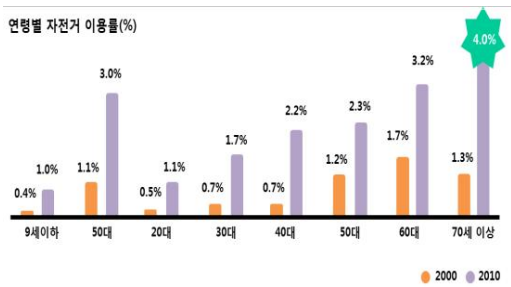
현대인들의 바쁜 생활 속에서 대중교통을 이용하는 인구가 증가하면서 교통 혼잡을 벗어나기 위해 대체수단으로 자전거를 이용하고, 삶의 질이 높이기 위해 여가생활을 충족시키려는 자전거 인구도 점차 증가하는 추세이

다. <그림 1.1>과같이 자전거를 이용하는 인구가 증가하면서 이제 자전거는 과거의 단순한 이동 수단에서 벗어나 생활 속의 일부로 자리 잡아가고 있다. 이러한 현상들로 관련된 애플리케이션이 등장하면서 좀 더 쾌적한 라이딩을 추구하기 위한 목적으로 많은 응용 프로그램들이 쏟아져 나오고 있다.

본 논문은 한신대학교 2015년도 2학기 컴퓨터공학부 종합설계2 프로젝트 결과물입니다.

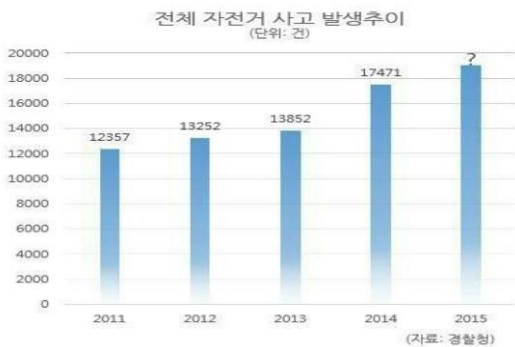
\*교신저자 : 이형우(hyungwoo8299@gmail.com)

접수일 : 2015년 12월 8일, 수정완료 : 2016년 2월 20일, 최종 게재 확정 : 2016년 3월 10일



[그림 1.1] 연령별 자전거 이용률

아래 <그림 1.2>와 같이 경찰청의 2011~2015년도 전체 자전거 사고 발생 추이에 따르면 2011년도 약 1만 2천 건의 자전거 사고가 지속적으로 증가하여 2015년도에는 약 2만 명에 이를 것이라 전망하고 있다. 이처럼 자전거 사고를 예방할 대안이 꼭 필요한 시점이다. 하지만 기존의 자전거 애플리케이션은 라이딩의 편의만 추구할 뿐, 안전에 관한 기능에 미흡한 것으로 나타났으며, 주요 기능들이 네이티브 앱에 한정되어 있어서 추가 확장이 어렵다는 문제점이 있다. 또한, 자전거 라이딩은 날씨에 큰 영향을 받기 때문에 사용자에게 기상 정보를 제공할 수 있는 기능이 포함되어 제공될 필요가 있다.



[그림 1.2] 연도별 자전거 사고 발생추이

이에 본 논문에서는 안드로이드와 아두이노의 블루투스 간 양방향 통신(two way-communication)을 이용하여 스마트 기기를 제어함으로써 기존에 안드로이드만을 이용한 한정적인 네이티브 기능의 한계를 극복하고자 한다. 본 연구를 통해 자전거 라이딩 앱의 활용성을 더욱 개선하여 사용자의 안전사고 예방에 기여할 수 있는 방법을 제시하였다. 아두이노와 모바일 디바이스 간의 통신은 블루투스 WPAN(Wireless Personal Area Network)을 이용하여 사용자들의 편의성과 효율성을 증

대하고자 하였다. 새로운 모바일 앱 구현을 통해 누구나 편리하게 목적지까지의 길 안내를 받을 수 있고 아두이노의 DOT MATRIX를 통해 내비게이션 정보를 제공함으로써 보다 직관적으로 기능 활용이 가능하다. 그리고 자신의 운동 기록을 통해 건강 유지를 할 수 있으며 기상 정보로 쾌적한 라이딩 환경을 제공할 수 있으며 결국에는 이를 통해 국민 생활 안전에 기여할 것으로 예상된다.

