자율주행차 운행 데이터 및 서비스 인터페이스를 위한 통합 소프트웨어 플랫폼 설계

Design of the Integrated Software Platform for Autonomous Vehicle Operation Data and Service Interface

임동민[1], 김현용[2], 김곤우[3,*]

Dong-Min Lim[1], Hyun-Yong Kim[2], Gon-Woo Kim[3,*]

[1]충북대학교 산업인공지능학과
[2]충북대학교 산업인공지능연구센터
[3]충북대학교 지능로봇공학과

[1]Department of Industrial Artificial Intelligence, Chungbuk National University, Cheongju, Chungbuk Korea
[2]Industrial AI Research Center, Chungbuk National University, Cheongju, Chungbuk Korea
[3]Department of Intelligent Robot Engineering, Chungbuk National University, Cheongju, Chungbuk Korea

Abstract

In order to commercialize autonomous vehicles, it is important to effectively manage driving data and respond to various services. Previous autonomous driving platforms have a scalability issue in that multiple H/W modules, protocols, and communication methods are mixed in the scope of service. Especially for heterogeneous autonomous driving platform, existing services are not reusable because of the different interfaces. Therefore, this paper presents a integrated software platform that contains interface the operation data of various heterogeneous autonomous vehicles to respond to various services such as state and control settings of autonomous vehicles, connection with integrated control systems, visualization HMI, external display devices, and reservation processing for On-Demand. This proposed platform named as VBS (Vehicle Bridge Stack) can reuse many existing services and extend the service area without redesign. The reliability of the proposed platform was verified by the data of a real autonomous driving car.

Keywords: *Autonomous vehicle, Driving data, Software platform, Service interface*

요 약

자율주행차 상용화를 위해서는 운행 데이터를 효과적으로 관리하여 다양한 서비스에 대응하는 것이 중요하다. 기존의 자율주행 플랫폼의 경우, 다수의 H/W 모듈, 프로토콜, 통신방식이 혼재되어 있어 확장성이 떨어지는 문제점이 있었다. 특히, 이기종 자율주행차의 경우 이러한 서로 다른 인터페이스 때문에 기존에 구축되어 있는 서비스들을 재사용할 수 없었다. 따라서 본 연구에서는 이기종 자율주행차의 운행 데이터를 인터페이스화 하여 자율주행 차량의 상태 및 제어설정, 통합관제 시스템과의 연계, 시각화 HMI (Human Machine Interface), 외부 표시장치, On-Demand를 위한 예약처리 등의 다양한 서비스에 대응할 수 있는 통합 소프트웨어 플랫폼을 설계하였다. 제안하는 소프트웨어 플랫폼, VBS(Vehicle Bridge Stack)는 이기종 자율주행차를 위한 기존의 다양한 서비스를 활용할 수 있을 뿐만 아니라, 서비스 범위를 재설계 없이 확장할 수 있다. 이 플랫폼은 실제 자율주행 차량의 데이터를 이용하여 성능 및 신뢰성 테스트를 통해 검증하였다.

Keywords: *자율주행차, 운행 데이터, 소프트웨어 플랫폼, 서비스 인터페이스*

* 교신저자: gwkim@cbnu.ac.kr

1. 서론

구글과 애플 같은 정보기술 업체는 물론 BMW, 벤츠, 도요타 등 완성차 업체들은 2025년을 자율주행 상용화의 원년으로 삼고 기술개발에 사활을 걸고 치열한 경쟁을 벌이고 있다[1]. 자율주행 구현을 위해서는 도로 위의 많은 정보들(예: 도로, 표지판, 신호등 등)을 수집하여 인공지능 학습용으로 사용할 방대한 데이터베이스를 구축하는 것이며, 이를 위해 '자율주행 데이터 수집 차량'도 활용될 전망이다. 이에 정부기관 및 지자체에서는 다양한 기종의 자율주행차로부터 운행 데이터를 수집하여 모니터링 하고 수집된 차량 운행 데이터베이스를 연구기관 및 기업연구소에 제공하여 자율주행차 고도화 및 상용화 기술개발을 촉진하는 선순환 구조를 구축하고자 노력하고 있다[2].

