

차량정보 분석과 제스처 인식을 위한 AVN 소프트웨어 구현

오규태*, 박인혜*, 이상엽°, 고재진**

Development of AVN Software Using Vehicle Information for Hand Gesture

Gyu-tae Oh*, Inhye Park*, Sang-yub Lee°, Jae-jin Ko**

요약

본 논문은 차량 내 AVN(Audio Video Navigation)에서 차량정보 분석과 제스처 인식이 가능한 소프트웨어 구조를 설계하고 구현 방법을 서술한다. 설계된 소프트웨어는 차량정보 분석을 위해 CAN(Controller Area Network) 통신 데이터 분석 모듈을 구현하여 차량의 주행 상태를 분석했다. AVN 소프트웨어는 분석된 정보를 웨어러블 디바이스의 제스처 정보와 융합토록 했다. 도출된 융합정보는 운전자의 명령 수행 단계로 매칭하고 서비스를 지원하는데 사용됐다. 설계된 AVN 소프트웨어는 기성 제품과 유사한 환경의 HW 플랫폼 상에 구현되어 차량 주행 상황과 동일하게 모사된 상황에서의 차량정보분석, 제스처 인식 수행 등의 기능을 지원함을 확인했다.

Key Words : Audio Video Navigation(AVN), Vehicle information, Controller Area Network(CAN), Wearable device, Gesture recognition

ABSTRACT

This paper describes the development of AVN(Audio Video Navigation) software for vehicle information analysis and gesture recognition. The module that examine the CAN(Controller Area Network) data of vehicle in the designed software analyzes the driving state. Using classified information, the AVN software converge vehicle information and hand gesture information. As the result, the derived data is used to match the service step and to perform the service. The designed AVN software was implemented in HW platform that common used in vehicles. And we confirmed the operation of vehicle analysing module and gesture recognition in a simulated environment that is similar with real world.

I. 연구배경

최근 차량 내 인포테인먼트 시스템이 급속도로 증가함에 따라 보다 편리하며 첨단 기능을 갖춘 지능형

차량 시스템들이 등장하고 있다. 다양한 차량 시스템들 중에서 주행 정보와 관련된 정보를 이용한 시스템들은 차량 주행에 있어 중요한 시스템이라 할 수 있다^[1,2]. 차량 주행 정보는 차량에서 사용하는 전장장치들

※ 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업핵심기술개발사업의 일환으로 수행하였음. [10062943, 지능형 운전지원 시스템(ADAS)을 위한 산업융합 웨어러블 디바이스 및 서비스 개발]

• First Author : Korea Electronic Technology Institute, gto@keti.re.kr, 정희원

° Corresponding Author : Korea Electronic Technology Institute, syublee@paper.korean.ac.kr, 정희원

* Korea Electronic Technology Institute, in-hye@keti.re.kr

** Korea Electronic Technology Institute, jaejini@keti.re.kr

논문번호 : KICS2017-01-019, Received January 16, 2017; Revised March 31, 2017; Accepted April 20, 2017

사이에서 정보교환을 필요로 하며 이에 따라 효율적인 통신 환경을 제공하는 CAN(Controller Area Network)이 등장하게 되었다. CAN은 현재 대부분의 차량에서 사용하는 프로토콜이며 CAN은 ECU(Electronic Control Unit) 장치들의 정보교환에 유용하게 사용 된다³⁾. 또한 차량 사용자들은 AVN(Audio Video Navigation)을 통하여 네비게이션 기능 외에 다양한 멀티미디어 콘텐츠를 사용한다.

최근에는 AVN의 다양한 서비스를 사용자의 음성 인식이나 에어 제스처를 통하여 이용할 수 있는 차량 보급이 점차 늘어나고 있다. 음성인식 방식은 주행에 방해요소가 적기 때문에 편의성은 높일 수 있지만 차량 내·외부에서 발생 가능한 여러 노이즈로 인해 음성 인식의 정확도가 떨어질 수 있는 단점이 있다⁴⁾.

