

OBD 차량 진단 정보를 위한 IoT 장치 구현

이 성 희*, 이 성 형*, 이 상 문**, 황 승 훈^o

IoT Equipment Implementation for OBD Car Diagnostic Information

Seong-Hee Lee*, Seong-Hyung Lee*, Sang-Moon Lee**, Seung-Hoon Hwang^o

요 약

차량의 OBD 커넥터로부터 획득한 차량 정보를 스마트기기와의 통신을 통해 확인하는 기존의 제품은 차량 내 혹은 차량의 근처에서만 통신이 가능할 뿐 외부로 전송하는 기능은 지원하지 않는다. 본 논문에서는 기존의 제품과 달리 OBD를 통해 얻은 차량 정보를 외부로 전송하여 서버에서 관리할 수 있는 OBD 차량 진단 정보 통신을 위한 IoT 장치의 구현에 관해 기술한다.

Key Words : On-board diagnostics, V2X, Wi-Fi, M2M, IoT

ABSTRACT

Existing devices are capable of communicating the OBD information only inside or close to the vehicle without supporting the data transmission to an external server. In this paper, we describe the implementation of IoT device, which can communicate the OBD information to the external server.

I. 서 론

통신을 이용하여 차량의 OBD로부터 얻은 정보를 전송하는 제품들은 이미 여러 개 존재한다. 이러한 제품들은 OBD 정보를 읽기 위해 블루투스를 이용하여 스마트폰 및 태블릿 PC 등과 통신하여 자동차 상태 정보를 확인하는 방식이다. 하지만 이러한 제품들은 OBD 정보를 차량 내의 스마트기기 간 통신만 지원할 뿐 외부로 전송하는 기능은 지원하지 않고 있다. 따라서 본 연구를 통해 기존의 개발된 제품들의 방식과 달리 차량 내 OBD 정보를 공공재 AP망을 이용하여 외부로 전송이 가능하도록 하여 개인뿐만 아니라 자동차 회사, 보험회사에 해당 정보를 제공할 수 있는 장치를 구현하여 그 결과를 고찰하고자 한다. 본 논문에서

서는 전체적인 시스템 디자인, CAN통신과 OBD에 대한 간단한 개념, 연구에 사용된 하드웨어 및 소프트웨어, 세부적인 기술개발 내용, 실험 결과 및 분석에 대해 다뤄 보고자 한다.

II. 전체적인 시스템 디자인

본 연구의 전체적인 시스템은 그림 1과 같다. 차량 정보를 OBD 모듈을 통해 획득하고 시리얼통신을 통해 Wi-Fi 모듈로 전송한다. Wi-Fi 모듈은 무선통신을 이용하여 외부AP로 전송하고, AP와 연결된 서버에서 이 정보를 수신하여 확인 및 관리하는 것이 본 연구의 전체적인 시스템이다. 본 연구에서 사용된 차량 OBD 정보는 RPM, 속도, 이동거리이며, 연구는 그림 1의

* First Author : Div. of Electronics & Electrical Eng, Dongguk University, blackbill257@naver.com, 학생회원

^o Corresponding Author : Div. of Electronics & Electrical Eng, Dongguk University, shwang@dongguk.edu, 종신회원

* Div. of Electronics & Electrical Eng, Dongguk University, 정회원

** NT Robot, smlee@ntrobot.net, 정회원

논문번호 : KICS2016-11-336, Received November 1, 2016; Revised November 14, 2016; Accepted November 14, 2016

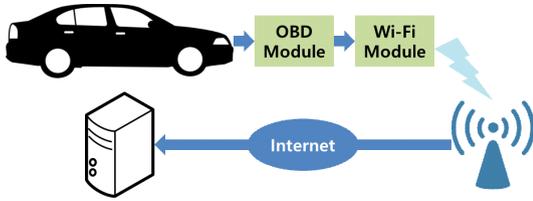


그림 1. 전체적인 시스템 디자인
Fig. 1. Overall system design

화살표 방향으로 순차적으로 진행하였다.

