# 무인자율주행차량의 시스템 아키텍처 및 통신 프로토콜 설계

# Development of System Architecture and Communication Protocol for Unmanned Ground Vehicle

문 희 창, 우 훈 제, 김 정 하*
(Hee Chang Moon, Hoon Je Woo, and Jung Ha Kim)

**Abstract :** This paper deals with the peer-to-peer data communication to connect each of distributed levels of developed unmanned system according to the JAUS. The JAUS is to support the acquisition of unmanned system by providing a mechanism for reducing system life-cycle costs. Each of distributed levels of the JAUS protocol divides into a system, some of subsystems, nodes and components/instances, each of which may be independent or interdependence. We have to distribute each of the levels because high performance is supported in order to create several sub-processor computing data in one processor with high CPU speed performance. To complement such disadvantage, we must think the concept that a distributed processing agrees with separating each of levels from the JAUS protocol. Therefore, each of distributed independent levels send data to another level and then it has to be able to process the received data in other levels. So, peer-to-peer communication has to control a data flow of distributed levels. In this research, we explain each of levels of the JAUS and peer-to-peer communication structure among the levels using our developed unmanned ground vehicle.

**Keywords :** Unmanned Ground Vehicle(UGV), DARPA challenge, peer-to-peer, JAUS, bandwidth, latency, header

## I. 연구 배경

20세기 초, 로봇에 대한 관심이 날로 높아지면서 선진국을 중심으로 활발한 로봇 연구가 진행 중이다. 특히, 로봇 산업 분야 중 가장 활발히 진행 중인 연구 분야로는 무인화 기술(Unmanned Autonomous Technique)을 들 수 있다. 이러한 기술은 인간의 간섭 및 제재 없이 주변 환경 정보를 탑재된 센서로 수집하고 수집된 데이터를 통해 적합한 상황 판단 및 행동을 행할 수 있어야 한다. 더욱이 상용 제품에서도 무인화 기술을 적용하여 사람이 제품을 조작하는 상황에서 제품 스스로 다시 판단하여 사람의 오판을 줄일 수 있는 보조 장치로 개발을 하고 있다. 예로, 현재 고급 상용차에 쓰이고 있는 차선 이탈 경고 장치(Lane Departure Warning System: LDWS), 차선 유지 보조 장치(Lane Keeping Assistance System: LKAS), 사각지대 경보 장치(Blind Spot Information System: BLIS)는 운전자가 차선 이탈에 대한 돌발 상황에 대처하지 못할 시 차량에 부착된 센서를 통해 차량 스스로 인지하고 차량의 조향을 제어할 수 있도록 하는 안전 장치 등을 들 수 있다. 이러한 기술들은 현재 개발 중이거나 개발이 완료되었다.

최근 미국은 이러한 무인화 기술을 적용한 새로운 개념의 전투를 고려하여 전투 선단에 무인 지상 차량(Unmanned Ground Vehicle: UGV) 및 무인 항공기(Unmanned Aerial Vehicle: UAV)를 배치하고 후방의 안전한 곳에 배치된 지휘 통제 차량이나 기지를 통해 모든 전투를 지휘할 수 있는 네트워크 기반의 미래 전투 체계(Future Combat System: FCS)를 개발 중에 있다[1]. 이 전투 체계 개발의 일환으로 미 국방성 산하 고등 연구계획국(Defense Advanced Research Project Agency: DARPA)은 무인화 기술의 인프라 확충을 위하여 대학 및 민간 기업 연구소에 기술 개발을 권장하고자 2004년부터 DARPA Grand Challenge라는 무인 자율 주행 차량 대회를 개최하고 있다.

2004년과 2005년 두 해에 개최된 이 대회는 사막과 초원으로 이루어진 약 200마일의 거리를 10시간 이내에 그림 1과 같은 무인차량이 스스로 주행하도록 하는 대회이다[2]. 또한, 2007년 도심지 내의 일반 도로에서 교통 신호 인식, 무인 차량의 선행 환경에 대한 반응, 주어진 임무에 대한 수행 능력, 원하는 목적지까지 안전하게 주행할 수 있도록 하는 DARPA Urban Challenge가 개최되었다. 이는 2015년까지 미군 수송차량의 1/3을 무인화하려는 미 국방성의 계획을 현실화시키고 있으며 특히 이라크 전쟁에서 지상 수송차량에 무인화 기술을 접목시킨 이례가 있다[3].

우리나라에서도 이러한 로봇 산업이 국가 정책 산업 중 하나로 중장기적인 계획을 세우고 산학에 개발 지원을 하고 있다. 특히, 무인 기술은 다른 산업의 활용도 측면에서 우수한 기술임에도 불구하고 기반 기술 및 개발의 어려움으로 인해 몇몇 기관 및 연구소에서만 연구가 진행되고 있다. 최근 들어 미국을 중심으로 한 군사 무기 체계의 무인화 작업이 본격화 되면서 우리나라도 무기 체계의 첨단화, 무인화에 박차

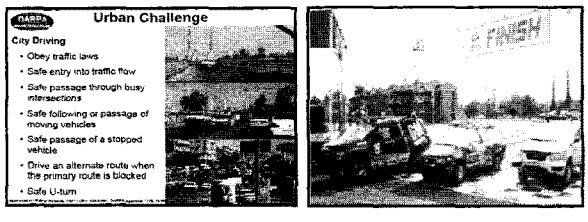



그림 1. DARPA urban challenge 및 출전 차량.
Fig. 1. DARPA urban challenge & participated vehicle.

* 책임저자(Corresponding Author)
논문접수 : 2008. 5. 15., 채택확정 : 2008. 6. 30.
문희창, 우훈제 : 국민대학교 자동차공학전문대학원
(hcmoon@kookmin.ac.kr/redbaron@kookmin.ac.kr)
김정하 : 국민대학교 기계자동차공학부(jhkim@kookmin.ac.kr)
※ 본 연구는 2008년도 국민대학교 우수연구센터사업비를 지원받아 수행된 연구임.

를 가하고 있다. 2012년까지 GOP(General Out-Post- 전방 일반 전초)의 경계력 강화 조치의 일환으로 GOP 과학화 경계 체계를 단계적으로 시험, 보급할 예정이며, 국방 과학 연구소에서는 2020년까지 야지용 무인 자율 주행 차량을 총 10단계의 자율 레벨을 구분하여 연구 목표를 설정하고 국내 유수의 방산업체 등을 위주로 무인 자율 주행 차량에 관한 연구가 활발히 진행되고 있다. 이렇듯 자동차에 대한 연구는 상용부문뿐만 아니라 군사 및 실생활에도 많이 보급되는 추세로 기술적, 경제적인 파급효과가 클 것으로 기대를 모으고 있다.