차량 내 제스처 기능 방식의 대부분은 카메라 기반의 방식을 사용하고 있다. 그러나 카메라 기반의 제스처 방식을 사용하는 차량의 모델은 개발 비용이 높고, 외부 조건에 의한 빛의 반사량, 빛의 번짐 등으로 인한 카메라 인식 오류의 단점이 있다. 또한 제스처 인식 영역이 지정되어있어 사용자는 지정된 영역에서 제스처 인식을 수행해야 하는 단점이 있다⁵⁾. 카메라를 이용한 에어 제스처 방식은 관련 연구에서 이용자에 따라 약 80%대의 인식률을 보이고 있다⁶⁾.

본 연구에서는 카메라 기반 방식의 제스처 인식의 단점을 보완한 웨어러블 디바이스를 사용한 제스처 인식 방법을 사용한다. 사용자 주변 환경의 영향과 제스처 위치 관계없이 동작 인식되어 차량 응용 시스템을 사용할 수 있어야 한다. 본 논문에서는 차량정보 분석과 웨어러블 디바이스를 사용하여 제스처 인식이 가능한 차량용 AVN 소프트웨어를 구현한다. 차량정보 분석을 위하여 CAN 통신 분석 모듈 구현과, 웨어러블 디바이스를 이용한 제스처 정보를 융합한다. 본 논문에서 구현된 AVN 소프트웨어는 주행 정보의 변화에 따라 제스처 인식을 제한하는데, 제스처 인식을 항상 가능하게 유지하지 않는 이유는 주행 중 운전자의 집중도를 떨어뜨릴 가능성이 존재하고, 차량의 움직임으로 인한 잘못된 제스처 인식을 수행할 가능성이 매우 높기 때문이다.

본 논문의 구성은 2장에서 구현하는 AVN 소프트웨어의 구조와 설계를 설명하고 3장에서는 2장에서 설명한 AVN 소프트웨어의 설계의 구현 환경과 동작에 대해서 설명한다. 마지막으로 4장에서는 구현 시스템의 결론에 대하여 설명함으로 본 논문을 마친다.

II. AVN 소프트웨어 구조 및 설계

이 장에서는 구현된 AVN 소프트웨어의 전체 시스템 구조를 설명한다. 구현된 AVN 소프트웨어는 차량 내 동작하는 AVN의 동작을 모사한 소프트웨어 플랫폼이다. 특히 사용자 서비스 지원을 위해서 웨어러블 밴드와 연동하여 제스처 인식 기능을 탑재한다. 제스처 인식 기능과 차량에서 발생하는 운전정보를 융합한 차량 정보와 제스처 정보를 융합해 부가 기능을 사용자에게 제공한다.

구현하는 AVN 소프트웨어는 차량 주행정보와 제스처 데이터의 분석 및 제어의 기능이 주요 기능이다. 따라서 구현된 본 시스템은 크게 두 분류의 핵심 입력 모듈 인터페이스를 갖는다. 차량 발생 정보를 수집하기 위한 CAN 수신 모듈과 제스처 정보 수집을 위한 제스처 인식 모듈이다. AVN 소프트웨어는 설명된 두 모듈의 데이터를 분석하여 현재 차량정보 상태를 파악 한 뒤 각 단계별로 제스처 기능이 제한되도록 하였다. CAN 수신 모듈에서 분석된 차량 정보는 웨어러블 밴드에서 발생한 제스처 인식 수행 여부를 결정하기 위해 이용된다. 이어지는 절에서 본 AVN 소프트웨어의 전체 시스템, CAN 수신 모듈, 제스처 인식 모듈에 대해 순서대로 설명한다.

2.1 전체 시스템 구조

차량정보 분석과 제스처 인식을 위한 AVN 소프트웨어의 전체 시스템 구조는 그림 1과 같다. AVN 소프트웨어의 플랫폼은 크게 세 가지 모듈로 구성되어 있다.