## 2. 관련 연구

### 2.1 LBS(Location Based Service) 및 안드로이드

위치기반 서비스(LBS: Location Based Service)는 아래 <그림 2.1>과 같이 사용자의 위치정보를 기반으로 한 서비스로 이동통신망이나 GPS를 통하여 획득된 위치정보를 활용하여 사용자에게 상황에 맞는 다양한 서비스를 제공하는 시스템이나 서비스를 의미한다. 최근에는 국내에서도 이동통신/스마트폰을 이용한 소셜 네트워킹 서비스에 관심이 높아지면서 LBS 서비스가 유망한 비즈니스 모델로 급부상하고 있다[1].

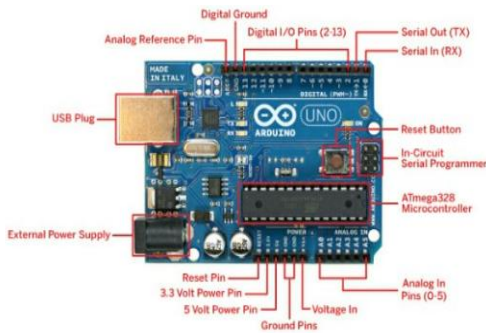


[그림 2.1] LBS : 위치기반 서비스

또한 구글에서 개발한 안드로이드(Android)은 휴대폰 운영 체제, 미들웨어, 응용프로그램을 한데 묶은 소프트웨어 플랫폼이다. 2007년 11월 세계 각국의 이동통신 관련 회사 연합체인 OHA(Open Handset Alliance)가 공개한 안드로이드는 리눅스(Linux) 2.6 Kernel을 기반으로 강력한 운영체제와 포괄적 라이브러리 세트, 풍부한 멀티미디어 사용자 인터페이스 및 스마트 디바이스 애플리케이션 등을 제공한다[2]. 또한 안드로이드 운영체제를 탑재한 스마트폰에서 GPS 정보를 수신할 수 있기 때문에 결국 모바일 단말을 이용하여 다양한 형태의 LBS 서비스를 구축할 수 있게 되었다.

## 2.2 블루투스(Bluetooth) 및 아두이노(Arduino)

컴퓨터, 프린터, 이동전화기 단말기, 개인 휴대 단말기 등 정보 통신 기기는 물론 각종 디지털 가전 제품을 우선 접속 장치 없이 무선으로 연결해주는 근거리 무선 네트워킹 기술로 블루투스(Bluetooth) 기술이 개발되었다. 블루투스 프로토콜을 이용할 경우 스마트 단말과 IoT 장비간 근거리에서 케이블 없이 네트워크 연결이 가능하게 되었다[3]. 따라서 블루투스 기능을 이용하여 스마트폰과 주변기기 등을 서로 연결/통신할 수 있는 서비스 환경이 구축되었다. 주변기기에는 아래 <그림 2.2>와 같이 아두이노(Arduino)를 탑재하여 다양한 형태의 센싱 기능을 스마트폰과 연계/접목할 수 있으며, 반대로 스마트폰에서 주변기기로 원하는 정보를 블루투스 프로토콜을 이용하여 전송할 수 있는 양방향 네트워크 환경이 구축되었다.



[그림 2.2] 아두이노 기반 주변기기 구성도

아두이노는 오픈 소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성된 보드와 관련 개발 도구 및 환경을 의미한다. 처음에 AVR(Automatic Voltage Regulator)을 기반으로 만들어졌으며, 아트 멜 AVR 계열의 보드가 현재 가장 많이 판매되고 있다. 이밖에도 ARM 계열의 Cortex-M0(Arduino Pro)과 Cortex-M3 (Arduino Due)를 이용한 제품도 존재한다.

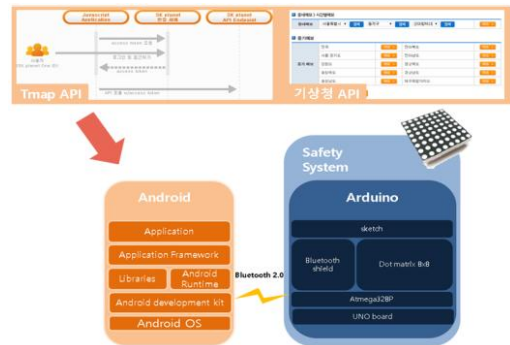
아두이노는 다수의 스위치나 센서로부터 값을 받아들이고, LED나 모터 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 제품 및 서비스를 구성할 수 있다[4]. 따라서 본 논문에서는 스마트폰으로부터 안전한 자전거 라이딩 서비스를 제공하기 위해 아두이노 기반 도트 매트릭스 출력 장치를 연결하여 자전거 진행 방향 표시 또는 문자 등을 제공하도록 구현하는 것을 목적으로 한다.

