

버스 승하차 사전 알림 시스템

박준영, 김두현, 김수호, 박진우, 최병조
인천대학교 임베디드시스템공학과

e-mail: {juny1905, ohs342, procode9, bestjwjwjw}@naver.com
bjc97r@inu.ac.kr

Bus boarding advance notice system

Doo-Hyeon Kim, Su-ho Kim, Jun-young Park, Jin-woo Park, Byeong-jo Choi
Dept. of Embedded Systems Engineering, Incheon National University

요 약

본 논문은 공공데이터 API를 활용한 ITS(Intelligent Transport System)를 구현하여, 버스 승하차 관련 문제를 개선하는 버스 승하차 사전 알림 시스템에 관한 것이다. 수도권 및 대도시 전체 버스 관련 민원신고 중 가장 많이 차지하는 부분이 버스의 정류장 무정차 통과 및 승차거부 문제다. 이를 해결하기 위해 사전에 승객의 승하차 여부를 버스 운전자에게 미리 알리는 시스템을 제안한다. 이 시스템은 승객용 애플리케이션, 중앙 서버 그리고 버스에 설치되는 디바이스로 구성되어 있으며 예비 승하차 승객이 애플리케이션을 이용하여 자신이 원하는 버스에 대한 정보를 MQTT Broker를 이용하여 서버에 보내고 서버는 해당 버스의 정보와 현재 노선의 운행 현황을 주기적으로 확인하여 요청 정류장 이전에 도달하였을 때 버스 운전자 측 디바이스에 신호를 보내어 버스 운전자가 예비 승하차 승객이 있음을 사전에 알 수 있게 한다. 이 시스템을 통해 무정차 및 승차거부 문제를 근본적으로 해결하여 사용자 편의·승객 안전·일반 차량 운전자 안전·도로 교통 안정화를 도모한다.

1. 서론

전체 불편 민원 가운데 가장 많이 차지하는 민원은 ‘승하차 전 출발 및 무정차 통과’이다[1]. 많은 버스 승객들이 버스 무정차 관련 불편함을 꾸준히 겪고 있다. 이는 버스 관련 민원에도 고스란히 반영되어 있다. 인천시의 경우 무정차 관련 신고율이 48.4%[2], 서울시는 58.5%[3], 부천시 42.3%[4]로, 서울/경기/인천에 걸쳐, 수도권 민원 신고의 절반에 가까운 비율을 차지하고 있다. 무정차 문제를 해결하기 위해 집중 단속 등 노력을 하고 있지만, 비효율적이며 실질적인 효과도 없는 것으로 나타났다.

무정차 하는 버스를 막기 위해 승객들이 도로에 나와 승차 신호를 보내는 위험한 상황이 자주 보이며, 이러한 문제를 해결하기 위해 ‘버스 베이’를 설치하였으나, 버스 베이에 제대로 서는 버스는 30%뿐이다. 또한 버스 베이를 이용한 승객 4명 중 1명은 승하차 때 위험을 경험했으며, 전체 자동차 운전자 중 80.3%가 버스 베이를 이용하지 않는 버스 때문에 운전 중 위험한 순간을 경험했다고 답변했다[5].

이렇듯 버스 무정차 문제는 단순히 버스 이용자들의 불편함에만 영향을 미치는 것이 아니라, 승객 및 일반 차량 안전 문제에도 영향을 미치고 있다.

또한 본 연구 팀에서 다수의 버스기사들을 상대로 인터

뷰한 결과, 많은 버스 기사들이 버스정류장을 앞두고 차선 변경함에 있어서 위험을 느끼고 있음을 알게 되었다. 그리고 버스정류장에 승객이 있는지 없는지 미리 알 수 있다면 안전하게 차선을 변경할 수 있다는 것을 확인하였다. 특히 차선 변경이 어려운 곳이나 신호가 복잡한 곳, 출퇴근길과 같은 교통 정체 상황에서 다음 버스정류장에 해당 버스를 탈 승객이 있음을 미리 알 수 있다면, 버스기사가 주행함에 있어서 도움이 많이 될 수 있다는 것을 확인하였다.

정류장에 탑승할 승객이 있음을 미리 알리는 것은 승객과 버스 운전기사, 일반 차량 운전자, 무정차 단속 공무원 등 여러 사람들에게 도움이 될 것으로 분석했다. 승차 문제를 개선함과 동시에 하차 문제 또한 개선할 수 있을 것으로 보아 이에 대해 종합적으로 해결책이 될 수 있는 ‘버스 승하차 사전 알림 시스템’을 제안한다.