## II. 연구 목적

무인 자율 주행 차량은 LDWS나 LKAS와 같이 하나의 체계 또는 프로세서로 작동하는 것이 아니라 서로 다른 임무를 지닌 여러 개의 체계들이 적절한 우선권(priority)을 가지고 차량을 제어한다. 쉽게 사람이 차량을 운전할 때 보고, 듣고, 생각할 수 있는 형태를 무인 체계 상에서는 센서의 수치 데이터와 무인 체계의 제어 알고리즘을 바탕으로 차량을 동작시킬 수 있어야 한다. 특히, 무인 자율 차량을 제어하기 위해서는 항법, 장애물 인식, 차량 제어, 원격 제어, 경로 계획, 통합 체계로 분산하여 제어하게 된다.

이렇게 무인 자율 주행 차량의 체계를 다수의 분산된 제어 계층으로 나누는 이유는 무인 자율 주행 차량에 부착된 여러 종류의 센서가 대용량의 데이터를 고속으로 보내며, 어떤 한 계층의 체계에서 얻은 센서 데이터를 다른 계층에서 요청 시 응답할 수 있어야 하기 때문이다. 즉, 여러 계층이 유기적으로 서로의 데이터를 공유할 수 있어야 한다. 또한 다수의 대용량의 데이터를 하나의 계층에서 고속으로 처리하기 위해서는 체계 내부에 여러 서브 프로세서가 생성되어야 하며 각 서브 프로세서는 동기적으로 임의의 알고리즘을 통해 계산해야 한다. 그러나, 하나의 체계 또는 프로세서만을 이용해 차량을 실시간 제어하기에는 불가능하며 실현 가능하도록 체계를 구성하더라도 안정성 및 체계의 높은 성능이 뒷받침되어야 한다. 이에 무인 자율 주행 차량의 각 계층은 하나의 체계 또는 프로세서가 아닌 다수의 계층으로 나누어져 있어야 하며 분산 처리되어야 한다.

이러한 무인 체계 및 무인 자율 주행 차량의 분산 처리를 위하여 DARPA에서는 JAUS(Joint Architecture for Unmanned Systems)를 참조 규약을 정하고 있는데 JAUS는 각 계층을 4~계층으로 나누어 처리하고 있다. 각 계층은 서로 유기적으로 묶여 있어 실시간으로 들어오는 고속의 대용량 데이터를 처리하며 처리된 데이터를 다시 동기적으로 통합하여 마지막으로 차량의 조향, 속도 및 브레이크의 데이터 수치와 제어에 대한 우선 순위를 결정하도록 한다.

이렇듯 본 논문은 JAUS 규약에 의해 분산된 계층을 서로 독립적, 상호보완적, 유기적으로 구속하기 위해서 현재 개발된 무인 자율 주행 차량의 분산된 각각의 체계에 대한 설명과 아키텍처 설계, 체계간 통신 프로토콜 설계에 대해 설명한다.

## III. JAUS 와 무인자율주행차량의 시스템 아키텍처

### 1. JAUS 개요

JAUS는 무인 체계 및 무인 자율 주행 차량에 대한 연구, 개발과 데이터 획득의 사용에 대해 정의하도록 미 의회가 승인하고 미 국방성이 제시한 참조 규약(reference protocol)이다. 이 규약은 무인 체계를 개발함에 있어 체계를 효율적으로 구성하고 각 체계 또는 계층간의 인터페이스 방법을 제시함으로써 원활한 데이터 송수신이 가능하도록 되어 있으며, 다른 무인 체계에 이식하여도 호환성을 유지함에 제약이 없다는 것을 제시하고 있다. 즉, JAUS 규약을 따른 무인 정찰 로봇의 체계를 무인 자율 주행 차량의 체계에 그대로 이식하여도 무인 자율 주행 차량의 각 체계 간 인터페이스가 서로 능동적으로 연결될 수 있도록 설계가 가능하다는 것이며 표준 규약인 JAUS 규약을 따른 어떠한 무인 개체들 간의 통신 연결도 가능하다는 것이다. 그밖에, JAUS는 무인 체계 구축에 들어가는 비용과 유지 보수에 따른 기간을 최소화하는 것에도 목적을 두고 있다.

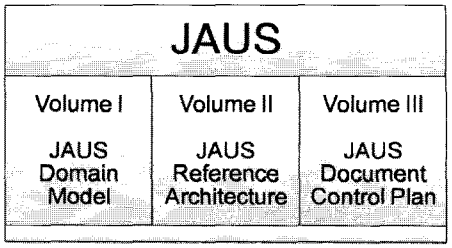



그림 2. JAUS 부분 별 명세.
Fig. 2. Specification of JAUS.

표 1. RA의 세부 사항.
Table 1. Notes of RA.

| Number of Part | Definition | Note |
|---|---|---|
| RA Part 1 | Architecture Framework | JAUS의 구조를 설명 |
| RA Part 2 | Message Definition | JAUS 정의 명세서<br>이 문서는 JAUS 메시지의 모든 헤더와 데이터 형을 명세화하였고 각 계층별 통신 관리에 대해 설명 |
| RA Part 3 | Message Set | JAUS의 모든 메시지에 대한 명세서 |

JAUS는 크게 무인 체계를 보조하기 위해 개발자나 사용자가 이미 알고 있거나 잠재적으로 알고 있는 체계 동작에 필요한 모든 장치의 모델들을 제시하며 그림 2와 같이 무인 체계의 함수와 정보를 정의하는 Domain Model(DM)[4], 무인 체계 상에서 JAUS를 참조하여 무인시스템의 체계를 설계할 기술적 명세를 제시한 Reference Architecture(RA), 마지막으로 무인시스템 체계 개발 시 변경 및 요청 사항을 추적하고 확인하기 위해 기록된 과정을 정의한 JAUS Document Control Plan으로 나누어져 있다.

표 1은 RA의 세부 사항을 보여주고 있다. 앞서 밝힌 JAUS volume I, II, III는 모두 국제 표준(The International System of Units: SI)과 JTA(The Joint Technical Architecture - 미 국방성에서 군 체계 별 상호 작용 및 연결에 대해 정의한 표준 기술 아키텍처)을 따르고 있다[5].