III. CAN 통신과 OBD

본 장에서는 본 연구에 이용된 기본지식인 CAN통신과 OBD에 대해서 간략히 살펴보고자 한다.

3.1 CAN 통신

CAN 통신 (혹은 CAN bus)은 Controller Area network의 약자로 자동차용 네트워크의 표준이며, 차량 내에서 호스트 없이 ECU등의 마이크로 컨트롤러나 장치들이 서로 통신하기 위한 멀티마스터 방식으로 설계되었다¹¹. CAN 통신은 메시지 기반의 프로토콜이며 한 개의 데이터링크를 기준으로 여러 개의 노드들이 하위에 연결되어있는 Serial BUS 시스템이다. CAN 통신의 대표적인 특징으로는 잡음에 강하고, 실시간 메시지 통신이 가능하고, 통신에 우선순위가 있고, 하드웨어적으로 설정된 ID만 선택하여 수신 가능하다.

3.2 OBD

OBD는 On Board Diagnostics의 약자로 차량 자가 진단 및 보고를 의미하는 차량용어로서, 차량의 다양한 장치의 상태를 운전자와 자동차 엔지니어에게 제공한다. OBD 시스템과 CAN 통신의 관련성에 대해 살펴보면, 차량내부통신인 CAN 통신을 통해 차량의 센서들로부터 ECU로 전달된 자동차 주요 계통 정보나 기능고장 정보를 OBD 시스템에 의해 외부장치에서 볼 수 있게 한다. 표 1은 OBD의 차량 진단 모드이다^{2,3}.

IV. 제작된 H/W와 사용된 S/W

본 연구를 위해 제작한 하드웨어는 OBD모듈과 Wi-Fi모듈이다. 본 장에서는 각 모듈의 간단한 기능과 구성요소에 대해 살펴보고, 자세한 동작원리에 대해서는 다음 장에서 다루도록 한다.

표 1. OBD 차량 진단 모드

Table 1. OBD vehicle diagnostic mode

Mode	Description
MODE 1	Show current data
MODE 2	Show freeze frame data
MODE 3	Show stored Diagnostic Trouble Codes
MODE 4	Clear Diagnostic Trouble Codes and stored values
MODE 5	Test results, oxygen sensor monitoring
MODE 6	Test results, other component/system monitoring
MODE 7	Show pending Diagnostic Trouble Codes
MODE 8	Control operation of on-board component/system
MODE 9	Request vehicle information

4.1 OBD 모듈

OBD 모듈은 OBD 컨넥터에 연결되어 차량의 OBD 정보를 획득하고, 획득한 정보를 시리얼통신을 통해 Wi-Fi 모듈로 전송하는 기능을 한다. OBD 모듈의 사진과 구조는 그림 2, 3과 같고, 각 구성요소의 기능은 표 2에 정리되어 있다.

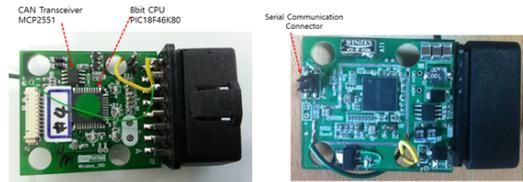


그림 2. OBD모듈의 전면부와 후면부
Fig. 2. Front and back side of OBD module

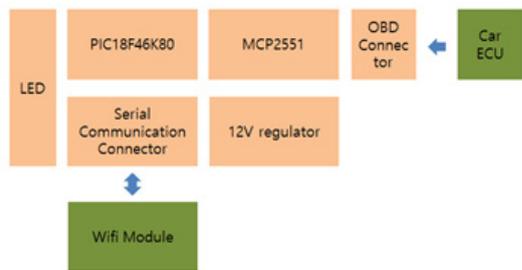


그림 3. OBD 모듈의 구조
Fig. 3. Structure of OBD module

4.2 Wi-Fi 모듈

Wi-Fi 모듈은 OBD 모듈로부터 차량 OBD정보를 수신받고, 수신받은 정보를 무선통신을 통해 AP로 전송하는 기능을 한다. Wi-Fi 모듈의 사진과 구조는 그