그러나 자율주행차는 전통적인 기계 중심에서 벗어나 전자, IT, 인공지능 등 다양한 ICT 기술이 융복합된 기기이기 때문에 적용기술 또한 다양해서 통일된 표준이 없는 실정이다. 기업, 학교, 연구기관과 오픈 플랫폼 등의 다양한 자율주행 관련 연구들에서는 각각 다르게 정의된 운영 데이터와 서로 다른 통신 프로토콜을 사용하기 때문에 사용자가 원하는 정보를 얻기가 쉽지 않다. 이에 다양한 기종의 자율주행차의 운행 정보를 수집·통합하기 위해서는 각 기관마다 운행 정보에 대한 데이터 모델링부터 각 정보의 통신방식까지 개발해야 하기 때문에 시간과 비용이 투자되는 어려움이 있다.

따라서 자율주행 상용화에 박차를 가하기 위해서는 주행 정보를 효율적으로 수집·통합하고 이 데이터를 활용하여 다양한 서비스를 제공할 수 있는 자율주행차 운행 데이터 통합 플랫폼이 필요하다.

본 연구에서는 이러한 문제를 해결하기 위해 다양한 기종의 자율주행차 운행 데이터를 손쉽게 인터페이스할 뿐만 아니라 데이터의 분류 및 처리를 통해 데이터의 HMI(Human Machine Interface) 시각화, 외부 표시장치로의 출력, 통합관제센터로의 데이터 전송 등의 다양한 서비스를 제공할 수 있는 통합 소프트웨어 플랫폼을 제시하고자 한다. 이를 통해 자율주행차의 운행 데이터를 이용하여 고객이 필요한 서비스를 제공하기 위한 플랫폼 개발을 단순화함으로써 개발비용과 장애를 최소화하고 자율주행차 실증과 상용화에 기여하고자 한다.

2. 소프트웨어 플랫폼 설계

2.1 소프트웨어 플랫폼 개념

소프트웨어 공학에서 플랫폼(platform)이란 서비스의 개발, 실행 및 기기의 동작에 공통적으로 활용되는 H/W와 S/W의 결합으로, 다양한 기능을 제공해 주는 공통 실행환경을 일컫는다[3]. 본 논문에서는 그림 1에서 보는 바와 같이 다양한 방식의 자율주행차의 운행 데이터를 입력받아 데이터를 분류하고 가공한 후 자율주행차로 데이터를 출력하거나 사용자가 원하는 서비스를 제공하는 것으로써 소프트웨어 플랫폼을 말하며, 본 논문에서는 'VBS (Vehicle Bridge Stack)'라 명명한다.

1) 데이터 정의

데이터 취득 과정은 자율주행 플랫폼에서 제공가능한 데이터를 기준으로 표 1과 같이 정의한다. 이기종 플랫폼마다 데이터 종류가 달라질 수 있기 때문에 커스터마이징(customizing)이 필요하다[4].