AVN 플랫폼은 제스처 인식을 위하여 웨어러블 디바이스와 블루투스 방식으로 진행되기 때문에 블루투스 라이브러리를 사용하였다. 또, 웨어러블 디바이스의 센싱 데이터를 해석 하는 제스처 라이브러리와 차량 정보 분석을 위하여 CAN 데이터 수신을 위한 라이브러리로 이루어진다.

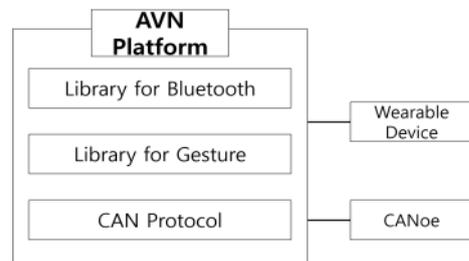


그림 1. AVN 플랫폼 구조
Fig. 1. Structure of AVN platform

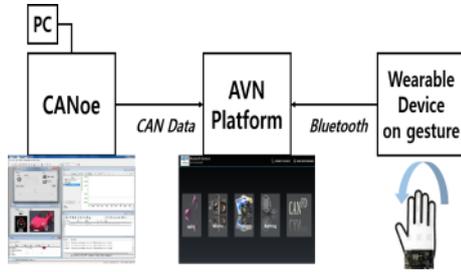


그림 2. 전체 시스템 디자인
Fig. 2. Full system design

AVN 소프트웨어는 세 가지 모듈의 기능을 처리한다. 전체 시스템은 그림 2에 나타나있다. 차량용 AVN 소프트웨어는 CANoe에서 실제 차량 정보를 보내게 되면, 전달 받은 데이터를 해석 및 분석한다. 동시에 제스처 기능을 사용할 수 있도록 제스처의 센싱 데이터를 블루투스 통신을 이용하여 수집하여 사용자에게 차량 주행 상태에 따라 제스처의 사용 유무를 전달 및 제어한다.

2.2 CAN 모듈

CAN 데이터 송신을 위한 모듈은 차량정보 분석을 위해 ECU 테스트를 생성할 수 있는 CANoe 툴을 사용하여 통신을 진행하였다. 위 툴은 실제 차량 데이터를 모사한 CAN 통신을 적용하여 CAN 통신 데이터 모니터링, 저장 및 임의의 데이터 송신이 가능한 개발 툴이다. CAN 데이터의 수신구조는 그림 3과 같다.

그림 3의 구조는 AVN 플랫폼에 CAN 데이터를 전달하기 위해 CAN 데이터를 생성하는 다른 PC에서 CAN 데이터를 생성하여 전달하는 구조이다. CAN 데이터 송신은 CAN 통신 프로토콜에 정의된 명령에 따라 진행하였다⁷⁾.

CAN 데이터는 HEX/ASCII 값으로 전달되며 송신 데이터는 그림 4와 같다. 그림 4는 차량 주행정보 데이터를 갖는 CAN 데이터의 송신을 보여주고 있다. AVN 플랫폼에서는 전달 받은 CAN 데이터를 수집/분석하여 차량 주행정보에 따라 세 단계로 분류한다. AVN 소프트웨어에서는 분류 된 세 단계의 주행정보

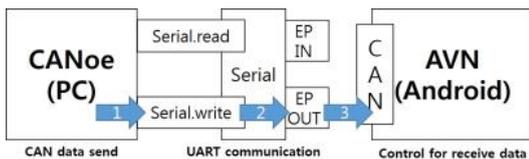


그림 3. CAN 데이터 수신구조
Fig. 3. CAN data reception structure

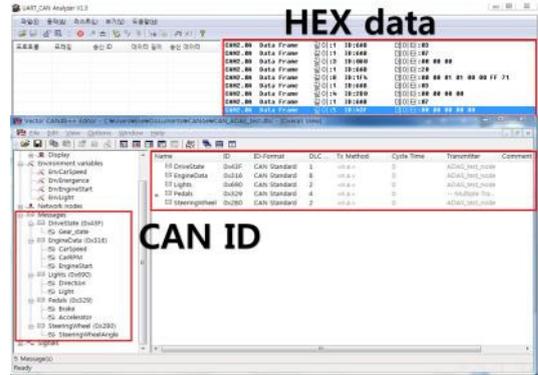


그림 4. CAN 송신 데이터
Fig. 4. CAN transmission data

를 이용하여 각 단계별 제스처 인식 가능 여부를 판단하게 된다.

