

# 아두이노를 활용한 어린이 교통사고 예방 시스템 설계 및 구현에 관한 연구

이호림\*, 김혜주\*, 이희재\*

\*수원대학교 정보통신공학과

horim27@gmail.com, gpwndid98@naver.com, usj11@naver.com

## A Study on the Design and Implementation of the Child Traffic Accident Prevention System using Arduino

Horim Lee\*, Hyeju Kim\*, Heejae Lee\*

\*Dept. of Information and Telecommunication Engineering, University of Suwon

### 요 약

최근 ‘민식이 법’으로 인해 교통사고에 대한 이슈가 대두되고 있다. 어린이는 안전한 횡단을 보장받을 수 있지만, 운전자들이 불편함을 감수해야 하는 단점도 존재한다. 본 논문에서는 어린이도 지금보다 더 안전하게 횡단할 수 있고, 운전자들도 스쿨존에서의 불안감 및 부담을 줄이기 위해 아두이노를 활용한 어린이 교통사고 예방 시스템을 설계하고 구현한다.

### 1. 서론

최근 ‘민식이 법’으로 인하여 스쿨존에서의 사고에 대한 처벌 수위가 강화됐다. 이는 운전자의 불안감을 조성하는 과잉처벌법으로 여론이 조성되고 있다. 실제 ‘민식이 법’ 개정을 요구하는 청원은 35만 명을 돌파하며 해당 법안에 부당함을 느끼고 있는 운전자가 많다는 것을 보여준다. 또한 ‘민식이 법’이 생겨난 후 일부 내비게이션 회사에서 내비게이션 서비스에 스쿨존 우회 기능을 탑재하여 판매하고 있다 [1]. 일부 내비게이션 회사에서 정부에서 제공하는 스쿨존 정보를 이용해 목적지까지 스쿨존을 회피해 안내하는 스쿨존 우회 기능을 탑재하여 판매하고 있다. 하지만 이는 모든 스쿨존을 피해 갈 경우 목적지까지 걸리는 시간이 급격히 늘어날 수 있고, 스쿨존을 차량 통행이 불편한 곳으로 안내한다면 오히려 사고에 취약해지는 결과를 낳을 수 있다.

본 논문에서는 어린이와 운전자 모두를 고려하여 어린이들은 안전하게 횡단할 수 있을 뿐만 아니라 운전자들은 안전하게 운전할 수 있도록 시스템을 설계하고 구현하였다. 이는 운전자가 상황을 미리 인지하고 대처할 수 있도록 하여 피해를 최소화할 수 있다. 특히 어린이 위치를 정확히 인지하고 운전자가 감지할 수 있도록 Android Studio와 임베디드 시스템(Arduino)을 활용해 어린이 교통사고 예방 시스템을 구현했다. 위 시스템은 갑자기 일어날 수 있는 어린이의 교통사고를 예방할 수 있다. 주요 기능으로는 어린이를 사용자로 하는 애플리케이션과 임베디드 시

스템 (Arduino)에 내장되어있는 차량과의 거리계산을 통해 어린이가 차량 반경 5m 이내로 들어오면 차량 속도를 자동으로 제어한다. 또한, 어린이가 차량 반경 10m 이내로 들어오면서부터 위험을 알리는 버저를 울리며 운전자에게 경각심을 주어 부주의로 발생할 수 있는 스쿨존 및 어린이 사고를 미리 방지할 수 있다.

### 2. 시스템 구성

#### 2.1 Raspberry pi

영국의 라즈베리 재단에서 만든 신용카드 크기의 초소형 컴퓨터이다 [2]. 가격은 5~10만 원 대로 부담감이 줄어들기 때문에, 아두이노와 함께 많은 개발자가 활용하고 있다. 그뿐만 아니라 접근성이 뛰어나 일반인도 손쉽게 접근할 수 있다. 본 논문에서는 Raspberry Pi 4를 이용하여 웹서버 구축 및 차량과 모바일의 거리를 계산하여 차량의 제어 신호를 보내는 기능을 구현하였다.

#### 2.2 관계형 데이터베이스(Relational Database)

관계형 데이터베이스란 상호 관련된 데이터 포인트에 대한 액세스를 저장하고 제공하는 데이터베이스의 한 형태이다. 데이터를 테이블에 직관적으로 간단하게 나타내는 관계형 모델을 기반으로 하며, 각 레코드는 일반적으로 각 속성에 대한 값을 가지므로 데이터 포인트 간 관계를 손쉽게 구축할 수 있다 [3]. 보다 데이터를 안전하고 일관성 있게 관리하기 위해 본 논문에서는 관계형 데이터베이스를 통해 구현했다.