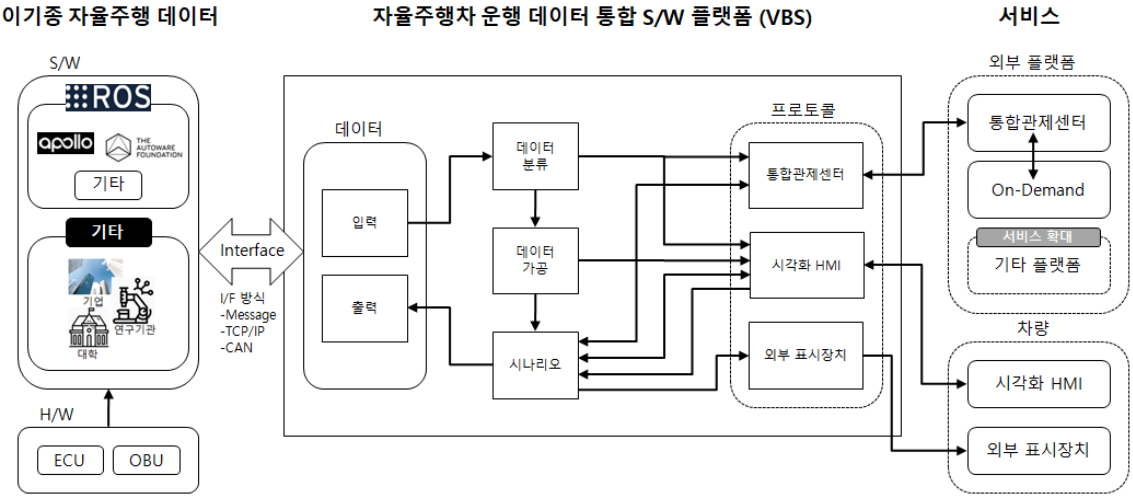


그림 1. 통합 소프트웨어 플랫폼(VBS)의 구성도

표 1. 데이터 정의

구분	항목	세부항목
입력	차량 정보	속도, GPS, 기어상태(P/R/N/D), 장비상태(GPS, 카메라, 라이다, OBU, CAN, ACU), 자율주행모드 상태(자율주행, 조향제어, 가/감속)
	환경 정보	주변인지차량정보(8대 인지), 차선정보, 교통신호정보
출력	설정 모드	HMI 연동 활성화, 속도제한 설정, 자율주행모드(자율주행, 조향제어, 가/감속)

2) 통신 정의

본 연구에서는 ROS (Robot Operating System)[4, 5] 기반의 자율주행 플랫폼 내에 VBS를 미들웨어 형태로 내장하고 있기 때문에 Message 방식으로 인터페이스 하였다. VBS를 플랫폼 내에 탑재할 수 있다면 데이터 교환에 효율적인 Message 방식으로 인터페이스 함으로써 통신에 필요한 H/W 부가 장비에 대한 비용 절감, 통신 전송속도, 통신장애, 확장성의 측면에서 유리하다. 플랫폼 특성상 내장하기 어렵다면 VBS를 임베디드 형태의 단말기에 탑재시켜 플랫폼이 지원하는 TCP/IP 또는 CAN 방식의 통신 프로토콜을 설계하고 플랫폼과의 인터페이스 한다.

3) 데이터 입력 및 출력

자율주행 플랫폼의 입력 데이터와 자율주행차 설정정보의 출력 데이터는 ROS 메시지 형식으로 처리한다.

4) 데이터 분류

입력된 데이터는 통합관제센터, 수요응답형(On-Demand), 시각화 HMI, 외부 표시장치 등에 적절한 서비스를 제공하기 위해 분류하고 저장한다.

5) 데이터 가공

보다 확장된 서비스를 제공하기 위해서는 입수 데이터를 가공할 필요가 있다. 본 연구에서는 차량의 좌·우회전 방향정보 계산, 전방 30m 이내의 신호등의 정보를 HMI를 통해 시각화하여 표시하였다.

6) 시나리오 설정

통합관제와 연계된 On-Demand 서비스 중에는 사용자가 예약하면 운전자가 HMI를 통해 승인하는 절차가 있다. VBS에서는 서비스 업체와 협의된 절차를 시나리오에 반영하여 예약/승인/취소 등의 프로세스를 관리한다.

HMI로는 자율주행차 제어모드 설정정보가 입력되면 상태를 저장하고 차량내 다른 서비스 단말기와 정보를 동기화하여 관리한다.

외부 표시장치인 전광판에 표시되는 알림문구는 시나리오에서 정의한 형태로 표현한다.

7) 프로토콜

통신 인터페이스 확장성을 고려하여 Socket[6] 방식으로 통신한다. 서비스가 추가되어도 데이터의 중복성에 따른 프로토콜 재사용 설계가 가능하고 유/무선의 유연한 통신이 가능하다.

8) 서비스 종류

차량 내부에서 사용하는 서비스로는 운전자와 승객이 자율주행 정보를 쉽게 확인하고 조작할 수 있는 시각화 HMI가 있고, 차량 외부 표시장치를 통해서는 자율주행차 주변에서 차량의 정보를 확인할 수 있다. 외부 플랫폼과 연계되는 서비스는 자율주행차의 데이터를 취득하여 모니터링하는 통합관제센터, 통합관제센터와 연계하여 자율주행차를 탑승예약하는 서비스(On-Demand) 등 다양한 서비스 플랫폼과의 확장이 가능하다.

2.2 통합 S/W 플랫폼(VBS)의 구조

VBS의 구조는 그림 2와 같이 자율주행 플랫폼(Self-driving platform), 커스터마이징(Customizing), 코어 (Core), 프로토콜(Protocol), 서비스 (Service)의 5개의 레이어로 구성되어 있다.

1) 자율주행 플랫폼

이기종의 자율주행 플랫폼으로는 기업, 대학, 연구기관에서 개발 중인 각각 다른 플랫폼과 ROS 기반의 오픈 플랫폼인 아폴로, 오토웨어 등이 있다.