2.3 웨어러블 디바이스 모듈

제스처 정보를 발생하여 AVN 플랫폼으로 전달하는 웨어러블 디바이스의 하드웨어 플랫폼은 그림 5에 나타났다.

그림 5의 웨어러블 디바이스는 제스처 정보를 수집하는 6축 센서와 블루투스 모듈이 장착되어 있다. MCU는 Cortex-M 시리즈인 EFM32 HG32를 이용하여 제스처 센싱 수집과 블루투스 통신을 진행하며 센서 값의 1차 가공이 가능하도록 했다. 핵심 모듈인 센싱과 블루투스 통신의 보드는 그림 5에서 점선으로 표시하였으며, 전원 공급을 위한 레퍼런스 플랫폼은 점선 외의 부분이다. AVN 플랫폼에서 제스처 센싱 데이터를 해석하기 위해 정형화된 데이터 프로토콜이 필요하다. 센싱 데이터 전달 과정에서 발생하는 오류 존재 여부와 센싱 데이터의 전송 시점을 파악해야 되기 때문이다.

다음 그림 6에 센싱 데이터 프레임의 구조를 나타냈다. 정형화된 각각의 2byte 데이터들은 시작과 보내는 센싱 데이터의 수, 제스처 센싱 데이터, 종료로 구

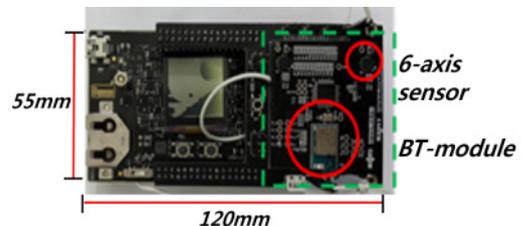


그림 5. 웨어러블 디바이스 플랫폼
Fig. 5. Wearable device platform

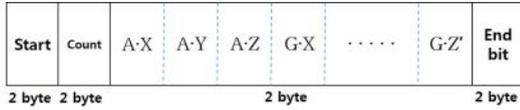


그림 6. 센싱 데이터 프레임
Fig. 6. Sensing data frame

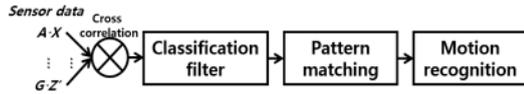


그림 7. 제스처 인식 시스템의 구성
Fig. 7. Configuration of Gesture Recognition System

분되어 있다. 그림 6의 센싱 데이터 프레임에서 시작 비트와 종료 비트의 값이 일치하지 않는다면 센싱 데이터 전달 시 프레임의 오류로 판단하여 수집된 데이터 정보는 버려지게 된다.

그림 7은 제스처 인식 시스템의 구성이다. 그림 7의 구성도는 헤드유닛 시스템에서 동작한다. 모션인식을 위해 웨어러블 디바이스에서 발생한 센서 값의 상호 상관관계(cross-correlation)를 계산한다. 그 다음으로, 얻어진 상관관계 정보를 이용해 무의미한 값을 필터링 하고 저장된 동작 DB 값과 비교하여 사용자의 제스처를 인식하도록 구성되었다. 설명된 알고리즘은 웨어러블 디바이스로부터 전달받은 6축 센싱 데이터를 이용하여 사용자의 제스처를 인식하는 방식이다⁵⁾.