표 2. OBD 모듈의 주요구성 및 기능
Table 2. Main configuration and function of OBD module

Configuration	Function
PIC18F46K80	8 bit CPU. Interprets CAN signal as OBDII protocol, extracts necessary information and transmits it to Wi-Fi module
MCP2551	Since the CAN signal of the CPU used inside the board operates at 3.3V, it changes the signal strength that can communicate with the vehicle
12V regulator	Change the voltage to be used on the board
LED	Display the status of communication with vehicle via CAN interface

림 4, 5와 같고, 각 구성요소의 기능은 표 3에 정리되어 있다.

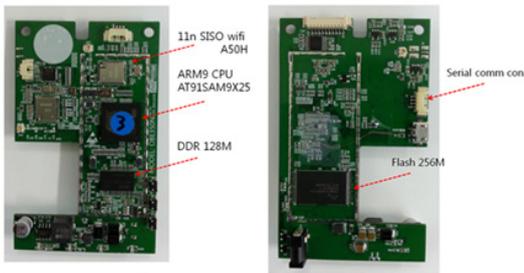


그림 4. Wi-Fi모듈의 전면부와 후면부
Fig. 4. Front and back side of Wi-Fi module

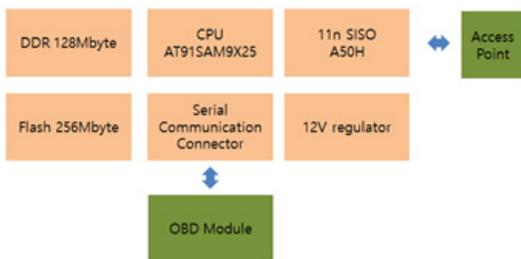


그림 5. Wi-Fi 모듈의 구조
Fig. 5. Structure of Wi-Fi module

표 3. Wi-Fi 모듈의 주요구성 및 기능
Table 3. Main configuration and function of Wi-Fi module

Configuration	Function
AT91SAM9X25	32bit ARM9 CPU. Embedded linux ported to work, Wi-Fi driver and OBD communication protocol ported

11n SISO	Wi-Fi module. It supports the 802.11n protocol. Supports up to 75Mbps in SISO
12V regulator	Change the voltage to be used on the board
DDR 128Mbyte	Volatile memory. 128Mbyte
Flash 256Mbyte	Non-volatile memory. 256Mbyte

4.3 사용된 S/W

OBD모듈의 CPU에 포팅되어 있는 코드는 MPLAB이라는 C언어 기반의 프로그램을 이용하여 제작 및 포팅하였다. Wi-Fi 모듈의 CPU에 포팅되어 있는 코드는 임베디드 C를 이용하여 제작 및 포팅하였다.

V. 세부연구사항

본 연구를 진행함에 있어, 상기 그림 1의 화살표방향으로 순차적으로 진행하였으며, 본 장에서는 각 과정의 연구과정 및 결과에 대해 자세히 살펴보고자 한다.

5.1 OBD 모듈을 통한 차량 OBD정보 획득

차량의 OBD 정보를 획득하기 위해 OBD 모듈을 사용하고, 제대로 된 정보가 획득했는지 확인하기 위해 Teraterm 이라는 터미널 프로그램을 사용하였다. OBD 모듈을 차량의 OBD 단자에 연결하고 시동을 걸면 OBD 모듈에 전원이 공급되면서 작동을 시작한다. 모듈은 CPU에 포팅된 코드에 의해 엔진 RPM와 차량의 속도정보를 차량으로부터 획득하고, 이동거리는 $S = v * t$ 라는 공식에 의해 얻어지게 된다. 3가지 차량 OBD 정보를 구분하기 위해 RPM, SPEED, DIS라는 변수를 사용하였다. 실험은 시동만 켜놓는 경우, RPM만 증가시키는 경우, 차량을 이동시키는 경우로 나누어 진행하였다. 실제로 시뮬레이션을 진행한 결과 분석은 그림 6과 같다.