## 3. 아두이노를 이용한 위치정보 기반 안전한 라이딩 모바일 시스템 설계

### 3.1 시스템 구조 설계

본 연구에서 설계한 시스템의 전체 구조는 <그림 3.1>과 같이 안드로이드 기반 모바일 기기(Android)와 블루투스 기능을 탑재한 아두이노 펌웨어(Arduino)로 구성되어 있다. 안드로이드는 SK 플래닛에서 제공하는 Tmap API와 기상청 오픈 API를 이용하여 각각의 서버에서 제공하는 지도(map)과 날씨 정보를 실시간을 파싱할 수 있는 기능을 구현하였다.

본 연구에서 구현한 모바일 앱은 실시간 기상 정보나 실시간 운동 정보 등 사용자에게 자전거 라이딩 정보를 제공한다. 아두이노 기반 펌웨어는 안드로이드의 전환 정보를 블루투스를 통해 문자열을 수신 받은 다음, 문자열을 정수로 변환한 다음, 정수에 해당되는 조건문을 실행시켜서 도트 매트릭스 형태로 목적지에 대한 실시간 내비게이션 경로 정보를 표시하도록 구현하였다.



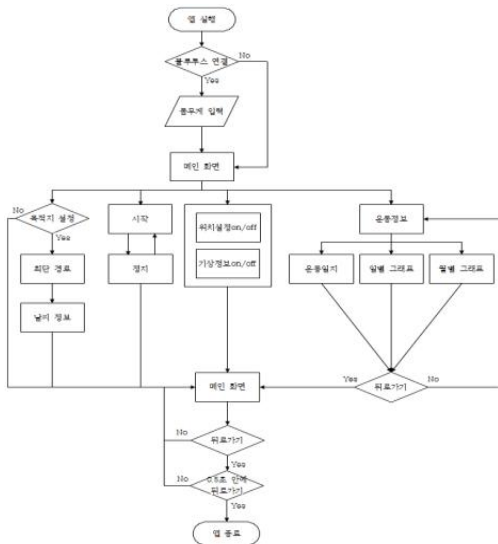
[그림 3.1] 시스템 구조도

### 3.2 시스템 흐름도

본 연구에서 구현한 시스템 내부의 상세 흐름도는 아래 <그림 3.2>와 같이 애플리케이션 실행 후 먼저 블루투스의 활성 여부를 확인한다. 블루투스 탐색을 누르면 5~10초 후에 연결 버튼이 생성되고 연결 버튼을 클릭하여 아두이노와 연동 과정을 수행한다. 블루투스가 활성화가 된다면 소모 칼로리나 아두이노를 통해 내비게이션 정보를 제공받을 수 있으며, 만일 활성화되지 않았을 경우에는 위와 같은 기능을 이용하지 못하도록 구현하였다. 메인 화면에서는 T-map 지도를 바탕으로 목적지 검색, 실시간 라이딩 정보 및 운동 일지 정보를 사용자에게 제

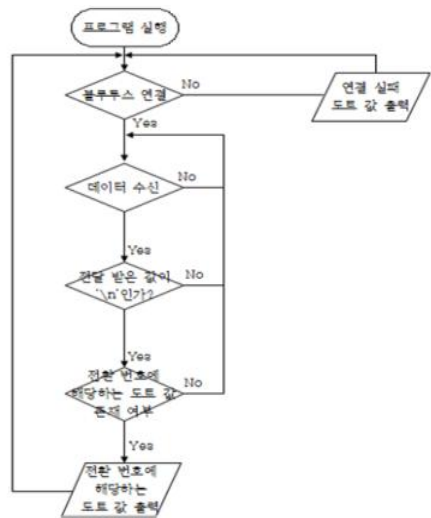
공한다.

위치정보 체크박스를 통해 사용자의 현재 위치를 지도의 중심점으로 이동시키는 기능을 설정하도록 구현하였고, 날씨 정보 체크 박스는 지도 상단을 가리는 레이아웃의 존재 여부를 표시하도록 구현하였다. 화면 하단의 toggle 버튼을 실행하면 현재 속력, 이동거리, 소모 칼로리 및 이동시간이 실시간으로 표시되도록 구현하였다. 지도를 통해 원하는 목적지 정보 아이콘 부분을 길게 누르고 있을 경우, 현재 위치에서 목적지까지의 최단 경로와 내비게이션 정보가 실시간으로 제공되며 남은 거리, 목적지 주소 및 기상 정보를 추가로 제공하도록 구현하였다. 이때 내비게이션의 전환 정보는 블루투스를 통해 아두이노 디바이스로 문자열 형태로 전송이 된다. 또한 운동 일지 버튼을 클릭하게 되면 날씨, 일별, 월별 tap 화면이 나타난다. 해당 요일별 운동 일지는 하루를 기준으로 평균 속도, 이동거리, 소모 칼로리, 이용 시간이 나타나며 저장된 정보를 모두 볼 수 있으며, 일별 정보는 일별로 총 이동거리와 총 소모 칼로리 양의 변화 추이를 achartengine 라이브러리를 사용하여 막대 그래프 형태로 표시되도록 구현하였다. 월별 정보 또한 월별 총 이동거리와 소모 칼로리 양을 막대그래프로 보여준다. back press handler를 통해 뒤로 가기를 제어해줌으로써 Activity간 이동 과정에서 안정성을 확보할 수 있도록 구현하였다.