사전 알림 시스템을 통해 승객은 애플리케이션을 사용하여 탑승할 정류장에서 특정 버스로 승하차 요청을 보낼 수 있다. 이러한 요청은 해당 버스 운전기사에게 음성 및 시각적 알림을 통해 전달할 수 있다.

2. 설계 및 구조

(그림 1)은 버스 승하차 사전 알림 시스템의 사용 시나리오를 나타낸 것이다. 스마트폰 애플리케이션에서 사용자로부터 정보를 입력받는다.

여기에는 탈 것인지 내릴 것인지에 대한 사용자의 요청

* 이 논문은 2018 한이음 ICT멘토링 프로젝트의 연구비 지원을 받은 결과물입니다.

정보, 사용자가 요청한 버스 정보, 버스 위치정보가 포함된다. 입력받은 정보들을 MQTT 서버에 보낸다. 서버는 공공 API를 통해 사용자가 요청한 버스에 대한 정보를 10초마다 확인한다. 버스가 사용자가 요구한 정류장의 전 정류장에 도착하면 버스 디바이스로 신호를 전송한다.



(그림 1) 승하차 사전 알림 시스템 사용 시나리오

3. 주요 기술

3.1. MQTT

MQTT는 통신 프로토콜 중 하나로, M2M(Machine to Machine)과 IoT(Internet of Things)에서 사용하기 위해 만들어졌다. 임베디드 환경에서도 낮은 전력, 낮은 대역폭 환경 등 기존보다 개선된 에너지 효율을 사용할 수 있도록 설계됐다[6]. 데이터를 토대로 기기들을 제어할 수 있는데 장점이 있어서 MQTT 프로토콜을 사용했다.

(그림 5)과 같이 MQTT Broker는 특정 주제(Topic)를 가지고 발행(publish)된 메시지를 같은 주제를 구독(Subscribe)하는 대상에게 전달해주는 중재자 역할을 한다[7].

본 연구에서는 탑승할 승객이 애플리케이션에서 선택한 타요/내려요 여부, 노선 번호, 정류장 번호, 버스 번호판 정보가 담긴 메시지를 ‘버스 요청’(BUS_REQUEST)이라는 주제를 가지게 하여 MQTT Broker로 발행한다. 이때 서버는 상시 ‘버스 요청’주제를 구독하고 있기에 사용자의 버스 요청 메시지를 가지고 온다. 서버는 받아온 메시지에 담긴 정보를 바탕으로 해당 버스가 사용자가 요청한 정류장의 직전 정류장에 도착했을 때 버스 내에 장착된 디바이스로 타요/내려요 여부, 요청 정류장 이름, 도움 필요 여부를 ‘버스 응답’(BUS_RESPONSE) 주제로 MQTT Broker에 발행한다. 디바이스 또한 상시 ‘버스 응답’ 주제를 구독하고 있기에 메시지가 도달하면 그 정보를 바탕으로 운전자에게 TTS(Text-To-Speech) 및 LED 알림판으로 알림을 제공한다.

3.2. 버스 디바이스

운전기사에게 다음 정류장에 멈춰야 할 것을 직관적으로 알려주기 위하여 LED 표시 기능이 있으며, 소형 스피커를 통해, 음성으로도 알려준다. 여기에는 TTS 기술을 적용하였다. 디바이스 제어기로는 개발의 확장성을 위하여 라즈베리 파이(Raspberry Pi)를 사용하였다.

3.3. 교통약자를 위한 추가 기능

승객들의 승하차에 대한 어려움을 개선하기 위해 만들어진 시스템인 만큼 교통약자의 경우 특별 알림을 주는 기능이 있다. 이는 교통약자의 이동 권리를 보장하는 기능으로써, 장애인, 임산부 등 혼자 힘으로 버스 승하차가 어려운 사람의 경우 일반적인 알림과 다른 알림으로, 버스기사에게 도움이 필요함을 미리 알린다. 버스기사는 휠체어 승강 설비 등의 교통약자를 위한 이동 편의시설을 미리 준비할 수 있다.

4. 구현

어플리케이션과 서버, 버스 디바이스의 구현 결과를 보여준다.