그림 6에서 보는 것과 같이, 시동만 켜놓는 경우 RPM은 차량의 기본 RPM값을 가지고 속도와 이동거리는 변화가 없는 것을 확인할 수 있다. RPM만 증가시키는 경우 차량의 계기판과 같이 RPM이 증가하고 속도와 이동거리는 변화가 없다. 마지막으로 차량을 운전하는 경우 3가지 정보가 모두 바뀐 것을 확인할 수 있다^[4,5].

5.2 기기간 시리얼 통신을 통한 OBD 정보 교환

기기간 시리얼 통신을 통한 OBD 정보의 작동을 확인하기 전, 우리는 먼저 시리얼통신의 정상 작동 여

RP1 = 712A SPEED = 0A DIS = 22A	사용한 경우라면 아무변화가 없다
RP1 = 712A SPEED = 0A DIS = 22A	
RP1 = 712A SPEED = 0A DIS = 22A	
RP1 = 726A SPEED = 0A DIS = 22A	기어를 P에 놓고 가속페달만 밟아 RPM이 증가시키는 구간 (속도, 이동거리는 변화가 없다)
RP1 = 750A SPEED = 0A DIS = 22A	
RP1 = 1462A SPEED = 0A DIS = 22A	
RP1 = 1525A SPEED = 0A DIS = 22A	
RP1 = 1112A SPEED = 4A DIS = 22A	
RP1 = 926A SPEED = 6A DIS = 23A	가속페달을 밟지 RPM과 속도, 이동거리가 올라가는 것을 확인할 수 있다.
RP1 = 1212A SPEED = 6A DIS = 23A	
RP1 = 1387A SPEED = 8A DIS = 24A	
RP1 = 1112A SPEED = 8A DIS = 25A	
RP1 = 1012A SPEED = 8A DIS = 25A	
RP1 = 962A SPEED = 6A DIS = 26A	
RP1 = 876A SPEED = 6A DIS = 26A	
RP1 = 812A SPEED = 4A DIS = 27A	
RP1 = 687A SPEED = 2A DIS = 27A	
RP1 = 762A SPEED = 0A DIS = 27A	
RP1 = 737A SPEED = 0A DIS = 27A	

그림 6. 시뮬레이션 결과 분석
Fig. 6. Simulation results analysis

부를 확인하기 위해 OBD 모듈 2개를 이용하여 송신단에서는 OBD 정보를 전송하고, 수신단에서는 Echo 코드를 포팅하여 수신한 정보를 그대로 다시 보내도록 설계하였다. 수신측 OBD모듈에 시리얼포트를 연결하여 컴퓨터로 데이터를 확인하여 시리얼 통신의 작동여부를 확인하였다. Teraterm을 통해 차량의 OBD정보를 컴퓨터로 확인하는 것은 C언어 기반의 Sprintf라는 함수를 이용하여 구현하기 때문에 큰 문제가 되지 않지만, 기기 간 통신을 할 경우 송신측에서 취득한 OBD정보를 그대로 전송하게 되면 수신측에서는 데이터가 RPM정보인지, 속도정보인지 이동거리 정보인지 구분할 수가 없다. 따라서 TLV(Type, Length, Value) Protocol을 적용시켜 세가지 정보를 구분하였고, 적용시킨 Protocol frame구조는 아래 표 4와 같다. 3가지 정보는 표 5와 같이 frame^[2]의 Opcode로 구분을 하며 각각의 데이터는 frame^[3]-frame^[6]에 채워져 전송된다. F0와 E0를 통해 각각의 정보에 대한 frame의 시작과 끝을 구분하며 Length는 5로 3가지 경우의 데이터에 대해 동일하다 [6,7].