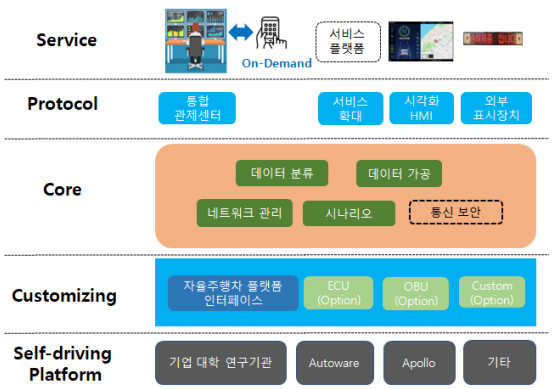


그림 2. 통합 S/W 플랫폼(VBS)의 구조

2) 커스터마이징

자율주행 플랫폼의 데이터를 입출력할 경우 데이터를 정의하고 인터페이스를 커스터마이징 한다. 그리고 옵션 항목인 ECU, OBU, Custom 등은 플랫폼 확장성을 고려하여 설계하였다.

3) 코어

핵심 로직인 데이터 분류 및 가공을 처리하고, 플랫폼의 시나리오 정의 및 관리, 네트워크 설정 관리, 플랫폼 설정 등의 기능을 담당한다. 점선 블록인 통신보안은 향후 고려할 데이터 보안을 나타낸다.

4) 프로토콜

다양한 서비스에 대응할 수 있도록 통신방식은 Socket으로 단일화하였다. Message Set은 JSON, MQTT 등의 다양한 구성이 가능하도록 확장성 있게 설계하였다.

5) 서비스

본 연구에서 적용한 통합관제, 관제와 연계된 On-Demand, 시각화 HMI, 외부 표시장치 서비스뿐만 아니라 새로운 서비스 플랫폼과의 연계를 통해 확장된 서비스를 제공할 수 있다.

3. 인터페이스 설계

이 장에서는 데이터 입력부터 서비스되는 과정을 시각화 HMI와 외부 표시장치 위주로 설명한다.

3.1 입력 데이터 인터페이스

자율주행차 운행 데이터는 VBS로 100ms 주기로 입력되며, 그림 3의 ROS 'topic' 정보와 같이 메시지의 헤더 정보와 정의된 데이터 형식으로 되어 있고 메시지를 'subscribe'(수신)하여 입력 데이터를 취득한다.

3.2 서비스에 필요한 데이터 가공 사례

입력 데이터인 현재 속도(ego_speed)와 자율주행 경로정보(local_poly)를 이용하여 차량의 좌·우회전에 대한 방향정보를 계산한다. 자율주행차 좌표계를 x축은 차량의 진행방향, y축은 자차 기준 횡방향으로 정의할 때, 목표 곡선경로를 다음의 다항식으로 표현할 수 있다.

$$y = a_2x^2 + a_1x + a_0$$

따라서 2초 후 차량의 방향은 x=2×ego_speed 일 때, 그림 4와 같이 y가 (+)부호이면 좌회전, (-)부호이면 우회전으로 판단한다.