[그림 3.2] 안드로이드 내부 작동 순서도

<그림 3.3>와 같이 아두이노 기반 주변기기 내부에서의 작동 흐름은 아래 <그림 3.3>과 같이 블루투스 연동 여부를 체크하는 모듈로 시작한다. 블루투스가 연동되지 않았을 경우에는 무표정 모양의 도트 매트릭스를 출력하여 사용자에게 아두이노 기반 주변기기와 연결되지 않았음을 표시되도록 구현하였다. 만일 블루투스 연동이 된다면 옷은 표정을 나타내어 연결됨을 표시한다. 반복 루프 과정을 수행하며 안드로이드 디바이스로부터 데이터 값을 수신하였는지를 주기적으로 체크를 한다. 안드로이드로 단말로부터 문자열을 수신 받았을 경우 수신된 문자열을 하나의 문자씩 버퍼에 저장시킨다. 안드로이드로부터 '\n'이라는 문자열 마지막 부분을 받으면 문자가 합쳐진 버퍼 값을 atoi를 통해 정수로 변환한 뒤, 전환 기호에 맞는 조건문을 실행시켜 내비게이션 경로 정보에 해당하는 도트 값을 출력되도록 구현하였다.



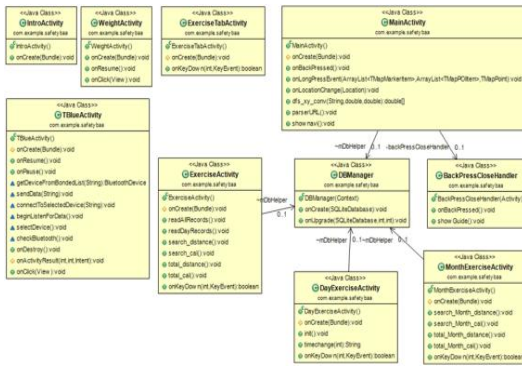
[그림 3.3] 아두이노 내부 작동 순서도

### 3.3 클래스 다이어그램 및 데이터베이스 구축

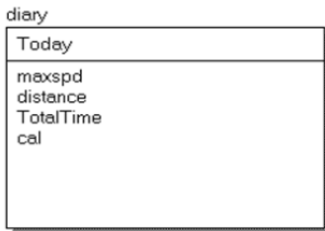
아래 <그림 3.4>은 안드로이드 단말 내부 소프트웨어 구현을 위한 클래스 다이어그램 구성을 보인다. 블루투스 클래스(TBlueActivity)와 메인화면(MainActivity)를 중심으로 각 기능별로 총 10 가지의 클래스로 나누어 구현하였다.

또한 SQLite 데이터베이스를 통해 'diary'라는 하나의 테이블로 구성하여 오늘 날짜를 primary key로 평균 속도, 총 이동거리, 총 이용 시간, 소모 칼로리 양 정보를 관리하도록 구현하였다. 또한 DB 정보 내에 운동 일지 정

보를 기록하도록 하였으며, 친구 사용자에게 다양한 기능을 제공할 수 있도록 확장성을 고려하였다.



[그림 3.4] 모바일 앱 내부 클래스 다이어그램



[그림 3.5] 앱 내부 데이터베이스 테이블

## 4. 위치정보 기반 안전한 자전거 모바일 앱 구현 결과

### 4.1 개발 환경

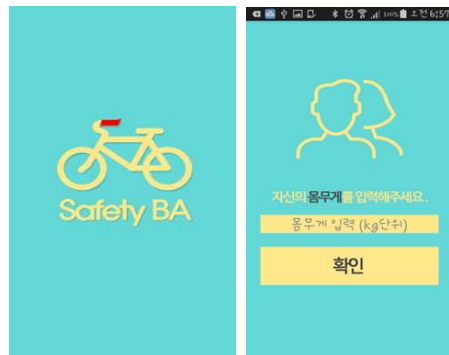
안드로이드와 아두이노 간의 연결을 위해 블루투스 기능을 이용하였고, 아두이노 스케치를 이용하여 펌웨어 컴파일 과정을 수행하였다. 안드로이드 DataBase의 효과적인 설계를 위해 Firefox에서 제공하는 SQLite Manager를 사용했고, 사용자 인터페이스는 안드로이드 기반 JAVA 개발 언어와 XML을 사용해 구현하였다. 그 외에 추가적인 개발 환경은 다음 [표 4.1]과 같이 통합 IDE eclipse 도구를 이용하여 블루투스 통신, LED 출력 및 센싱 기능을 제공하였다. 기상청 날씨 정보에 대한 파싱 기능을 수행하기 위해 XML 모듈을 탑재하였고, 안드로이드 단말을 이용하여 직접 실행 과정을 테스트하였다.