2. JAUS의 목적 및 기술적 구속력

JAUS의 목적은 크게 4가지로 설명할 수 있다. 첫째, 무인 체계의 개발을 위한 효율적 아키텍처 설계와 인터페이스를 정의하며, 두 번째, 무인 체계의 개발에 있어서 이전 기술의 재사용을 통한 개발 비용 절감이 있다. 세 번째는 개발 기간 단축 및 새로운 무인 체계 보완을 위한 참고사항을 정의하였으며, 마지막으로 미래 기술 발전에 따른 이전 체계의 손쉬운 업그레이드를 목적으로 두고 있다.

이러한 목적은 다음의 기술적 구속력에 따라 더욱 세부화 할 수 있다. 첫째, 무인 체계는 다양한 임무를 수행하기 때문에 무인 체계를 구축할 수 있는 플랫폼-차량, 로봇, 헬기, 비행기, 선박, 운영체제, 센서 등에 제약이 없다. 두 번째, 무인 체계는 주변환경의 상태나 정보 취득에 대해 독자적으로 수행할 수 있도록 설계한다. 세 번째, 단일 무인 체계는 새로운 임무와 더욱 진화된 자율성을 적용하기 위한 체계의 생산 사이클 효율을 이끌어 내야 하기 때문에 무인 체계 설계자는 각 요소 체계의 필요한 부품을 이용하여 컴퓨터 하드웨어 아키텍처를 유연성 있도록 설계해야 한다. 즉, 모듈화된 컴퓨터 하드웨어 구조를 지녀야만 하드웨어의 다양성을 이끌어 낼 수 있다는 것이다. 마지막으로 무인 체계는 하드웨어의 독립성도 중요하지만 이보다 기술적인 접근이 더욱 집중되고 있다. 즉, 하드웨어를 통제하는 소프트웨어의 기술적 독립성도 중요하다는 것이다.

3. JAUS 아키텍처와 요소 정보

그림 3은 JAUS의 가상적인 계층을 보여주고 있다. 다음의 계층을 설명하자면 상단 최상위 계층에 하나의 system을 두고 그 밑으로 하나 또는 여러 개의 subsystem으로 구성되어 있으며, 각 subsystem은 하부로 하나 또는 여러 개의 node들로 구성되어 있다. 그리고 하나의 node는 하부로 하나 또는 여러 개의 component로 나눌 수 있으며 하나의 component는 하나 또는 여러 개의 instance로 나눌 수 있다. 하지만 component는 instance와 같이 사용할 수 있어 node는 component없이 instance로 바로 연결할 수 있다.

JAUS는 여러 계층으로 체계를 분산시켜 각 계층 안에 있는 체계 구성이 하나 이상의 프로세서를 하위 계층에 두고 있는 형식을 띠고 있으며 각 계층 간 또는 같은 계층의 모든 체계는 데이터 통신을 통해서 유기적으로 연결된 분산 컴퓨팅을 행할 수 있게 만들 수 있다.

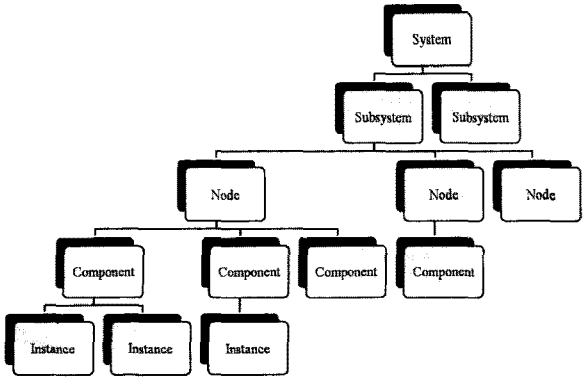



그림 3. JAUS 계층도.

Fig. 3. Hierarchy of JAUS.

표 2. JAUS 계층.

Table 2. JAUS Nomenclature.

| 계층 | 소개 | 비고 |
|---|---|---|
| System | 하위계층에 대해 대표할 수 있는 추상계층 | 최상위 계층 |
| Subsystem | 하나의 무인 체계가 내장되어 있는 개체 | 무인자동차<br>무인잠수정<br>무인비행선 |
| Node | Subsystem 상에서 동작하고 있는 개별적 운영 체제 | 하위 계층과 통신 메시지 트래픽 제어 |
| Component | 실행 가능한 프로세서 또는 소프트웨어 | 단일 기능 수행 |
| Instance | Component와 연결된 센서 단위 | 다중프로세싱 기능 수행 |

표 2는 JAUS의 각 계층에 대한 소개를 하고 있으며, 각 계층은 독립적인 의미를 가지거나 또는 각 계층 별 상호 의존적인 의미를 가지고 있다. System은 전체 아키텍처의 대표할 수 있는 추상적인 의미를 가지며 subsystem은 system의 추상적 목적을 물리적인 존재로 만들 수 있는 어떠한 개체를 뜻하고 있다. Node는 subsystem이 어떠한 목적을 수행하도록 만든 운영 체제라고 말할 수 있으며 단일 프로세서나 유기적으로 연결된 여러 프로세서로 된 운영 체제로 구현되어 있다. 그리고 하위 계층에 있는 component와 instance의 통신 트래픽 흐름을 제어하는 기능 또한 가지고 있다. 각 node들의 하부 계층에는 하나 또는 여러 개의 component와 instance로 구성되어 있으며, 이들은 JAUS 계층에서 최하위 계층으로 구분할 수 있고 응집적 기능 단위로 보통 소프트웨어와 하드웨어(센서 및 구동계 등)로 구현되어 있다.

4. 무인 자율 주행 차량 체계

무인 자율 주행 차량의 구조는 크게 5개의 체계로 나눌 수 있다. 그림 4는 위의 5개 체계를 간략하게 설명한 블록 다이어그램이다.

무인 자율 주행 차량의 가/감속 및 조향 제어를 하며 차량을 직접적으로 통합 제어할 수 있는 차량 체계, GPS와 관성/관계 항법을 이용하여 목적지까지 차량을 유도하는 항법 체

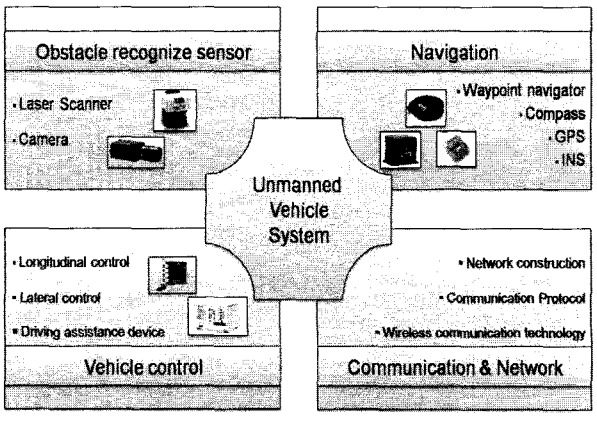



그림 4. 무인 자율 주행 차량의 블록 다이어그램.