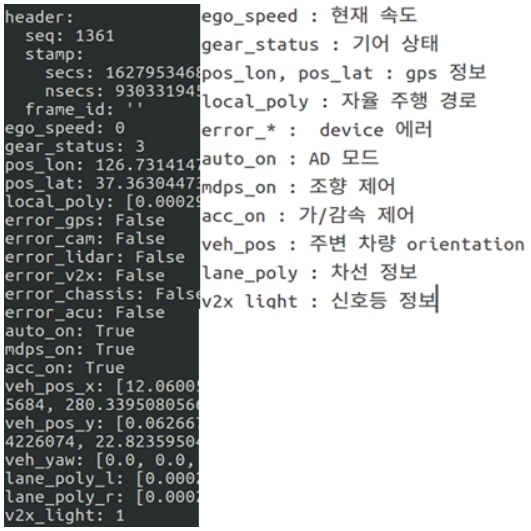


그림 3. 입력 데이터 topic 화면 및 데이터 설명

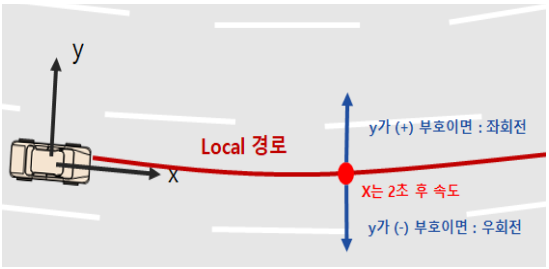


그림 4. 차량 좌·우회전 계산 방법

3.3 VBS 에서 HMI 서비스 데이터 전송

표1에 열거된 입력된 차량 운행정보와 환경정보와 함께 VBS에서 계산한 경로커브(방향), 관제연결상태, On-Demand 예약 등의 추가정보를 JSON 메시지 포맷으로 변환하여 Socket 통신으로 HMI 단말기에 전송한다. HMI는 전송받은 데이터로 차량의 운행정보를 그림으로 표시한다.

3.4 HMI에서 자율주행차 제어설정 설정

HMI 화면에서 자율주행 제어모드인 전체자율모드, 조향자율모드, 엑셀자율모드의 선택과 자율주행의 속도제한의 설정을 할 수 있다. 이 설정정보를 JSON 형식으로 조합해서 Socket 통신으로 VBS에 전송한다. 표 2는 HMI에서 VBS로 입력되는 설정정보를 보여준다.

표 2. HMI에서 VBS로 입력되는 데이터

구분	항목	세부항목
출력	차량 정보	자율주행모드 설정(자율주행, 조향제어, 가/감속), 속도제한 설정
	환경 정보	HMI 연동 활성화, 운행 노선 번호, 로직 초기화, 예약 승인, 승객 탑승여부

3.5 자율차에 제어 설정정보 전송

전송받은 HMI의 자율주행 제어설정 정보를 시나리오 모듈에서 제어상태를 저장하고 그림 5와 같이 자율주행 플랫폼에 ROS 메시지 방식으로 전송한다. 그리고 저장된 제어상태 정보로 차량내 다른 단말기와 제어설정 정보를 동기화한다.

```
header:
  seq: 22
  stamp:
    secs: 0
    nsecs:
  frame_id: ''
hmi_on: True
vx_set: 46
vx_set_trigger:
route_index: 2
route_trigger:
initialize_on:
auto_on: True
mdps_on: True
acc_on: True
error_hmi: Fals
```

hmi_on : hmi 연동 활성화
vx_set : 자율주행 최대 속도 설정
route_index : 노선 번호
route_trigger : 메시지 신호 trigger
initialize_on : 자율주행 로직 초기화
auto_on : 자율주행 모드
mdps_on : 조향 모드
acc_on : 가감속 모드
error_hmi : hmi 작동 에러

그림 5. 출력 데이터 topic 화면 및 데이터 설명

3.6 외부 표시장치

자율주행차의 외부 전광판에 정의된 시나리오는 표 3과 같다. 운행 및 안전에 필요한 정보를 외부 전광판 컨트롤러에 Socket 통신으로 전송하면 알림문자를 출력한다.

표 3. 전광판 시나리오

운행모드	메시지
평상시 (무조작)	자율주행 운행 중입니다. 안전운전 바랍니다.
속도 감속시	감속 중입니다.
비상등 점멸시	안전운전
좌측 방향지시등	◀◀좌측이동
우측 방향지시등	우측이동▶▶
문열림 시	승하차 중입니다.

4. 시험 방법 및 결과

4.1. 장치 구성

장치 구성은 그림 6과 같이 노트북, 라우터(유무선 통신 환경), 안드로이드 패드(시각화 HMI APP), 전광판(외부 표시장치)로 구성되어 있다. 노트북은 실증 자율주행차 운행정보가 기록되어 있는 'rosbag'을 재생하고, VBS를 실행하면 서비스 데이터를 입수한다. 라우터는 네트워크망을 구성한다. 유선(LAN)으로는 인터넷망, 노트북, 전광판을 연결하고 무선(WiFi)은 안드로이드 패드를 연결한다. IP는 장비에 직접 통신이 가능하도록 고정 IP로 할당한다. 안드로이드 패드는 시각화 HMI App을 실행하고 VBS의 데이터를 시각화하고 제어설정 시 VBS에 설정된 정보를 전송하는 기능을 한다. HMI App은 VBS 설계 내용을 기반으로 개발하였다. 전광판은 VBS의 시나리오에 정의된 문자가 수신되면 해당 문자를 전광판에 표시한다.