그림 7은 수신측이 수신한 정보를 그대로 전송하는 데이터를 캡처한 그림으로 표 4의 frame구조와 동일하게 전송되는 것을 확인할 수 있다. 시리얼 통신의 정상 작동 유무를 확인하기 위해 실시된 이 실험의 CPU의 성능이 좋지않은 OBD 모듈을 사용하였기 때

표 4. TLV 프로토콜 프레임 구조
Table 4. Structure of TLV Protocol frame

SYNC	frame[0]	frame[1]	frame[2]
0xF0	0x91 CMD	0x05 Length	Opcode
frame[3]-[6]	cs		EOF
DATA	check sum	0xE0	

표 5. Opcode 종류
Table 5. Opcode type

Type	RPM	SPEED	DISTANCE
Opcode	0x00	0x01	0x02

문에 한번에 3가지정보를 수신 받아 처리할 수 없다. 왜냐하면 한번에 30byte를 전송 받아 처리하게 되면 수신부에서 과부하가 발생하여 정상적으로 수신이 되지 않기 때문이다. 따라서 송신부에서 OBD 정보를 송신할 때 delay를 주어 한번에 1 byte 씩 전송함으로써 수신부의 과부하를 줄여 정상적으로 수신되도록 하였다. 한번에 1byte씩 전송하더라도 수신부에서는 전체적으로 frame구조가 유지되며 3가지 정보가 순서대로 정상적으로 수신되는 것을 확인할 수 있다.

실제로 OBD 모듈과 Wi-Fi 모듈간의 시리얼통신에서는 Wi-Fi 모듈의 CPU성능이 훨씬 좋기 때문에 한번에 30byte를 처리할 수 있으므로 송신부에서 별도의 delay를 적용시키지 않았다.



그림 7. 전송된 데이터
Fig. 7. Transmitted data

5.3 Wi-Fi 모듈을 이용한 무선통신

그림 8은 자동차로부터 획득한 OBD정보를 OBD 모듈과 Wi-Fi모듈을 거쳐 AP까지 전송하는 과정을 나타낸 그림이다.

Wi-Fi 모듈의 작동원리를 살펴보면, 먼저 UDP 소켓설정을 통해 서버와 미리 정의한 통신을 설정한다. 실제로 포팅된 코드에 의해 서버와 UDP 통신으로 정의하고 서버 IP는 192.168.0.151, 포트번호는 15151로 정의 하였다. Wi-Fi모듈은 station으로설정되어 있고 미리 설정된 SSID와 암호화에 따라 일반 AP와 연결되도록 설정하였다. 실험에 사용된 AP는 iptime에서 제조한 일반 상품을 사용하였다. 소켓설정이 끝나면 시리얼 통신 설정을 통해 OBD 모듈로부터 데이

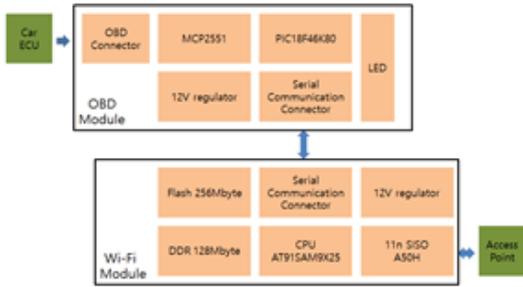


그림 8. 정보 전송 과정
Fig. 8. Process of information transfer

터 수신하기 위한 설정을 진행한다. 설정이 끝나면 OBD 모듈로부터 데이터를 수신하고 데이터의 정상 여부를 checksum을 통해 확인한 후, 정상적인 데이터만 버퍼에 저장한다. 그리고 마지막으로 버퍼에 저장된 3가지 OBD 정보를 표 6과 같이 설정한 프레임 구조로 전송한다. 9105는 헤더, 뒤에 15바이트는 OBD 데이터이다. 00, 01, 02는 각각 RPM, SPEED, DISTANCE의 opcode이며 각각의 데이터는 opcode 뒤에 실려 전송된다^{6,7)}.