Fig. 4. Block diagram of UGV's system.

그림 5. 무인 자율 주행 차량의 모습.
Fig. 5. KUGV(Kookmin Unmanned Ground Vehicle).

계, 목적지로 주행하는 중 장애물 감지 센서(레이저스캐너, 카메라 등)를 이용하여 차량 주변의 장애물 정보를 생성하고 전달하는 장애물인식체계, 원격 및 분산 제어를 위한 네트워크 체계가 있으며, 위의 언급한 체계들의 모든 정보를 취합하여 행위자(behavior)가 우선순위가 정해져 있는 명령을 통해 차량의 최적 주행 상태를 만들어 내는 통합 체계로 나누어진다. 그림 5는 시험 및 검증을 위해 실제 상용 차량을 이용하여 무인자율주행차량의 각 체계 및 센서를 장착한 모습이다.

### 5. JAUS에 따른 무인 자율 주행 차량 아키텍처

앞서 무인 자율 주행 차량의 체계는 크게 5개로 나누어진다고 밝혔다. 본 연구를 위해 개발한 무인 자율 주행 차량의 아키텍처를 JAUS 아키텍처를 이용하여 나누어 보면 그림 6과 같다.

먼저 system 계층의 추상적인 이름 MechaNet이라는 최상위 계층이 존재하며 subsystem 계층에는 물리적 개체들이 존재한다. 하나의 subsystem의 하위 계층에는 총 5개의 node들이 존재하는데 각각 장애물, 항법, 차량 제어 등과 각 node에서 얻을 수 있는 데이터를 규합하는 통합 체계 arbiter, 각 subsystem의 node와 유기적 데이터 통신 트래픽을 총괄하는 node manager가 존재한다. 그리고 각 node별 component와 instance를 살펴 보면, 먼저 장애물 node에는 레이저 센서와 CCD 카메라가 존재하고, 항법 node에서는 GPS와 IMU, 디지털 나침반이 위치한다. 다음으로 차량 제어 node에는 차량의 종/횡 방향을 제어하는 스마트 모터와 비상 정지 계층이 존재하며, 통합 node에서는 전역 경로와 지역 경로 맵을 읽어들여 차량이 최종 목표에 도달할 수 있도록 경로를 생성하는 계획 경로 계층이 존재한다. 그 밖에 각 계층에서 새롭게 필요로 하는 모듈 단위들을 손쉽게 연결할 수 있다.

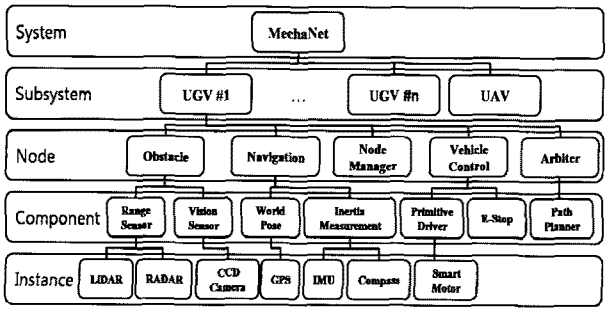



그림 6. 무인 자율 주행 차량의 전체 아키텍처.
Fig. 6. Architecture of UGV's system.

## IV. 무인자율주행차량의 통신 프로토콜

JAUS에서는 무인 체계에 대한 통신 아키텍처 및 통신 메시지에 대한 참조 규약 또한 정의하고 있다. 바로 JAUS volume II인 RA에 part 2, 3이다. 이러한 규약을 정의하는 이유는 크게 두 가지가 있을 수 있다. 첫째는 표준 무인 체계의 규약 메시지를 참조한다면 JAUS 통신 아키텍처를 사용하는 모든 무인 체계는 하나의 JAUS로 묶일 수 있는 동시에 단 하나의 명령으로 전체 체계를 움직일 수 있는 명령 체계로 발전할 수 있다. 두 번째는 각 계층 별 인터페이스가 서로 다르다는 것이다. 즉, 센서 개별의 인터페이스가 아날로그 또는 디지털 신호를 이용하거나 디지털을 사용시 각 표준 별 통신 인터페이스가 다르다는 것이다.

그림 7은 무인 자율 주행 차량에서 사용하고 있는 각종 센서들의 인터페이스를 보여주고 있다. 센서들은 RS232, RS422, IEEE1394, LAN, Analog signal과 같은 물리적인 인터페이스를 이용하여 각 component 또는 node와 통신 연결을 하고 있다[6].

이에 각 instance 또는 component에서 나오는 데이터를 JAUS 메시지 세트를 이용하거나 자체적으로 개발한 통신 메시지 프로토콜을 이용하여 필요한 node에 보낼 수 있는 통신 아키텍처를 설계해야 한다.

### 1. JAUS 통신 표준 규정

JAUS 규약의 volume II인 RA의 part 2를 보면 JAUS의 통신 설계에 대한 명세를 정의하고 있다. 이 부분은 무인 체계의 각 계층을 연결하는 통신 메시지 표준과 메시지 통신에 대한 중요한 참조 및 제시를 서술하고 있다.

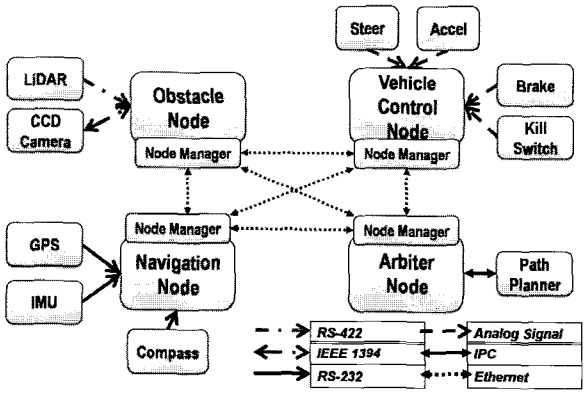



그림 7. 각 node 간 통신망.
Fig. 7. Networking among the node.

표 3. JAUS 통신 표준.
Table 3. Communication standards of JAUS.

| | Description | Note |
|---|---|---|
| 1 | Textual Data Representation | Latin-1 ISO/IEC 8859 Latin-1 Standard character set(Unicode) |
| 2 | Numerical Data Representation | Byte:unsigned 8bits<br>Short Integer: signed 16bits<br>Integer: signed 32bits<br>Long Integer: signed 32bits<br>Unsigned Short/Integer/Long Integer: unsigned 16bits/32bits/32bits<br>Float/Long Float:<br>IEEE signed 32bits /64bits floating num |
| 3 | Byte-Ordering | Little Endian |

2. JAUS 기본 통신 위상

JAUS에서는 기본적인 통신 설계를 그림 8과 같은 위상으로 정의하고 있다. 다수의 subsystem은 하나의 system 안에서 존재하며 각 subsystem들은 subsystem 네트워크로 묶여있다. 즉, 개별적으로 임무를 수행할 수 있는 무인 체계가 다수의 무인 체계, 소대, 분대 또는 그룹으로 무리 지어 임무를 분담할 수 있다는 것이다.