시험절차는 'rosbag'을 재생하고 VBS를 실행하면 자율주행 운행 데이터가 입력되고 서비스에 해당하는 데이터를 전송하게 된다. HMI에서는 자율주행차의 운행정보를 시각화한 영상을 볼 수 있고, 전광판에서는 자율주행 알림문자를 확인할 수 있다.

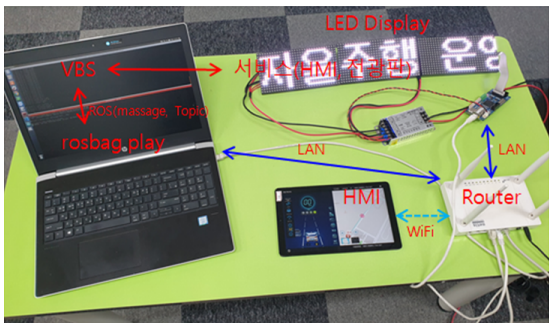


그림 6. 장치 구성

4.2 시험 방법

시험절차는 그림 7과 같이 'rosbag'을 재생하고 VBS를 실행하면 자율주행 운행 데이터가 입력되고 서비스에 해당하는 데이터 전송하게 된다. 본 연구에서는 시각화 HMI와 외부표시장치 서비스 위주로 설명 한다.

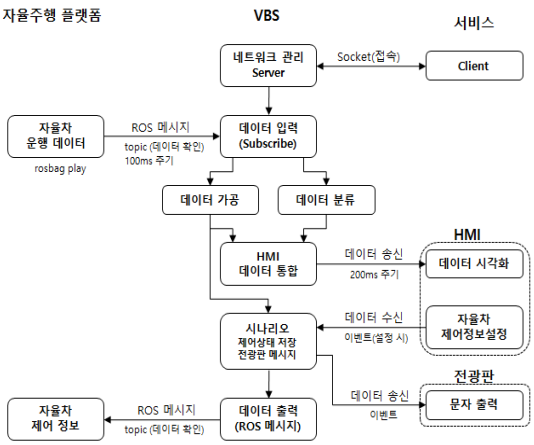


그림 7. 데이터 처리 과정 및 통신

1) 운행 데이터 입력

자율주행차 운행정보가 기록되어 있는 'rosbag'을 재생하고 'topic'으로 메시지를 확인해 보면 운행 데이터가 100ms 간격으로 주기적으로 업데이트되고 있음을 확인할 수 있다. 메시지 내용을 수신(subscribe)하여 입력정보를 저장한다.

2) HMI 서비스

좌·우회전에 대한 데이터 처리가 필요하여 데이터가공하여 계산된 방향정보를 취득하고 입력데이터의 HMI에 필요한 데이터를 JSON 포맷으로 조합하여 200ms 주기로 송신한다. 200ms 주기로 한 이유는 App에서 자율차 주변 GUI(원근법 처리, 주변 차량 표현, 자차 효과)를 표현할 수 있는 시간을 고려해서 설정하였다. HMI App에서 수신된 운행 데이터를 바탕으로 지도상의 차량 위치, 센서 상태, 현재 속도, 자차주변 차량 위치, 좌/우 방향, 기어, 자율주행 모드 상태, 제한 속도 등을 시각적으로 확인 된다. HMI에서 자율주행차 제어설정에 대한 설정을 할 경우 제어설정 정보를 JSON 형식으로 VBS에 송신한다. VBS에서 수신된 제어설정은 시나리오 모듈에 저장하고 ROS 메시지로 자유주행 플랫폼으로 전송한다.

3) 전광판 서비스

표 3과 같이 시나리오에 정의된 알림문구를 전광판 컨트롤러에 전송하면 해당 문자를 전광판에서 확인할 수 있다.

4.3 실장시험 및 결과

시흥시 배곧 신도시에서 자율주행차량 아이오닉 6대, e카운티 1대를 매일 특정 시간대에 일반 시민을 대상으로 수요응답형 On-Demand 서비스를 하고 있다. 승객이 탑승하면 좌석에 앉아 HMI 화면을 보게 된다. 기존의 HMI는 시각화와 비즈니스(business) 로직을 모두 수행하는 Full-stack HMI로써 차량 운행 중 진동과 충격의 원인으로 단말기 장애, 통신 불량, 케이블 결선 불량 등의 잦은 고장으로 사용할 수 없는 상태가 되어 개선이 필요하였다. 따라서 본 연구에서는 기능과 서비스 확장성을 고려하여 HMI는 주로 시각화와 간단한 처리만을 담당하고, VBS가 비즈니스 로직을 수행하는 Front-end HMI 방식으로 접근하여 소프트웨어 플랫폼, VBS를 설계하게 되었다.