표 6. AP로 전송되는 프레임 구조
Table 6. Frame structure of sent to AP

HDR (2byte)		RPM (5byte)		SPEED (5byte)		DISTANCE (5byte)	
0x91	0x05	0x00	data	0x01	data	0x02	data

5.4 서버에서 데이터 확인

서버에서 사용하는 프로그램의 구조는 그림 9와 같다.

<1>은 포트번호로 서버의 포트번호를 지정하는 부분이다. 위에서 설명했듯이 Wi-Fi 모듈에서 미리 정의한 15151로 설정하였다. <2>는 데이터 도착시간을 알려주는 부분이고 <3>은 Wi-Fi 모듈의 IP를 알려주는 부분이다. <4>는 OBD에서 보내온 데이터를 표시하

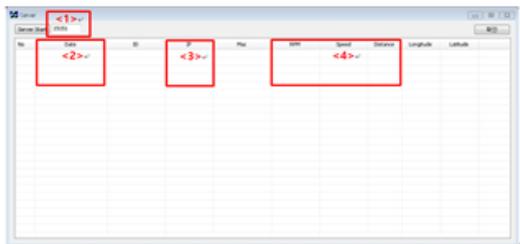


그림 9. 서버 프로그램
Fig. 9. Server program

는 부분으로 RPM, 속도 이동거리 각각의 데이터 값을 표현한다. 그림 10은 실제로 서버에서 정상적으로 데이터를 출력하는 모습이다.

그림 10. 데이터 출력
Fig. 10. Data output

VI. 실험 결과 및 분석

본 연구의 전체적인 시스템의 실제동작 방식은 그림 11과 같다.

실제로 차량을 이용한 테스트는 본교 캠퍼스에서 진행하였으며, 그 결과는 그림 12, 13, 14와 같다. 그림에 나와 있는 것과 같이 차량의 정보가 서버에서 빠르게 나타나는 것을 확인 할 수 있다.

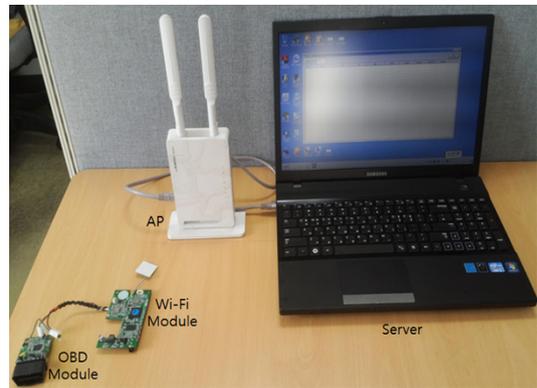


그림 11. 전체 시스템
Fig. 11. Entire system

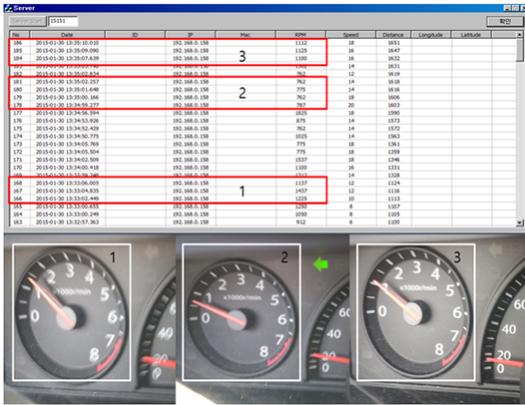


그림 12. RPM 정보 확인
Fig. 12. RPM Information check



그림 13. 속도 정보 확인
Fig. 13. Speed Information check



그림 14. 이동 거리 정보 확인
Fig. 14. Distance Information check

VII. 결 론

본 논문에서는 OBD 차량 진단 정보 통신을 위한 IoT 장치를 구현하기 위한 전체적인 기술개발 내용을 소개하였다. OBD 추출 H/W는 차량에 직접 테스트해

본 결과 정상적으로 작동하였고, 기기 간 시리얼 통신과 Wi-Fi와 AP의 무선통신도 정상적으로 작동하는 것을 확인 하였다. 차후에 이 기술을 활용하면 개인, 자동차 및 보험회사에서 해당정보를 빅데이터 처리하여 다양한 부가가치를 가진 정보로 새로이 이용할 수 있을 것이다⁸⁻¹⁰.