또한, subsystem 네트워크 하위에는 subsystem 안의 node별 통신 기능, node 네트워크로 묶여있다. 각 node들은 자신에 맞는 기능을 수행을 하면서도 다른 계층의 node와의 통신으로 자신이 필요한 정보를 얻을 수 있도록 되어 있다.

이러한 JAUS의 통신 위상은 기본적으로 분산된 컴퓨팅 기능을 수행할 수 있도록 만들어져 있으며 peer-to-peer와 같은 통신 위상을 사용하고 있다.

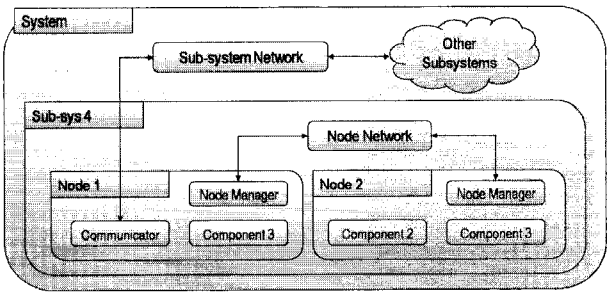



그림 8. JAUS 통신 위상.

Fig. 8. JAUS communication topology.

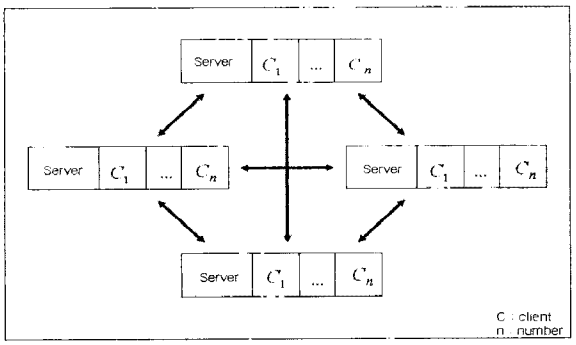



그림 9. 별 형 위상.

Fig. 9. Star topology.

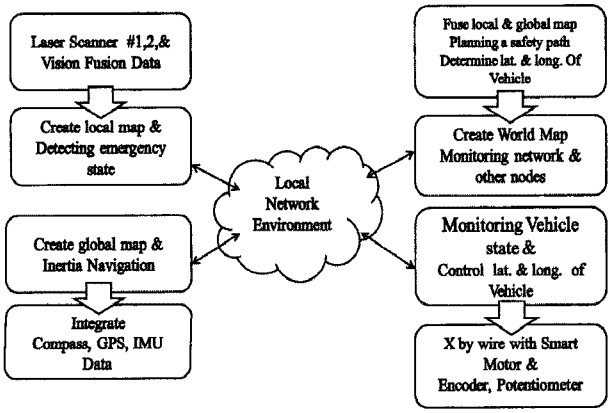



그림 10. 무인 자율 주행 차량의 통신 블록 선도.

Fig. 10. Communication block diagram of UGV.

3. 무인 자율 주행 차량의 통신 위상

JAUS의 기본 통신 위상을 바탕으로 현재 개발한 무인 자율 주행 차량의 통신 위상은 peer-to-peer 방식이 가지고 있는 별형 위상(star topology)을 사용하고 있다. 기본적으로 통신의 연결은 호스트 대 호스트 형식으로 구현이 가능하며, 1대 1 전송이 가능한 유니캐스트 통신 방식, 1대 n전송 방법인 브로드캐스트 통신 방식 그리고 n대 n전송이 가능한 멀티캐스트 통신 방식을 모두 사용이 가능하도록 되어 있다. 이러한 기본 통신 연결 방식을 사용하고 있는 peer-to-peer 방식의 아키텍처 모델에 대한 개념은 많은 곳에서 알려져 있었지만 실제적으로 적용하기에는 아직 유효하지 않았다. 이 개념이 발달한 배경은 통신 인프라의 폭발적 증가였다.

Peer-to-peer 방식은 각 객체간의 동기화와 일관성을 보장하기 위한 통신 아키텍처이다. 각 객체간의 직접 통신 방법을 쓰기 때문에 낮은 지연성과 종단에 연결된 장치의 수에 따라 대역폭을 제한할 수 있고 만약 객체간의 통신 중에 한쪽의 통신이 끊어지더라도 다른 객체들과의 통신은 계속적으로 지속되는 장점을 가지고 있다[7].

무인 자율 주행 차량의 체계는 각 계층별 특성 때문에 연결 방식에도 다른 방법으로 연결이 되어 있어 통합적인 연결 해결 방안이 필요하게 된다. 이에 각 node별 통신은 peer-to-peer와 같은 통신 방식으로 연결되어 있고 그 하위 계층의 연결 방식이 다양하더라도 통합 규약인 JAUS를 이용하여 하나의 연결 방식으로 바꿀 수 있다면 통합적 연결 방안이 해결할 수 있다는 것이다[8].

구체적인 통신 연결 블록선도는 그림 10과 같은 형태를 갖추고 있다.

## V. 통신 프로토콜 및 알고리즘

1. 통신 메시지 헤더

무인 자율 주행 차량에 사용하고 있는 메시지 헤더에 대해서 알아보고자 한다. 메시지 헤더의 크기는 총 9바이트 크기이며, 표 4에 통신 메시지 헤더 필드에 대한 정보를 보여주고 있다.

데이터의 길이 필드는 공용체(union)로 되어 있어 data length_1 필드 및 data length_2로 나뉘어져 있으며 이는 항법 node의 GPS와 IMU 데이터를 동시에 받을 수 있거나 통합 node에서 차량 제어 node로 조향각 및 차량 속도의 값을 보

표 4. 통신 메시지 헤더.