Full-stack HMI를 Front-end HMI와 VBS라는 2-tier로 구현함에 따라 그림 8과 같이 통신과 제어 케이블의 수는 11개에서 4개로, H/W 모듈도 12개에서 5개로, 프로토콜은 5종에서 3종으로 단순화하고, 통신 종류는 5종으로 단일화할 수 있었다. 반면 서비스는 차량 내에서만 제공하던 것에서 외부 서비스인 관제서버연계, On-Demand 예약처리까지 4개로 쉽게 확장이 가능하였다.

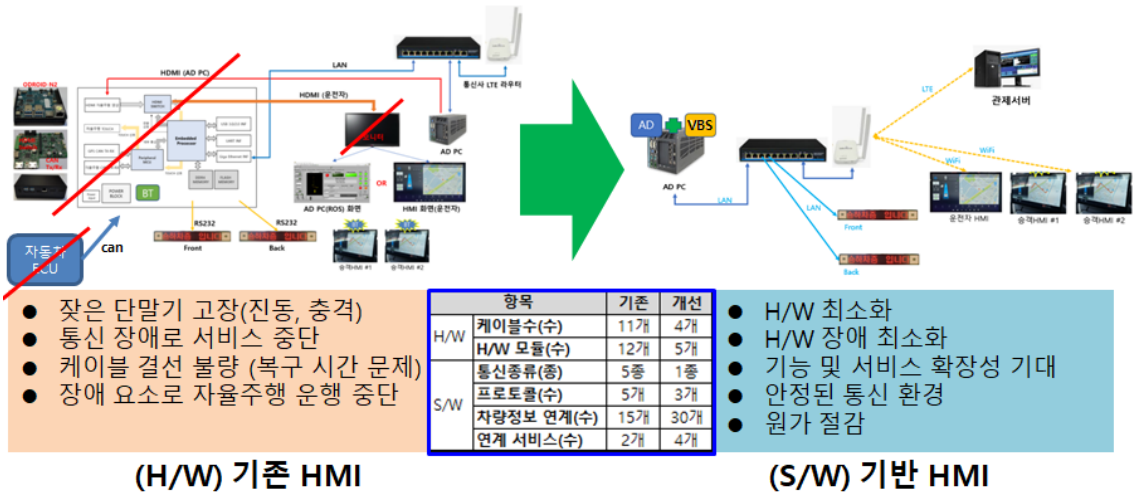
그리고 S/W와 통신의 신뢰성을 확보하기 위하여 표 4와 같은 Aging 시험을 진행하였으며, 이때 발생한 S/W 멈춤이나 통신 끊김의 문제를 보완하여 안정된 품질을 확보할 수 있었다.

표 4. 신뢰성 시험 방법

시험 항목	시험 방법
Aging	통신주기 - 200ms 주기 데이터 (량) - 약 400 Bytes 테스트 시간 - 12시간 (4회)
통신 재연결	테스트 단말기 - 3대 테스트 주기 - 비주기 단말기 조작 방법 - 교차 단말의 App 종료 - 시작 시 재연결 확인

VBS를 그림 9와 같이 실증 차량 7대에 적용·시험한 결과 차량 내·외부에서 다양한 서비스가 안정적으로 이루어지는 것을 확인하였다.

오퍼레이터(운전자)는 HMI 패드 하나로 차량의 상태를 확인하고 자율주행차의 제어모드를 설정하며, 예약자 승인 처리도 가능하다. 승객은 뒤좌석 패드로 차량의 운행상태를 실시간 확인할 수 있다.



	항목	기존	개선
H/W	케이블수(수)	11개	4개
	H/W 모듈(수)	12개	5개
S/W	통신종류(종)	5종	1종
	프로토콜(수)	5개	3개
	차량정보 연계(수)	15개	30개
	연계 서비스(수)	2개	4개

그림 8. VBS을 적용하여 시험한 결과 및 효과

차량 외부 전광판으로는 주위 차량과 보행자에게 운행 및 안전에 대한 알림문구를 출력한다. 통합관제센터에서는 7대의 차량위치 정보를 확인할 수 있고, 탑승하려는 승객은 On-Demand App으로 예약하고 승인절차 후 차량이 도착하면 이용할 수 있다.