[표 4.1] 개발 환경

구분	항목	적용내역
OS	Android	에러 탐지 및 테스트
개발환경 (IDE)	eclipse	전반적인 프로그래밍
개발도구	Android studio	개발 셋팅 시간 단축
	SQLite manager	DB 설계
개발언어	XML, JAVA	안드로이드 앱 구성
라이브러리	achartengine	그래프 라이브러리
	T-map	지도 라이브러리
	Ledcontrol	LED 라이브러리
디바이스	Arduino UNO	센서 사용 컨트롤러
출력장치	Dot-Matrix	LED 출력
통신	HC-06	블루투스 통신
개발언어	C++	아두이노 프로그래밍

### 4.2 모바일 앱 구현 결과

디바이스에 설치된 애플리케이션을 실행할 경우 <그림 4.1>과 같이 초기 실행 및 설정 화면이 나타나며, 몸무게는 칼로리를 계산하기 위해 입력한다. 칼로리는 소비 칼로리계수에 의해 계산된다. 소모 칼로리 = 몸무게 x 평균 속도에 따른 칼로리 x 시간(분)으로 계산한다. 여기서는 onLocationChange()를 통해 1초당 소모되는 칼로리를 지속적으로 더함으로써 소모 칼로리 양을 구한다.



[그림 4.1] 앱 실행 및 설정 화면

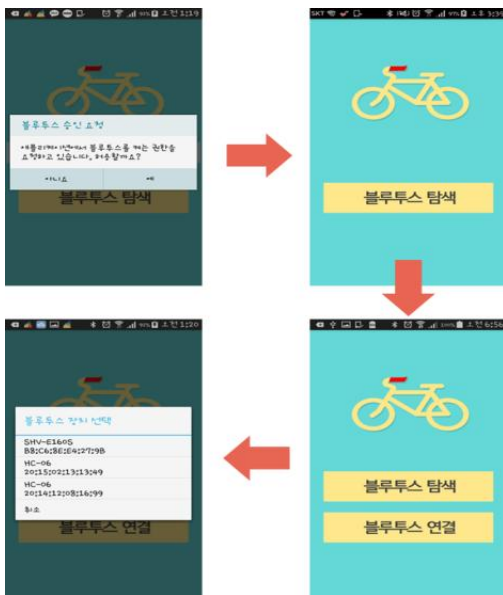
<그림 4.2>와 같이 블루투스 연결 단계를 표시하는 기능을 구현하였다. BluetoothAdapter 클래스를 이용하여 블루투스 지원이 가능한지 확인을 하고 활성화 과정을 수행한다. 블루투스 탐색 버튼을 클릭하면 BluetoothDevice를 통해 블루투스 장치의 이름이 주어졌을 때 해당 블루투스 장치 객체를 페어링 된 장치 목록에서 찾아낸다. 그리고 5~10초 뒤 블루투스 연결 버튼이



나타난다. 블루투스 연결을 클릭하면 블루투스 연동이 가능한 장치들이 이름과 MAC으로 나타나게 되고 사용자는 HC-06을 선택하여 연동을 한다. 블루투스를 작동시켜 아두이노와 송·수신하는 TBlueActivity는 앱이 종료되기 전까지 실행이 되어있어야 한다. 그렇기 때문에 안드로이드의 생명주기에 맞춰 registerReceiver를 onResume() 함수에 위치시킨다. onResume()은 현재 Activity가 background 상태로 상태 변환 시에 실행된다. 이로써 Activity 간 이동을 해도 블루투스 연결 상태는 지속되어 아두이노와 송·수신이 가능토록 구현하였다.