Table 4. Communication message header.

| Field # | Field Description | Type | Size(Byte) |
|---|---|---|---|
| 1 | Message ID | Byte | 3 |
| 2 | Emergency Stop | Byte | 1 |
| 3 | Initialize | Byte | 1 |
| 4 | Status | Byte | 1 |
| 5 | Sequence Num | Byte | 1 |
| 6 | Data Length | Word | 2 |
| 6.1 | Data Length_1 | Byte | 1 |
| 6.2 | Data Length_2 | Byte | 1 |
| | Total Bytes | | 9 |

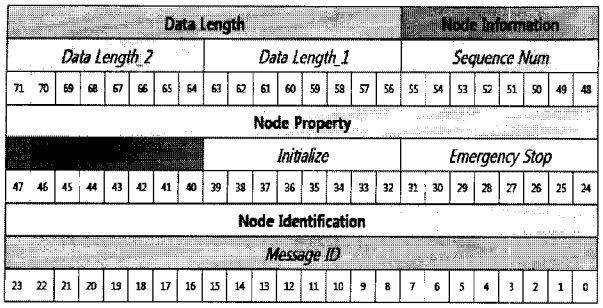



그림 11. 통신 메시지 헤더 필드.
Fig. 11. Communication message header fields.

내도록 설계하였다. 설계된 데이터 헤더의 각 데이터 포맷을 표 4에 정의하였다.

그림 11은 무인 자율 차량에서 쓰인 통신 메시지 헤더 필드를 표현하였다.

2. 통신 알고리즘

무인 자율 주행 차량에 사용하고 있는 통신 방법은 각 계층간 존재하고 있는 통신 프로세스는 미들웨어(Middleware) 방식으로 되어 있으며 하나의 node 상에 있는 component들은 IPC(Inter-Process Communication) 방식을 이용하여 데이터를 유기적으로 송 · 수신하고 있다.

각 node들의 통신 방법은 비동기적으로 통신 데이터를 요청, 요구할 수 있도록 설계되어 있다. 각 node들의 통신 데이터 요청 순환 시간은 50Hz이며, 요청 받은 다른 node들의 데이터 응답은 받는 즉시 이루어진다.

각 node들은 자기 자신에게 연결된 센서들의 데이터들을 한 곳으로 모으거나 각 node들 간 통신이 가능하도록 로컬 네트워킹으로 묶여 있다. 서로 통신을 하고 있는 node중에 가장 빈번하게 데이터를 받고 있는 node는 통합 node이며, 통합 node는 장애물 인식 node와 현재 진행 방향 및 위치를 알려주는 항법 node의 데이터 중에서 우선 순위를 두어 조향각 값과 속도값을 결정하게 된다. 이렇게 결정된 조향각 값과

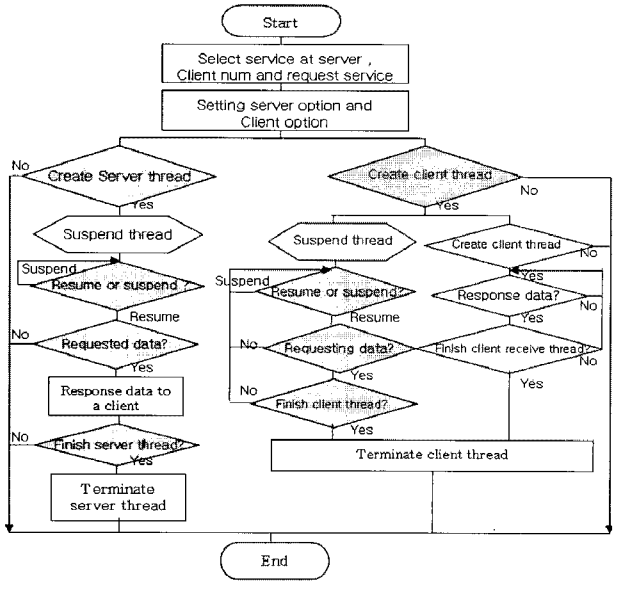



그림 12. 통신 알고리즘의 순서도.
Fig. 12. Flowchart of communication algorithm.

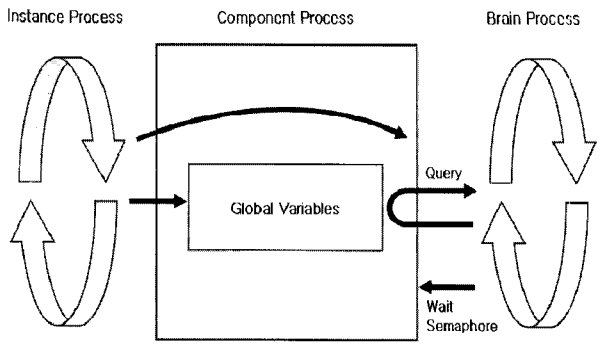



그림 13. 각 node 별 통신 데이터 획득 방법.
Fig. 13. Acquisition method of data among nodes.

속도값은 차량 제어 node로 보내어 차를 제어하게 된다. 다시 차량 제어 node에서 속도값을 통합 node로 피드백하여 현재 속도를 사용자가 볼 수 있도록 한다. 각 node 계층과 component 계층에서 데이터의 획득 방법은 그림 13과 같은 형식으로 얻게 된다.

각 component(sensor process)들은 현재 연결된 센서의 데이터를 계산하여 자기 자신의 상위 계층인 node 계층의 공유 메모리(global memory)에 앞서 계산된 데이터를 채운다. 계산된 데이터는 계속 공유 메모리에 저장하며 다른 node(brain process)의 데이터 요청이 있을 때까지 기다리며(wait) 요청이 있을 시 동기화 블록을 생성 데이터가 공유 메모리에 저장을 막고 요청이 있는 다른 node에 전달하게 된다. 이 획득 방법은 대역폭(Bandwidth)이 한정된 상황에서 지연성(latency)을 낮추는데 가장 효율적인 방식이다.

## VI. 실험 및 결과

1. 개발 환경

본 연구에서 개발한 통신 프로그램은 BSD(Berkeley Software Distribution) 호환 소켓인 윈도우 소켓(winsock)을 이용하여 개발하였다. 버전은 2.2, 윈도우 API를 사용하였으며, 개발 환경은 Windows XP™ 상이다. 네트워크 코어 라이브러리와 미들웨어용 GUI 프로그램을 개발하기 위해 개발 플랫폼인 Visual Studio 2005와 LabVIEW 8.5를 사용하였다.