III. 구현환경 및 동작

이 장에서는 구현 환경과 AVN 소프트웨어 동작에 대해서 설명하겠다. 그림 8은 구현에 사용되는 차량용 AVN 플랫폼이다.

차량용 AVN 플랫폼의 시스템은 안드로이드 기반으로 구현하였다. 구현한 소프트웨어의 구성은 기본적으로 차량용 AVN과 같이 터치 동작으로 진행 할 수



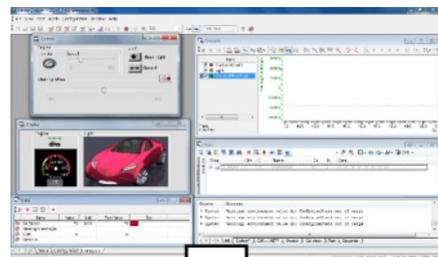
그림 8. 차량용 AVN 플랫폼 HW
Fig. 8. Vehicle AVN Platform HW

있다. AVN 소프트웨어는 웨어러블 디바이스와의 동작을 위해서는 블루투스 연결을 통하여 진행할 수 있도록 구현하였다.

그림 9는 CANoe를 이용한 차량 주행정보를 AVN 소프트웨어로 전달을 나타낸다. 주행정보 데이터는 차량 주행 정보에 관련된 주요 정보만 전달한다. 주요 정보로는 현재 시동 상태, 시속, 핸들의 조향각, 라이트 점화 상태의 정보를 전달 할 수 있다. 그림 9에서처럼 AVN 소프트웨어는 실시간으로 차량 주행 정보를 CAN 데이터 신호를 통하여 전원공급이 되는 동안 실시간으로 전달 받아 분석할 수 있다.

그림 10은 AVN 소프트웨어의 동작 절차를 나타낸 흐름도이다. 그림 10의 흐름도에 나타난 바와 같이 차량 주행정보를 이용하여 세단계의 차량 상태 결정 기준에 따라 제스처 인식의 유무가 구분되어 진다. 현재 차량 주행 정보가 1단계는 모든 제스처의 사용이 가능하다. 2단계는 부분적 제스처만 가능한 상태이다. 3단계는 모든 제스처가 동작하지 않는 상황이다. 이처럼 현재 차량 주행정보에 따라 제스처 동작 여부를 구분 짓는 이유는 특정 주행 상태에서는 제스처를 수행하지 못하게 설정하기 위해서이다.

그림 11은 본 논문에서 구현하는 전체 시스템의 구성 화면이다. 그림 11에서 ①은 AVN 소프트웨어의 디스플레이 화면이며, ②는 그림 7의 AVN 플랫폼으로 차량 주행정보와 제스처 센싱 데이터를 받아 분석