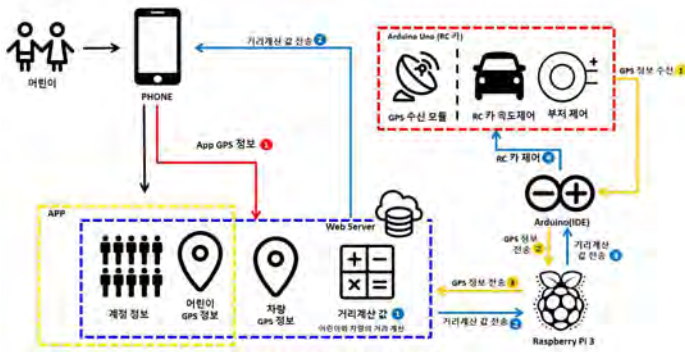
### 2.3 안드로이드 스튜디오(Android Studio)

구글이 안드로이드 앱 개발을 위해 JetBrains 사의 IntelliJ IDEA를 기반으로 만든 통합 개발 환경이다. 무료로 제공되고 있으며 Windows, macOS, 리눅스 등의 운영체제에서 지원한다 [4]. 본 논문에서는 Windows에서 서비스를 이용하며 본 서비스에서 지원하는 프로그래밍 언어 중 JAVA를 통해 애플리케이션의 주요 기능 개발과 디자인 설계를 구현하였다.

### 2.4 Arduino

아두이노는 물리적인 세계를 감지하고 제어할 수 있는 인터랙티브 객체들과 디지털 장치를 만들기 위한 도구로, 간단한 마이크로컨트롤러 (Microcontroller) 보드를 기반으로 한 오픈 소스 컴퓨팅 플랫폼과 소프트웨어 개발 환경을 말한다 [5]. 이처럼 차량에서 감지할 수 있고 제어할 수 있도록 본 논문에서 Arduino를 활용해 구현하였다.

## 3. 시스템 설계



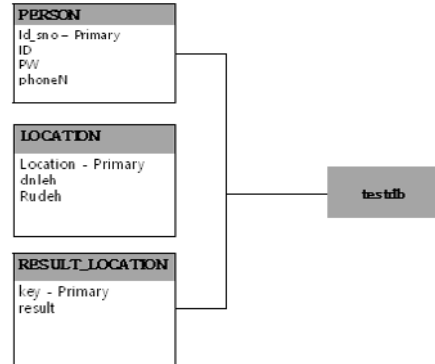
(그림 1) 전체 구성도

그림 1은 본 논문에서 제안한 시스템의 전체 흐름도를 보여준다. 애플리케이션의 경우 웹서버 연동 회원가입-로그인을 통해 메인화면 페이지로 접근한다. GPS 정보 전송을 위해 버튼을 누를 시 위치 권한 허용 페이지로 넘어간다. 권한 허용 후 GPS(위도, 경도) 정보가 Socket 통신을 통하여 Web Server로 들어간다. 아두이노의 경우 차량의 GPS 정보가 Serial 통신을 통해 Raspberry pi 3으로 전송된다. Raspberry pi 3에서는 아두이노 GPS 정보를 Database LOCATION table의 각 필드로 위도, 경도 값이 저장된다. Web Server에서 아두이노 및 모바일의 GPS 값을 python haversine 라이브러리를 사용하여 두 물체 간의 거리를 계산 후 그 값을 Database RESULT\_LOCATION table의 location 필드에 저장한다. 모바일 및 아두이노는 실시간으로 Database에 저장된 거리값을 요청한다. 값이 들어왔을 때 아두이노의 경우 거리가 10m 이내일시 40 db로, 5m 이내일시 차량의 속도제어 및 60 db로 버저를 울린다. 애플리케이션의 경우 거리가 1m 반경일 시 Dialog로 사용자에게 경고 알람을 준다.

## 4. 시스템 구현

### 4.1 DB설계

그림 2는 본 논문의 애플리케이션 구현과 데이터 간의 계산을 위해 사용한 데이터베이스를 나타낸 것이다. 본 논문에서는 testdb 한 개의 데이터베이스에서 PERSON, LOCATION, RESULT\_LOCATION 3개의 데이터베이스 테이블을 나타내고 있다.