2. 주행 실험

현재 개발한 무인 자율 주행 차량에는 4개의 각기 다른 node들이 존재한다. 이들 node는 그 하위 계층인 component와 instance 들의 데이터를 받아 계산을 하는데 항법 node는 차량의 진행 방향에 대한 $\theta_{nav}$ 및 RDDF(Route Data Definition File - 무인 자율 주행 차량이 진행할 목표점을 위도와 경도, 그 위 · 경도 좌표까지 진행할 속도값이 일련의 데이터 집합으로 정의되어 있는 파일) 파일 안의 각 경로점(way-point) 별 속도 $v_{nav}$와 장애물 감지 node에서는 차량회피방향에 대한 $\theta_{obs}$, 통합 node 는 이 두 node의 $\theta$ 값 $\theta_{nav}$와 $\theta_{obs}$, 두 가지의 데이터를 통합, 비교해 $\theta_{steering}$ 의 우선순위를 정하며 차량의 속도 또한 $v_{high\text{-}prior}$값으로 결정한다. 통합 node와 연결되어 있는 차량 제어 node는 받아온 조향각과 속도를 유지 시킨다. 다시 차량 제어 node는 통합 node로 현재 조향각과 속도 데이터를 피드백한다.

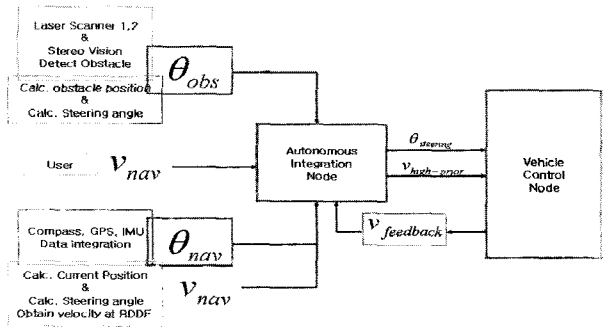



그림 14. 차량주행을 위한 데이터 흐름 블록선도.
Fig. 14. Block diagram for driving vehicle.

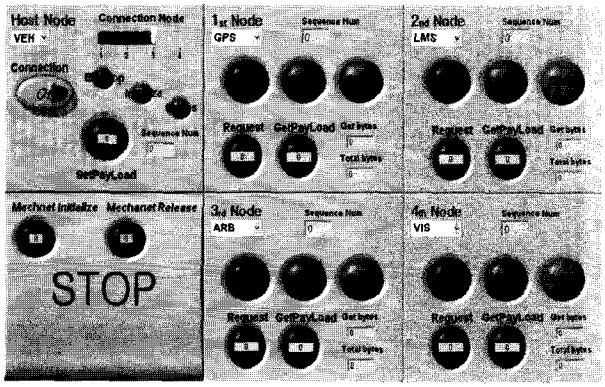



그림 15. 무인 자율 주행 차량 체계의 통신 미들웨어.
Fig. 15. Middleware of UGV system.

그림 14와 같은 흐름은 모두 통신 미들웨어에서 유기적인 제어가 뒷받침되어야 하며, 당연히 각 node간 양방향 통신이 이루어져야 하는 상황이 되어야 한다.

3. 통신 미들웨어

각 node별 상황을 한 눈에 볼 수 있거나 통신 트래픽을 모니터링할 수 있는 통신 미들웨어는 그림 15와 같이 디자인하였다.

로컬 상의 node들은 서로 약속된 어드레스와 포트를 참조하고 있기 때문에 서로 간 패킷 사이즈와 구성만을 알고 있다면 각 node간 연결을 쉽게 하도록 개발되어 있다. 패킷 구성은 JAUS의 표준 메시지 세트를 이용할 수도 있으며, 사용자 임의의 구성 메시지 세트를 이용할 수 있도록 범용적으로 만들어져 있다.

4. 결과

통신 패킷 확인을 위해 상용 프로그램인 TamoSoft© CommView를 사용하여 통신 트래픽 정보를 확인하였다. 그림 16은 무인 자율 주행 차량 동작 시 CommView에서 각 node별 통신 패킷에 대한 정보 및 요청/응답된 데이터 순서와 흐름을 보여주고 있다. 각 패킷의 MAC address session에서의 node별 통신 흐름 방향을 알 수 있다. 또한 node에서 다른 node간의 통신 지연성(delta session)이 최대 1ms 이하로 나오는 것을 확인할 수 있다.

차량이 실제도로상에서 주행 시 전체시스템에서 발생하는 통신의 지연성 및 응답성을 보기 위해 통합 node와 차량 제어 node에서의 조향각 데이터를 주행 제어 요청이 내려진 시점에서 차량의 피드백 값을 통합 node로 응답한 시점까지의

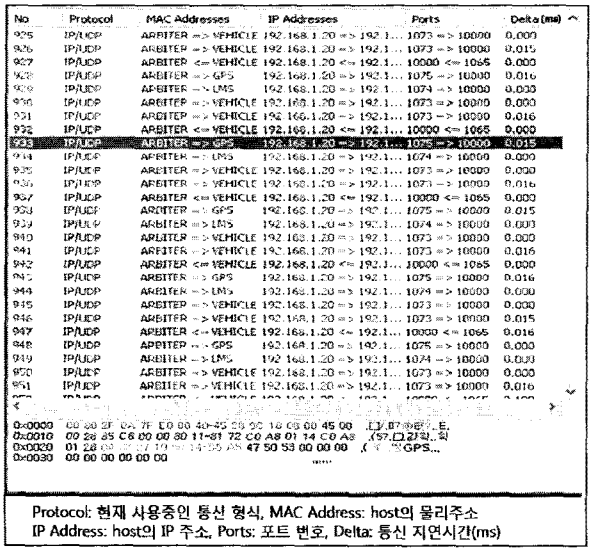



그림 16. 통신 트래픽.
Fig. 16. Communication traffic.

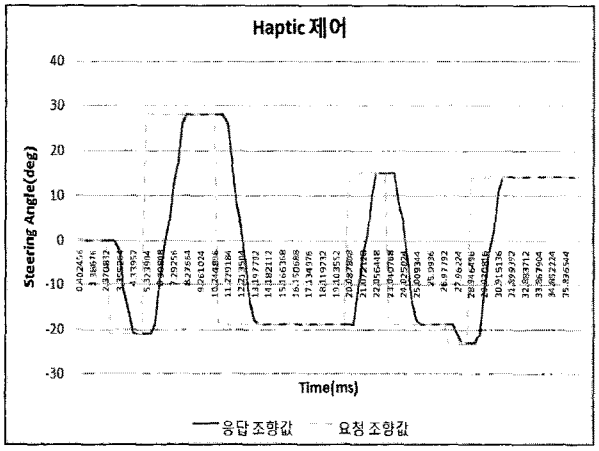



그림 17. 사용자 제어 시 조향값 반응성.
Fig. 17. Response when controlled by user.