Real-time CAN Data engine state, speed wheel, light

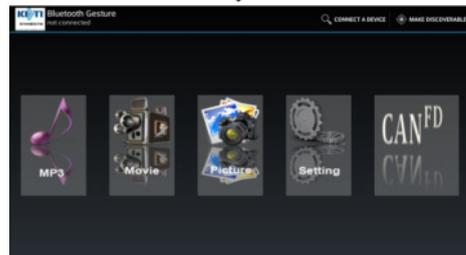


그림 9. 차량 주행정보 전달
Fig. 9. Forwarding of vehicle driving information

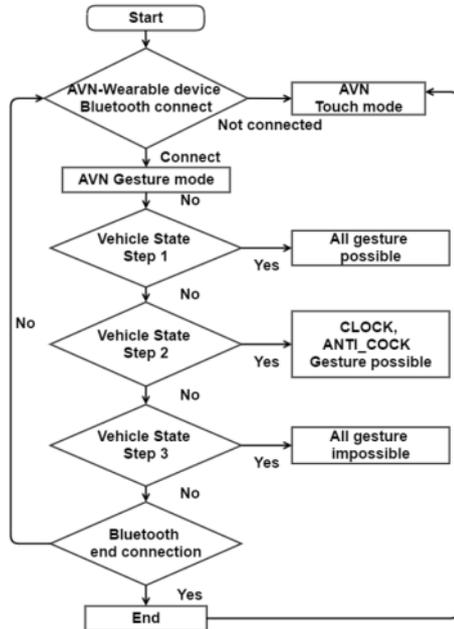


그림 10. AVN 소프트웨어 동작 절차
Fig. 10. AVN software operating procedure



그림 11. 전체 시스템 구성 환경
Fig. 11. System Configuration Environment

및 해석을 처리한다. ③은 차량 주행정보를 전달할 수 있도록 Canoe 톨을 이용한 톨이며, ④를 통하여 CAN 통신으로 AVN 플랫폼에게 차량 주행정보 데이터를 전달한다. ⑤는 그림 4에서 설명한 제스처 인식을 위한 웨어러블 디바이스이다.

본 논문의 구현 결과를 그림 12에 나타내고 있다. 제스처의 기능 정의는 현재 구현한 AVN에서 사용자 제공 서비스를 이용할 때 사용되도록 정의하였다. 제스처 동작의 직관상 판단되는 응용으로 메뉴이동, 이전메뉴 이동, 음소거, 메뉴 선택으로 구성되었다. 실제 BMW7 시리즈에 적용된 제스처와 유사하게 주행 중

State	Step. 1	Step. 2	Step. 3
AVN action			
Clock (Move menu)	OK	OK	NO
Anti Clock (Previous menu)	OK	OK	NO
Left/Right (Mute)	OK	NO	NO
Front (Select menu)	OK	NO	NO

그림 12. 차량 주행에 따른 제스처 동작
Fig. 12. Gesture operation in accordance with the vehicle running

응용 가능한 제스처를 이용하였다⁸⁾.

그림 12의 결과는 표 1의 기준에 따라서 동작하는 제스처의 인식의 결과이다. 차량 주행정보를 이용하여 세단계의 차량 상태 결정 기준에 따라 제스처 인식의 유무가 구분되어 진다. 현재 차량 주행 정보가 1단계는 엔진 시동이 걸려있으며 차량속도와 조향각의 변화가 없는 상태이다. 1단계 상태에서는 모든 제스처의 기능이 동작할 수 있는 단계이다. 2단계는 현재 차량 속도가 일정하지만 조향각의 변화가 ± 10 도 이하인 상태이다. 2단계에서는 시계방향 회전과 반시계방향 회전의 두 가지 제스처의 사용만 가능하다. 3단계는 현재 차량속도가 변하고 있는 상황이며 동시에 조향각의 변화도 변하고 있는 상황이다. 3단계에서는 모든 제스처의 기능이 제한된다. 이처럼 현재 차량 주행정보에 따라 제스처 동작 여부를 구분 짓는 이유는 급정거 혹은 급발진, 급회전 같은 차량 주행상태의 급변하는 상황에서 제스처의 센싱 데이터 값 역시 급변하기 때문에 차량 주행 상태에 따른 제스처 유무를 구분지어 사용자에게 제스처의 기능 혹은 불가 여부를 알려주게 된다.

표 1. 차량 상태 결정 기준 테이블
Table 1. Vehicle condition decision table

	Vehicle state		
	Engine	Speed	Wheel
Step. 1	Off	No change	No change
	On	No change	No change
Step. 2	On	Constant speed	± 10 below
Step. 3	On	Change speed	± 10 excess

IV. 결론 및 향후 연구방향

본 논문에서는 차량정보 분석과 제스처 인식을 위한 차량용 AVN 소프트웨어를 설계 및 구현하였다.

차량정보 분석을 위해 차량 데이터를 전송할 수 있는 CANoe 인터페이스를 사용하여 실제 차량 주행정보를 AVN 소프트웨어에서 분석할 수 있도록 진행하였으며, 차량 주행정보를 이용하여 주행정보 상태에 따라 제스처 인식 가능 여부를 판단하여 사용자에게 제어 및 전달하는 AVN 소프트웨어를 구현하였다. 웨어러블 디바이스를 이용한 제스처 인식을 통하여 카메라 기반의 제스처 인식 시스템의 단점을 보완하였다. 본 논문에서 구현한 소프트웨어를 적용한다면 저가의 임베디드 장치에서 비용적인 측면에서 효율적인 효과를 가져 올 수 있다.