4.1. 타요, 내려요! 애플리케이션 구현



(그림 2) 대기화면 및 정류장 번호 검색 화면

4.1.1. 대기 화면 및 정류장 번호 검색 화면

(그림 2)의 우측에 나타난 화면에서 검색창에 직접 노선번호, 정류소 이름, 정류소 번호 등을 입력할 수 있으며, 필요에 따라 버튼 클릭 방식으로 원하는 서비스를 빠르게 이용할 수 있다.

4.1.2. 정류장 검색 화면



(그림 3) 정류장 검색 및 장소 설정 화면

(그림 3)의 우측에서 Public Data API를 이용하여 리스트 형태로 정류장 번호를 제공받을 수 있다. 또한, (그림 3)의 좌측에 나타난 지도의 선택 화면에서 구체적인 정류장을 선택할 수도 있다.

4.1.3. 버스 및 정류장 예약 화면

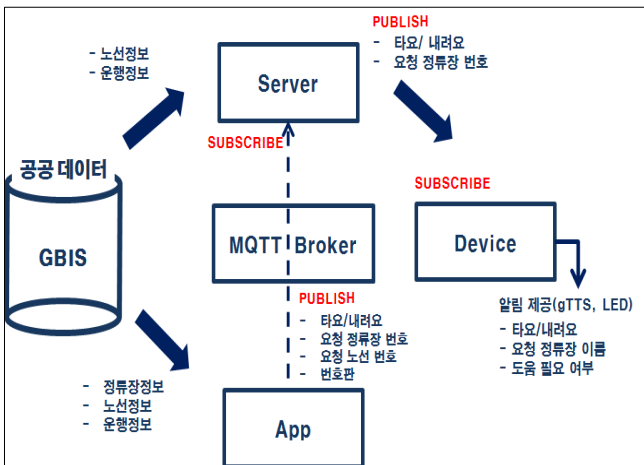


(그림 4) 정류장 선택 및 승하차 예약 화면

(그림 4)의 좌측 화면에서 탈 버스와 내릴 정류장을 선택하여 예약할 수 있다. 정류장의 이름과 함께 실제로 서버에서 조건을 판별할 때 사용할 정류장 순서 번호와 현재 운행 중인 버스 상태를 확인할 수 있다.

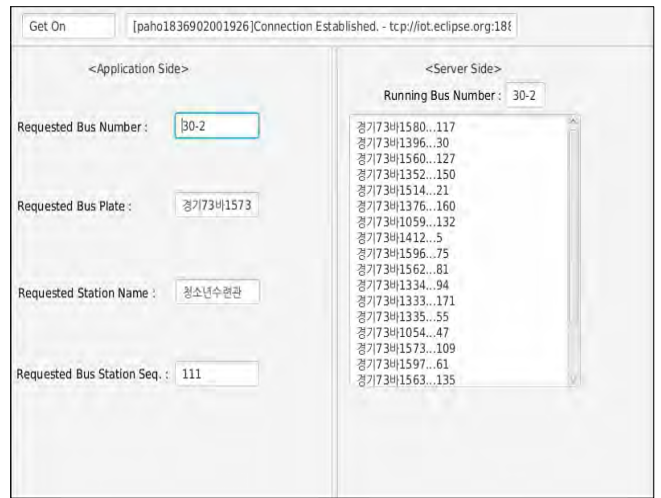
(그림 4)의 우측은 예약이 성공적으로 완료된 모습이다. 모바일 애플리케이션을 통해 승차 예약이나 하차 예약을 하면, MQTT 서버로 해당 정류장과 버스 정보가 담긴 메시지를 보낸다. 서버에서는 이를 공공데이터에서 받은 버스 운행 현황과 계속 비교하여, 적절한 시기에 버스에게 신호를 보낸다.

4.2. 서버 구현



(그림 5) 시스템 다이어그램

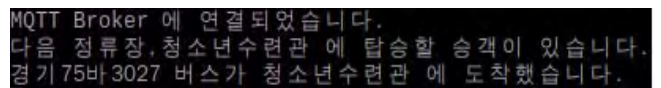
(그림 5)의 서비스 루틴에 의해 버스 운전자 측 디바이스에 신호가 가게 된다.