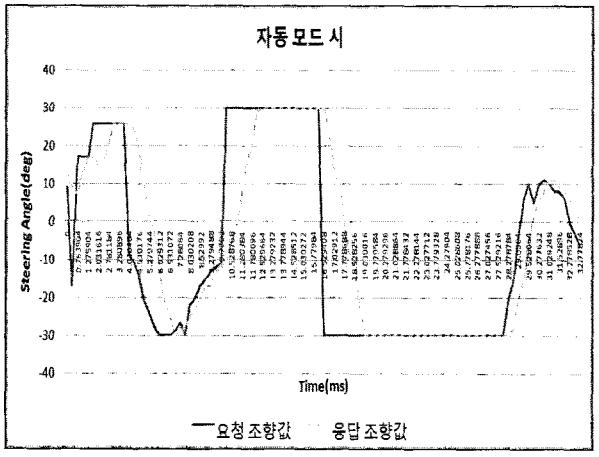



그림 18. 무인 자율 주행 모드 시 조향값 반응성.
Fig. 18. Response when controlled under autonomous.

반응 데이터를 그래프로 도식화하였다. Haptic 컨트롤을 이용하여 반응한 그래프는 그림 17과 같고 실제 무인 자율 주행 모드 시에 반응한 그래프는 그림 18과 같다.

반응 시간은 통합 node에서 명령이 내려진 시간부터 차량이 명령에 반응하는 첫 번째 피드백 값을 받을 때까지 걸린 시간을 말한다. 조향 반응시간은 500ms 이하이며 명령 이후 목표 조향까지의 반응 시간은 차량의 무게와 노면의 거칠기, 타이어의 상태에 의한 반발력을 고려한 시간에 조향 제어 모터의 토크가 걸리는 시간 지연을 합한 총 지연 시간으로 약 2~3초 이내인 것을 그래프 안에서 확인할 수 있다.

현재 진행된 실험 수치 결과 값은 각 node별 데이터 요청 및 응답 수가 평균 초당 159패킷이고 사용한 데이터 크기는 평균 초당 11,161바이트 전송하여 실험을 진행하였다.

## VII. 결론

본 연구에서 개발한 무인 자율 주행 차량의 체계 아키텍처와 통신 아키텍처는 분산 컴퓨팅의 개념에 입각한 JAUS의 규약을 따랐다. 분산된 계층은 각 system 계층, subsystem 계층, node 계층, component 계층, instance 계층으로 나누어진다. 분산된 각 node 는 component 계층에서 받아온 센싱 데이터를 특정 알고리즘을 통해 계산하여 데이터를 가장 빠른 시간 내에 요청한 다른 node 로 보내야 하며, 다른 node 상의 센싱 데이터를 요청 할 때에 빠른 시간 내에 응답 하여야 한다. 이러한 이유로 현재 개발된 네트워크 라이브러리는 각 component나 instance를 위한 소프트웨어에 바로 적용할 수 있도록 미들웨어 형식의 BSD 호환 윈도우 소켓으로 개발하였으며 각 계층은 요청과 응답이 필요 시에만 이루어지기 때문에 통신 지연성이 낮은 장점이 있다.

그러나, 현재 개발된 네트워크 미들웨어는 정적인 등록 상태로 제작되어 새로운 node가 미들웨어로 그룹 등록을 원할 경우 동적으로 등록시킬 수 없다는 단점이 있다. 이는 앞으로 보완해야 할 사항이다.

마지막으로 현재 개발된 무인 자율 주행 차량을 위한 체계 아키텍처와 통신 아키텍처는 JAUS의 규약 버전이 계속 업데이트 되고 있는 상황이므로 앞으로도 업 버전이 개발될 상황이다. 이렇듯, 미국은 무인 기술을 위해 통합적이고 표준화된 JAUS 규약 등과 같은 연구에 많은 투자를 하고 있다. 우리나라도 JAUS와 같은 체계적인 아키텍처와 유기적인 데이터 흐름을 이끌어 낼 수 있는 통신 아키텍처 설계의 독자적인 표준 규범 개발과 인프라 구축이 필요한 상황이며 각 대학 및 연구 기관 그리고 정부가 무인 기술과 관련된 표준 규범을 제정하는데 노력해야 할 것이다.

## 참고문헌


[1] 박용운, 지태영, 최덕선, 김준, 류철형, 박세진, "무인자율차량(XAV)의 구현," 제 13 회 지상 무기 체제 발전 세미나, vol. 11, no. 7, pp. 621-626, 2005.

[2] C. D. Crane III, David G. Armstrong II, "Team CIMAR's navigator: An unmanned ground vehicle for the 2005 DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 8, pp. 599-623, 2006.

[3] DARPA Grand Challenge 2005, Official Site, http://www.darpa.mil/grandchallenge.

[4] The AS-4/Joint Architecture for Unmanned System Working Group (JAUS WG), "Domain model," *JAUS WG*, vol. I, version 3.3, 2007.

[5] The AS-4/Joint Architecture for Unmanned System Working Group, "Reference architecture," *JAUS WG*, vol. II, Part 1, 2, 3, version 3.3, 2007.

[6] H. J. Woo, M. H. Kim, and J. H. Kim, "Development of multiple communication using JAUS message set for unmanned system," *International Conference on Control, Automation and Systems, ICCAS*, pp. 2374-2377, 2007.

[7] H. C. Moon, H. J. Woo, and J. H. Kim, "Peer-to-peer data communication for controlling unmanned system according to the JAUS," *International Conference on Control, Automation and Systems, ICCAS*, pp. 2741-2745, 2006.

[8] S. Singhal, M. Zyda, "Networked Virtual Environments: Design and Implementation," Addison Wesley, 2002.

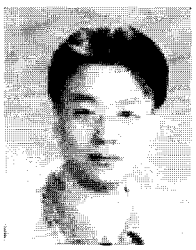


**문 희 창**

1976년 7월 5일생. 2001년 선문대학교 기계설계학과 졸업. 2003년 국민대자동차공학전문대학원 석사. 2003년~현재 동 대학원 박사과정 재학중. 관심분야는 무인자동차, 카메라를 이용한 장애물인식, 마이크로 로봇.

**김 정 하**

1959년 3월 21일생. 1981년 성균관대학교 기계공학과 졸업. 1986년 Univ. of Cincinnati 공학석사. 1990년 Univ. Pennsylvania 공학박사. 1994년~현재 국민대학교 기계자동차공학부 교수. 관심분야는 차량전자제어시스템, 이동로봇, 무인자동차.



**우 훈 제**

1976년 2월 23일생. 2002년 선문대학교 기계공학과 졸업. 2007년 국민대자동차공학전문대학원 석사. 2007년~현재 동 대학원 박사과정 재학중. 관심분야는 무인자동차, 무인시스템을 위한 시스템 아키텍처 설계 및 통신 아키텍처 설계.