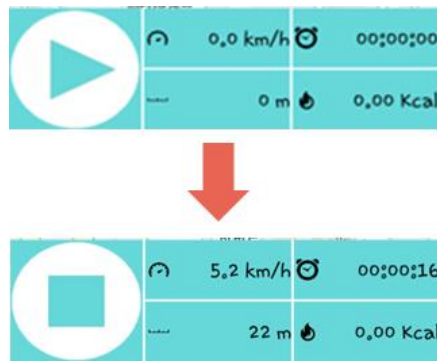
[그림 4.3] 메인화면 체크박스 on/off



[그림 4.2] 블루투스 탐색 및 연결 과정

<그림 4.3>은 메인 화면 상단의 체크박스이다. 본 논문에서 구현한 애플리케이션은 GPS를 통해 현재 위치의 좌표 값을 받아 지도에 표시한다. 첫 번째 위치 정보 체크박스는 활성화가 될 경우 Tmapview의 setCenterPoint()를 통해 현재 위치를 지도의 중앙값으로 이동시켜 실시간으로 처리를 한다. 비활성화될 경우에는 이를 무시함으로써 사용자가 지도를 자유롭게 이동시킬 수 있다. 두 번째 기상 정보 체크박스는 지도의 상단에 있는 레이아웃을 setVisible(View.GONE)을 이용하여 사용자의 편의에 따라 숨기거나 보이도록 구현하였다.

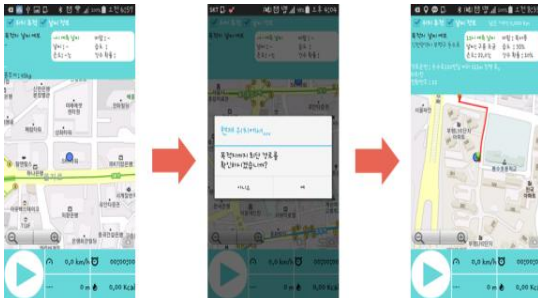
<그림 4.4>는 지도 하단의 실시간 정보 레이아웃이다. 시계 방향으로 현재 속도, 이용 시간, 소모 칼로리, 이동 거리를 나타낸다. 좌측의 Toggle 버튼의 시작 이미지를 클릭하게 되면 이미지가 바뀌면서 실시간으로 정보가 출력된다. 최소 2m 이동할 경우, onLocationChange()가 실행되며 알맞게 처리가 된다. 진행되고 있는 와중에 Toggle 버튼을 한 번 더 클릭하게 되면 실시간 정보들은 DB를 통해 현재 날짜를 primary key로 하나의 레코드 형태로 입력/저장된다. 만약 기본 키가 중복이 된다면 실시간 정보들이 추가로 업데이트 되도록 구현하였다.



[그림 4.4] 실시간 정보 on/off

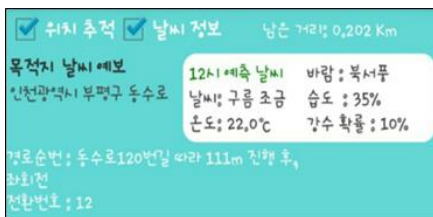
<그림 4.5>는 목적지 검색 기능을 나타낸다. 원하는 목적지를 logpress 하면 해당 위치에 마커가 표시된다. 현재 위치를 startpoint, 마커의 좌표를 endpoint로 정한다. Tmapdata의 findPathDataWithType를 통하여 BICYCLE\_PATH 타입과 startpoint, endpoint를 입력하면 자전거 타입 최단 경로가 나타난다. 이 최단 경로를 onLocationChange()에 위치시키면 지속적으로 불러들이면서 실시간으로 최단 경로를 구현한다. 목적지를 확인

하는 다이얼로그의 “예”를 누르면 최단 경로, 남은 거리, 목적지 주소, 기상정보, 내비게이션 정보가 나타난다.



[그림 4.5] 목적지 검색

<그림 4.5>의 상단 레이아웃을 상세히 살펴보면 아래 <그림 4.6>과 같다. 목적지 남은 거리는 최단 경로의 PolyLine의 거리를 getDistance로 받는다. 목적지 주소는 tmapdata의 reverseFeocoding을 통해 구한다. 내비게이션 정보는 geoJson을 통해 parsing 받는데, URL을 통해 startpoint와 endpoint의 좌표를 적절히 지정한 후 post 방식으로 요청을 한다. BufferedReader를 통해 한 줄씩 저장한다. “description”을 통해 길 안내 정보를 받고 “turnType”을 통해 전환 번호를 얻는다. 실시간 내비게이션이기 때문에 모든 값을 배열에 저장할 필요 없이 갱신될 때마다 첫 정보만을 가져와 출력시킨다. 날씨 정보는 기상청 RSS를 이용하여 동네예보 데이터를 xml 형식으로 parsing 한다. endpoint의 좌표 값을 기상청에서 요구하는 x, y 좌표값으로 변환한 뒤에 URL에 적용시킨다. 이 변환 과정은 기상청 내에서 제공해주는 오픈 소스를 활용한다. parser를 통해 “hour”, “temp”, “wfKor”, “wdKor”, “ws”, “reh”, “pop”를 받아 적절하게 표현시킨다.