(그림 6) 서버 모니터링 화면

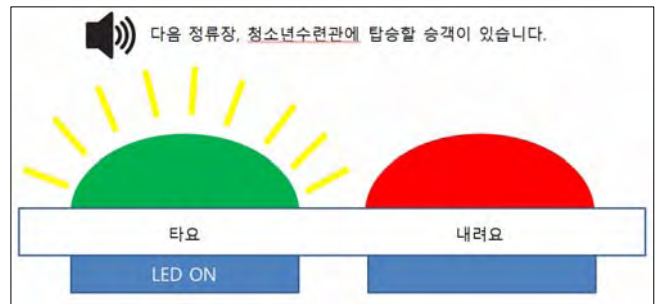
(그림 6)은 서버에서 APP으로부터 받은 데이터와 지정된 버스의 데이터를 모니터링하고 있는 화면이다.

4.3. 버스 디바이스 구현



(그림 7) 터미널에 출력된 디바이스 상태

디바이스의 연결 상태를 상시 확인하기 위하여 터미널에 메시지를 출력한다. 두 번째 줄에서 확인할 수 있는 탑승 승객 알림 메시지와 같이 음성합성을 이용하여 버스 운전수에게 알림을 제공한다.



(그림 8) 디바이스의 외형 예상도

(그림 8)은 버스 기사에게 버스가 정차해야 함을 알려주기 위해, LED를 켜진 모습의 예상 이미지이다. 하차할 승객이 있을 시 다른 LED가 켜지고, 승하차에 이상이 있는 승객의 메시지가 날아오면 LED가 깜빡인다.

5. 향후 연구

해당 시스템은 애플리케이션 사용자가 해당 정류장이나 버스 내부에 존재하지 않아도 타요, 내려요 알림을 보낸다. 이를 해결하기 위해 정류장, 버스와 스마트폰과의 GPS 대조를 통한 위치 대조시스템을 추가하여 부정 사용

자의 접근을 제한함으로써 더 신뢰성 높은 프로그램을 개발할 필요가 있다.

6. 결론

교통약자의 불편함과 버스 무정차 문제는 지속적으로 해결되지 않는 문제다. 이를 본 논문에서는 기존의 육안으로만 승객을 확인하고, 직접 하차 벨을 눌러야만 하는 기존 버스시스템을 개선하려 한다.

공공 Data, MQTT를 통한 저전력 설계, 무료 모바일 어플리케이션을 통해 문제 해결을 위한 경제적 부담을 최대한 줄이려고 하였다. 또한, 새로운 버스를 설계/생산하거나 기존의 버스를 개조하지 않고, 작은 디바이스를 설치하는 것만으로 동작하도록 하여 현실적으로 문제 해결하고자 했다.

그리고 버스 운전자와 승객 인터뷰를 통해, 본 논문이 제시하고 있는 시스템이 실제 사용자들에게 직접적인 도움이 될 수 있음을 확인하였다.

참고문헌

- [1] 서울시내버스 ‘무정차 통과’ 3년 연속 불편민원 1위, 뉴스1, 2016/11/03, <http://news1.kr/articles/?2819792>, (2018/09/07 - 접속 날짜)
- [2] ‘무정차 통과 열반네’ 버스 민원 절반... 인천시, 집중 단속, 연합뉴스, 2017/09/11 1:27, <http://www.yonhapnews.co.kr/bulletin/2017/09/11/0200000000AKR20170911064300065.HTML?input=1195m>, (2018/09/07 - 접속 날짜)
- [3] [‘서민의 발’이 무서워요!] 사고 나면 대형참사 ‘만원버스’... “이게 최선입니까?”, 헤럴드경제, 2018/05/15 08:06, <http://news.heraldcorp.com/view.php?ud=20180515000042>, (2018/09/07 - 접속 날짜)
- [4] 부천시, 시내버스 무정차 난폭 운행 등 압행어사 합동 점검반 뜯다, 경기일보, 2017/05/23, <http://www.kyeonggi.com/?mod=news&act=articleView&idxno=1355007>, (2018/09/07 - 접속 날짜)
- [5] ‘버스 베이’에 제대로 서는 버스 30%뿐, 한겨레, 2015/07/22 22:11, <http://www.hani.co.kr/arti/society/area/701461.html>, (2018/09/07 - 접속 날짜)
- [6] 라즈베리 파이에서 MQTT 프로토콜 이용 시 전력 소모 분석, 강승우 2017/12
- [7] 사물인터넷과 MQTT 기술, 심승현, 김학범 2014/12