(그림 2) 전체 구성도

그림 3, 4, 5는 각각 PERSON과 LOCATION, RESULT\_LOCATION 데이터베이스 테이블 정의를 나타내고 있다.

열: + 추가 - 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	id_sno	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME..
2	ID	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	PW	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	phoneN	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

(그림 3) PERSON

그림 3은 사용자의 Id와 Password를 통해 회원가입을 가능하게 하도록 ID, PW 필드를 생성하였고 사용자들의 구분을 위해 id\_sno 속성을 Primary Key로 지정하였다. 또한, 사용자가 8세~13세인 점에서 부모님 전화번호를 저장할 수 있는 phoneN 필드도 생성하였다.

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	location	INT	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME..
2	dnleh	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	rudeh	VARCHAR	30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

(그림 4) LOCATION

그림 4는 Arduino IDE에서 RC카가 현재 위치한 위도, 경도, Primary key 3개의 필드로 구성된 테이블이다.



(그림 5) RESULT\_LOCATION

그림 5는 Web Server에서 애플리케이션과 아두이노와의 거리를 계산한 데이터를 저장하는 result, Primary key 2개의 필드로 구성된 테이블이다.

#### 4.2 애플리케이션 설계

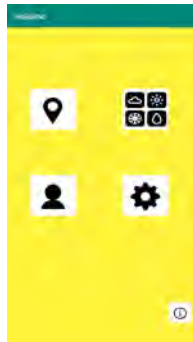
그림 6, 7, 8은 애플리케이션 구현 화면이다.



(그림 6) 로그인



(그림 7) 회원가입



(그림 8) 메인화면

그림 6은 로그인 페이지이다. 회원가입 버튼을 누르면 회원가입 할 수 있는 페이지로 넘어간다. Id와 Password를 입력하고 로그인 버튼을 누르면 메인화면 페이지로 이동하게 된다.

그림 7은 회원가입 페이지이다. Id, Password, 부모님 전화번호까지 입력하여 사용자를 등록할 수 있다.

그림 8은 메인화면 페이지이다. 메인화면 페이지에서는 GPS 서비스 권한 허용 버튼, 날씨에 따른 도로의 위험성 버튼, 계정 정보 버튼, 설정 버튼으로 구성되어 있다.



(그림 9) GPS 페이지



(그림 10) GPS 알림

그림 9, 10은 GPS 버튼을 눌렀을 때의 그림이다. “GPS” 버튼을 누르면 모바일에 내장된 위치 서비스 권한 허용 페이지로 넘어가며 위치 서비스를 허용하거나 거부할 수 있다. 서비스를 허용하면 그림 10에서 보이듯이 일정 시간마다 현재 위치한 위도, 경도 데이터를 나타낸다.

#### 5. 결론

본 논문에서는 ‘스쿨존’에서의 어린이 교통사고 예방 시스템을 애플리케이션 및 차량 제어 시스템 형태로 설계하고 이를 구현하였다. 본 논문에서 구현한 애플리케이션은 어린이 사용자들의 회원가입 및 로그인 기능을 구현하였다. 그뿐만 아니라 사용자의 현재 위치 정보를 실시간으로 모바일 화면에 보여준다. 차량 역시 현재 위치 정보를 실시간으로 전송하여 사용자와의 거리를 계산하고 그 값에 따라 버저를 울려 운전자가 어린이의 위치를 대략 알 수 있으므로 운전자에게 경각심을 주어 교통사고 발생률을 낮출 것이라 기대한다. 또한, 사용자가 운전자 위험 반경 이내로 들어올 시 차량의 속도제어 기능을 개발했다. 이는 갑작스럽게 접근하는 사용자를 미처 운전자가 반응할 수 없는 경우에 도움이 될 것이다.

이 논문은 2020년도 수원대학교 사제동행 프로그램 지원을 받아 수행된 결과물입니다.

본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다.

#### 참고문헌

[1] <http://www.kaupress.com/news/articleView.html?idxno=1358>  
 [2] [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)  
 [3] Oracle, "https://www.oracle.com/kr/database/what-is-a-relational-database/"  
 [4] Namuwiki - Android Studio, <https://namu.wiki/w/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C%20%EC%8A%A4%ED%8A%9C%EB%94%94%EC%98%A4>  
 [5] 두산백과 - Arduino, "https://terms.naver.com/entry.nhn?docId=2835912&cid=40942&categoryId=32828"