[그림 4.6] 목적지 검색 후 상세 정보

<그림 4.7>은 운동 일지를 나타낸다. zoomcontrol 우측의 작은 삼각형 버튼을 클릭하게 되면 올라오는 애니메이션과 함께 날별, 일별, 월별 Tap이 나타난다. 날별 란

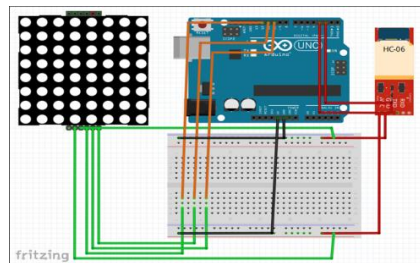
은 앞서 저장된 실시간 값들을 시각적으로 볼 수 있다. 상단의 화살표를 이용하여 앞 전의 레코드 값들을 볼 수 있다. 월별 란은 achartengine 라이브러리를 사용하여 총 이동거리, 총 소모 칼로리량을 일별 막대그래프로 보여 준다. 먼저 select를 통해 DB의 모든 레코드의 count를 구하고 primary key를 substring 하여 x축을 구성한다. 월별 란 또한 같은 achartengine 라이브러리를 이용한다. 먼저 Calendar.YEAR를 통하여 현재 년도를 구한다. 그 다음 일별 란과 같이 count를 구한 뒤, “Like %”를 통해 현재 구해진 년도와 1~12까지 월 값을 집어넣어 존재하는 이동거리 혹은 소모 칼로리 양의 SUM을 구하여 막대 그래프로 나타낸다.



[그림 4.7] 운동 일지

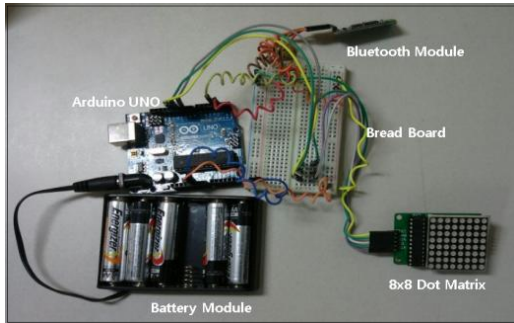
### 4.3 Hardware 구현

<그림 4.8>은 fritzing을 이용하여 아두이노의 구성도를 나타낸다. 아두이노 우노는 브레드 보드라고 불리는 무 뎀납 장치를 이용하여 도트 매트릭스와 블루투스 실드 (HC-06)를 연결하여 아두이노와 상호작용을 한다.



[그림 4.8] fritzing을 이용한 아두이노 구성도

<그림 4.9>는 실제 조립된 아두이노 기기의 전체적 모습이다. 아두이노 2, 3은 Bluetooth Module의 각각 TXD, RXD과 일치시킨다. 10, 11, 12는 DotMatrix Module의 각각 CS, CLK, DIN에 일치시킨다. 두 모듈 모두 GND와 5V에 VCC를 연결하면 구조는 마무리된다. 아래 <표 4.3>은 모듈의 Pin 구성을 보여준다.



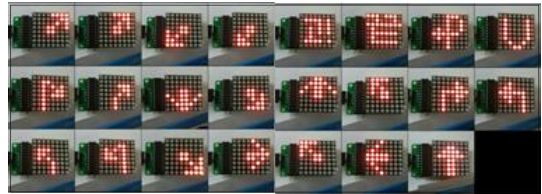
[그림 4.9] 아두이노 디바이스 전체 구성도

[표 4.2] Module연결 및 Pin 설정

Arduino UNO	Bluetooth Module	DotMatrix Module
2	TXD	
3	RXD	
10		CS
11		CLK
12		DIN
GND	GND	GND
+5V	VCC	VCC

아두이노는 LedControl 라이브러리를 이용하여 도트 매트릭스를 사용한다. 먼저 LedControl과 SoftwareSerial을 통해 DotMatrix와 Bluetooth 모듈의 핀 번호를 입력한다. setup()에서 블루투스 모듈과 시리얼 모니터를 loop()에서 블루투스 데이터 수신 여부를 기다린다. 처음 시작 시에는 안드로이드와 블루투스 연동이 되지 않았기 때문에 사용자가 인식하게끔 무표정의 도트를 출력한다. 안드로이드로부터 전환 번호를 수신 받으면 문자열의 끝부분 '\n'이 나타날 때까지 문자 하나씩 버퍼에 입력받는다. 그다음 버퍼의 내용을 writeArduinoOnMatrix()으로 보낸다. case 문은 문자열을 인식하지 못하기 때문에 버퍼를 char\*를 사용하여 옮겨 놓은 뒤에 atoi를 이용하여 integer로 변환/정제한다. 이 정제된 값을 통해 알맞은 조건문을 실행시켜 도트 매트릭스로 내비게이션을 출력하

도록 구현하였다. <그림 4.10>과 같이 안드로이드 단말로부터 받은 내비게이션 전환 번호를 아두이노 디바이스에서 총 23가지 방향 정보를 도트 매트릭스로 출력할 수 있도록 구현하였다.