본 구현을 통하여 향후 다양한 제스처 인식을 통하여 참조논문 [5]의 연구결과인 80% 정도의 제스처 인식률을 90% 이상까지 높일 예정이며, 실제 차량 탑승을 통하여 여러 가지 차량 주행 상황 및 돌발 상황에서 구현한 AVN 소프트웨어의 서비스를 적용할 수 있도록 보완 및 적용할 예정이다. 이를 통하여 실제 주행정보에 따른 차량 상태 결정 기준표를 작성할 예정이다.

References

[1] M. Im, "Vehicle communication network technology," *J. KICS*, vol. 24, no. 9, pp. 86-95, Sept. 2007.

[2] K. Lee and J. Choi, "Integrated vehicle safety control system for smart vehicles," *J. Auto Journal Publication*, vol. 34, no. 6, pp. 20-25, 2012.

[3] R. Bosch, *CAN Specification Version 2.0*, 1991.

[4] S.-J. Lee and S.-M. Kang, "A study of automatic evaluation platform for speech recognition engine in the vehicle environment," *J. KICS*, vol. 37C, no. 07, pp. 538-544, Jul. 2012.

[5] I. H. Park, S. Y. Lee, and J. J. Ko, "A research on development and service of wearable device for gesture recognition in Vehicle," *Information & Communications Magazine*, vol. 33, no. 6, pp. 39-46, Jun. 2016. (박인혜, 이상엽, 고재진, "차량 내 제스처 인식을 위한 밴드형 웨어러블 디바이스 개발 및 서비스에 관한 연구", *한국통신학회지*, 제33권 6호, pp. 39-46, 2016.)

[6] T. Kopinski, S. Geisler, and U. Handmann, "Gesture-based human-machine interaction for assistance systems," *IEEE Int. Conf. Inf. and Automation*, Aug. 2015.

[7] UART/CAN communication protocol, Retrieved Jan. 30, 2017. from <http://www.rialsys.co.kr>

[8] BMW 7 Series, Retrieved Feb. 3, 2017, from http://www.bmw.com/com/en/newvehicles/7series/sedan/2015/showroom/innovative_functionality.html

오 규 태 (Gyu-tae Oh)



2013년 : 백석대학교 공학사
 2015년 : 광운대학교 공학석사
 2016년~현재 : 전자부품연구원
 임베디드SW센터 연구원
 <관심분야> 임베디드시스템
 SW, 차량용 임베디드 시스템,
 웨어러블 디바이스

박 인 혜 (Inhye Park)



2008년 : 광운대학교 공학사
 2010년 : 광운대학교 공학석사
 2015년 : 광운대학교 공학박사
 2012년~현재 : 전자부품연구원
 임베디드SW센터 전임연구원
 <관심분야> 임베디드시스템
 SW, 차량용 임베디드 시스템,
 웨어러블 디바이스

이 상 열 (Sang-yub Lee)



2003년 : 연세대학교 공학사
2005년 : 연세대학교 공학석사
2015년~현재 : 고려대학교 컴퓨터정보학 박사과정
2005년~2009년 : 삼성전기 중앙연구소 선임연구원
2009년~현재 : 전자부품연구원 임베디드, SW센터 선임연구원

<관심분야> In-vehicle 네트워크 시스템, 차량용 임베디드 시스템

고 재 진 (Jae-jin Ko)



1997년 : 광운대학교 공학사
2005년 : 광운대학교 공학석사
2013년 : 광운대학교 공학박사
2000년~현재 : 전자부품연구원 임베디드, SW센터 센터장
<관심분야> 웨어러블 디바이스, 임베디드 시스템