References

- [1] Y. Park, *CAN communication*(2006), Retrieved Aug. 5, 2015, from <http://gts2000.co.kr/>.
- [2] K. Ha, J. Lee, Y. Heo, S. Choi, and M. Shin, "A study of a simulator development generating OBD diagnostic code," in *Proc. KIEE Conf.*, pp. 1157-1158, Pyeongchang, Korea, Jul. 2007.
- [3] G. Geraldo, "Differences between on board diagnostic systems (EOBD, OBD-II, OBD-BR1 and OBD-BR2)," *2006 SAE Brasil Congress and Exhibit*, Nov. 2006.
- [4] D&K ICT, *Controller Area Network(CAN) to RS232 Converter User manual*(2013), Retrieved Aug. 7, 2015, from <http://dnkit.co.kr/>
- [5] Snap-on, *Global OBD Vehicle Communication Software Manual*, Snap-on Incorporated, (pp. 8, 10, 25), 2013.
- [6] H. Yun and J. Choi, "Vehicle station gateway standard technologies trends for cooperative ITS applications," *KSAE 2013 Annu. Conf. Exhibition*, pp. 1422-1425, Goyang-si, Korea, Nov. 2013.
- [7] M. Mosko, *TLV Packet Format* (2014), Retrieved Aug. 30, 2015, from <http://www.ccnx.org/>
- [8] H.-M. Park, S.-H. Park, W. C. Park, and H.-M. Seo, "Implementation of road weather information system supporting intelligent transportation systems based on USN," *J. KICS*, vol. 35, no. 3, pp. 485-492, Mar. 2010.
- [9] H.-S. Seo, H.-U. Kim, D.-G. Noh, and S.-S. Lee, "Algorithm design and implementation for safe left turn at an intersection based on vehicle-to-vehicle communications," *J. KICS*, vol. 38, no. 2, pp. 165-171, Feb. 2012.
- [10] S. Son, T. Kim, Y. Jeon, and Y. Baek, "Smart camera technology to support high speed

video processing in vehicular network,” *J. KICS*, vol. 40, no. 1, pp. 152-164, Jan. 2015.

이 성 희 (Seong-Hee Lee)



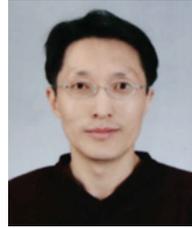
2014년 2월 : 동국대학교 전자
공학과 졸업
2016년 8월 : 동국대학교 전자
전기공학과 석사
<관심분야> 전자공학, 무선통
신, 광통신 공학

이 성 형 (Seong-Hyung Lee)



2014년 2월 : 동국대학교 전기
공학과 졸업
2015년 8월 : 동국대학교 전자
전기공학과 석사
2016년 : 롯데정보통신
<관심분야> 전자공학, 무선통
신, 기기간 통신

이 상 문 (Sang-Moon Lee)



1992년 2월 : 연세대학교 전기공
학과 졸업
1994년 8월 : 연세대학교 통신시
스템 석사
1994년~2000년 : 삼성전기 선임
연구원
2000년~2015년 : (주)제이애플
시스템 연구소장

2015년~현재 : (주)엔티로봇 기술이사
<관심분야> 무선통신, 유선 통신

황 승 훈 (Seung-Hoon Hwang)



1992년 2월 : 연세대학교 학사
1994년 2월 : 연세대학교 석사
1999년 2월 : 연세대학교 박사
1999년 2월~2005년 : LG전자,
이동통신 연구소 책임연구원
2005년 2월~현재 : 동국대학교
전자전기공학부 교수

<관심분야> 무선통신, IoT, CR, 스펙트럼엔지니어
링, 광통신