[그림 4.10] 전환 번호에 따른 23가지 출력 값

## 5. 결론 및 향후과제

본 논문에서는 안드로이드와 아두이노 디바이스간의 접목을 통하여 직관적이며 안전성을 개선한 내비게이션 기능을 사용자에게 제공하도록 구현하였다. 본 논문에서 구현한 기능을 통해 사용자는 교통사고 예방과 함께 안전하고 쾌적한 라이딩 서비스를 이용할 수가 있다. 한 가지 아쉬운 점이 있다면 SK 플래닛에서 제공해주는 내비게이션 정보와 최단 경로 기능 부분이다. 먼저 내비게이션 정보는 목적지까지의 전환 지점에 따라 정보가 존재하는데 학교 캠퍼스 내처럼 세밀한 위치까지는 제공해주지 않아서 내비게이션이 기대한 만큼 정확하게 나타나지 않는 경우가 많이 있었다. 그리고 개발자에게 하루에 한 정된 트래픽 량을 제공하기 때문에 다수의 사용자가 애플리케이션을 사용한다면 원활한 서비스를 제공받는데 다소 한계가 있다는 점이다. 물론 전체 트래픽 양은 SK 플래닛 개발자 센터에 문의할 경우 추가 확대가 가능하다는 점을 확인하였으나 앞으로도 지속적인 유지보수가 꼭 필요하다고 할 수 있다.

향후 과제로 아두이노 기반 다양한 센서를 이용하여 사용자 개개인이 요구하는 형태의 맞춤으로 서비스를 제공하는 것이다. 예를들어 소리에 민감한 사용자라면 스피커 센서를 부착시켜서 일정 속도 이상으로 주행할 경우 알람 메시지를 발생시킬 수도 있으며, 현재까지 소모된 칼로리 양을 소리로 제공할 수가 있다. 또 습도, 온도 센서 같은 기상 센서를 추가한다면 웹에서 기본적으로 제공하는 정보보다 정확한 실시간 정보를 전달하여 날씨에 민감한 유아나 노인에게 보다 향상된 서비스를 제공할 수 있을 것으로 기대된다.



REFERENCES

- [1] 데이코산업연구소, “급성장하는 위치기반서비스(LBS)와 증강현실 기술, 시장 및 사업동향”, 생테이코출판사, 2010
- [2] 안드로이드, <https://ko.wikipedia.org/wiki/안드로이드>, 위키백과
- [3] Jennifer Bray “블루투스” 홍릉 과학 출판부, 2001
- [4] 서민우, “아두이노 안드로이드로 45개 프로젝트 만들기”, 앤써북, 2014
- [5] 아두이노, <https://ko.wikipedia.org/wiki/아두이노>, 위키백과

김 동 건(Dong-Gun Kim)



- 2009년 2월 : 진산고등학교 졸업
- 2016년 2월 : 한신대 컴퓨터공학부 (학사)

<관심분야>

사물인터넷, 정보보호, 모바일 앱 개발

이 호 세(Ho-Se Lee)



- 2009년 2월 : 단대부속고등학교 졸업
- 2016년 2월 : 한신대 컴퓨터공학부 (학사)

<관심분야>

사물인터넷, 정보보호, 모바일 앱 개발

김 태 우(Tae-Woo Kim)



- 2009년 2월 : 병점고등학교 졸업
- 2016년 2월 : 한신대 컴퓨터공학부 (학사)

<관심분야>

사물인터넷, 정보보호, 모바일 앱 개발

김 소 영(So-Young Kim)



- 2009년 2월 : 인명여자고등학교 졸업
- 2016년 2월 : 한신대 컴퓨터공학부 (학사)

<관심분야>

사물인터넷, 정보보호, 모바일 앱 개발

이 형 우(Hyung-Woo Lee)

[중신회원]



- 1994년 2월 : 고려대학교 컴퓨터학과 (학사)
- 1996년 2월 : 고려대학교 컴퓨터학과 (석사)
- 1999년 2월 : 고려대학교 컴퓨터학과 (석사)

▪ 2003년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 교수

<관심분야>

사물인터넷, 정보보호, 모바일 보안 및 디지털 포렌식