그림 9. 자율주행차에 실증 실험

VBS는 S/W 모듈로써 이식성이 뛰어나 자율주행차 플랫폼 내에 탑재하거나 임베디드 형태로 배포할 수 있기 때문에 손쉽게 다양한 서비스를 활용할 수 있다.

5. 결론

본 연구를 통해 이기종 자율주행차의 운행 데이터를 수집하고 처리하여 사용자에게 다양한 서비스를 제공할 수 있는 통합 소프트웨어 플랫폼을 제시하였다. 기존의 시스템들이 많은 H/W 모듈, 다양한 통신방식과 프로토콜이 혼재되어 인터페이스가 어렵고 서비스 확장성에 문제가 있었으나, 본 연구를 통해 이기종 자율주행차의 운영 데이터의 입출력 및 관리와 통합관제센터, On-Demand, HMI 시각화 등의 서비스 이용이 원활한 것으로 나타났다.

뿐만 아니라, 자율주행 레벨4, 레벨5[7]를 대비해야 하는 관점에서 본 연구는 앞으로 완전자율주행을 위한 지역제어센터(Local Control Center, LC)와 연계하여 인프라 정보 연결 및 비상시 원격제어를 위한 인터페이스가 가능하다[8].

향후 본 연구를 기반으로 자율주행차 상용화 패키징 기술, 자율주행 플랫폼의 특정 기능의 보조 및 대체 기술, 데이터 보안 등의 연구를 진행할 예정이다. 또한 본 플랫폼의 서비스 확장을 위해 AI 기술을 활용한 차량 주변 정보를 수집·분석하는 서비스 모듈을 개발할 계획이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성 (Grand ICT연구센터) 사업의 연구결과로 수행되었음 (IITP-2021-2020-0-01462)

참 고 문 헌

[1] 백장균, "자율주행차 국내외 개발 현황," 산은조사월보, 제771호, pp. 17-36, 2020년 2월.
[2] 최원석, 최성곤, "자율주행자동차 기술 동향에 관한 연구", 컴퓨터정보통신연구소 논문지, 제29권 제1호, pp. 27-34, 2021년 5월.
[3] 이경희외 5인, "웹기반 SW 플랫폼 기술동향", 전자통신동향분석, 제26권 제5호, pp. 66-73, 2011년 10월.
[4] ROS.org, http://wiki.ros.org
[5] 표윤석, 조한철, 정려운, 임태훈, ROS 로봇 프로그래밍(개정증보판), 루비페이퍼, 부천, 2017
[6] James F. Kurose, Keith W. Ross, Computer Networking - A Top Down Approach(7th), PEARSON, 2016.
[7] 손주찬, 최정단, "자율주행자동차 시스템 기술 방향과 과제". 한국통신학회지(정보와통신), 제35권 제5호, pp. 21-27, 2018년.
[8] 문영준. "자율주행 기반 스마트 모빌리티". 방송과미디어, 제24권 제1호, pp. 49-55, 2019.

저 자 약 력

임동민
〈dmlim@automos.co.kr〉
2007년 평생교육진흥원 컴퓨터과학과 학사
2021년 충북대학교 산업인공지능학과 석사과정
〈주 관심분야 : 자율주행 및 데이터허브〉

김현용
〈kimhy365@cbnu.ac.kr〉
1996년 서울대학교 농공(기계)학과 학사
1998년 서울대학교 농공(기계)학과 석사
2018년 충북대학교 스마트팩토리학과 박사수료
2021년 충북대학교 산업인공지능학과 초빙교수
〈주 관심분야 : 스마트팩토리 및 머신비전〉

김곤우
〈gwkim@cbnu.ac.kr〉
2006년 서울대학교 전기컴퓨터공학부 박사
2006~2008년 한국생산기술연구원 로봇기술본부 위촉연구원
2008~2012년 원광대학교 전기전자 및 정보공학부 조교수
2012년~현재 충북대학교 지능로봇공학과 교수
〈주 관심분야 : 자율주행 및 지능로봇〉