

## 저비용 이중화 시스템 기반 교통신호제어 (시스템) 구현

# Implementation of the Traffic Control System based Low Cost Dual Modular Redundancy

이동우<sup>1</sup> · 나종화<sup>1\*</sup> · 김남선<sup>2</sup>

<sup>1</sup>한국항공대학교 항공전자연구소

<sup>2</sup>경찰대학교 치안정책연구소

Dong-Woo Lee<sup>1</sup> · Jong-Whoa Na<sup>1\*</sup> · Nam-Sun Kim<sup>2</sup>

<sup>1</sup>Avionics Research Institute, Korea Aerospace University, Gyeonggi-do, 10540, Korea

<sup>2</sup>Police Science Institute, Korean National Police University, Chungcheongnam-do, 31539, Korea

### [요 약]

본 논문은 교통신호 제어시스템 구축에 적용 가능한 하트비트 기반의 저비용 이중화 시스템을 구현하였다. 교통신호 제어시스템의 고장은 교통 혼란과 교통사고를 야기할 수 있다. 그러므로 고장감내 기술을 도입하여, 교통관제의 안전성과 신뢰성을 확보해야 한다. 이를 위해 오픈소스 하드웨어를 사용하여 이중화 보드를 구성하고, Linux HA를 사용하여 하트비트 기반의 고장 검출 및 복구 소프트웨어를 개발하였다. 교통신호 제어시스템의 기능을 검증하고, 결함주입 시험을 수행하여 장애에 따른 고장복구 시간을 측정하였다. 시험결과 고장복구 시간은 평균9초 이내로 확인되어 신뢰성 목표시간을 만족하는 것을 확인 하였다. 본 연구 결과를 기반으로 항공, 우주, 원자력 등 고 신뢰성 시스템이 요구되는 분야에 응용 될 수 있을 것으로 예상된다.

### [Abstract]

This paper investigates a low cost dual modular redundancy system based on heartbeat which can be applied to traffic control signal system. Failure of the traffic control signal system can cause traffic confusion and traffic accidents. Therefore safety and reliability of traffic control should be secured using fault tolerance technology. To do this, we configured a redundant board using the open source hardware and the heartbeat technique of Linux HA. The function of the traffic signal control system was verified and the fault recovery time was measured using fault injection test. As a result of the test, the fault recovery time was confirmed to be less than 9 seconds on average, confirming that the reliability target time is satisfied. Based on the results of this study, it is expected that it can be applied to fields requiring high reliability systems such as aviation, space, and nuclear power embedded systems.

**Key word** : Fault tolerant, Dual modular redundancy, Traffic control, Heart beat, Fault injection.

<https://doi.org/10.12673/jant.2017.21.5.491>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 3 July 2017; Revised 17 July 2017

Accepted (Publication) 2 September 2017 (30 October 2017)

\*Corresponding Author; Jong-Whoa Na

Tel: +82-2-2209-3671

E-mail: [jwna@kau.ac.kr](mailto:jwna@kau.ac.kr)

## I. 서론

교통 혼잡으로 발생하는 사회적 비용을 최소화하기 위하여, 사물 인터넷(IoT) 기반의 실시간 교통신호 제어시스템 연구가 진행되고 있다[1]-[3]. 실시간 교통신호 제어시스템은 차량검지기로부터 교통상황 정보를 수집하고, 각 교차로의 접근로별 포화도 및 대기차량길이를 추정하여 신호시간을 산정한다. 또한 실시간 교통신호 제어 시스템은 도로의 실제 교통상황에 적합한 신호제어 전략을 수립하여, 차량의 지체시간 및 정지수를 최소화한다. 만약 교통신호 제어시스템에 고장이 발생하면, 신호체계 혼선에 따른 심대한 교통 혼란과 교통사고를 야기할 수 있다. 특히 도로 현장에 설치되는 차량검지기과 교통제어 임베디드 시스템은 우천, 고온 등의 열악한 사용 환경으로 잦은 고장위험에 노출되어 있다. 그러므로 실시간 교통신호 제어시스템은 고장감내 기술을 도입하여, 효율적인 교통관제와 동시에 안전성과 신뢰성을 보장해야 한다.

고장감내 시스템은 특정 모듈의 오동작을 인지하고, 전체 시스템의 가용성과 기능 안전성을 보장한다. 이를 위해 고장감내 시스템은 다중화(redundancy) 기법을 적용하여 지속적인 임무 수행을 가능토록 한다. 다중화 기법은 항공기, 철도, 원자력 분야와 같은 안전 필수(safety-critical) 및 임무 필수(mission-critical) 임베디드 시스템에 필수적으로 적용된다[4]. 다중화 기법은 여유(redundant) 모듈을 투입하므로 개발비용 및 생산비용의 상승 외에도, 소비전력 증가로 운용비용이 증가한다. 개발 및 운용상의 비용 상승은 시스템 고장으로 발생하는 피해 예방 차원에서 상쇄할 수 있다. 그러나 수백에서 수 천 사이트를 운용해야 하는 IoT 시스템의 경우 다중화 기법 적용은 경제적 타당성을 확보하기 어렵다. 이러한 문제를 해결하기 위해서는 저비용으로 구축 가능한 고장감내 기술에 대한 연구가 필요하다.

따라서 본 연구에서는 교통신호 제어시스템 구축에 적용 가능한 하트비트(heart beat) 기반의 저비용 이중화 시스템에 주목하고자 한다. 다중화 기법을 교통신호 제어시스템에 도입하기 위해서는 저렴한 하드웨어 생산비용과 경량구조의 고장감내 운용 소프트웨어가 필요하다. 이를 위해 오픈소스 하드웨어(open source hardware)를 사용하여 이중화 보드를 구성하고, Linux HA를 사용하여 하트비트 검출 및 복구 소프트웨어를 개발하였다. (1)오픈소스 하드웨어는 설계 내역이 공개되어 있으므로 개발 목적과 기능에 맞게 커스터마이징이 가능하다. 특히, 불필요한 하드웨어 구성요소를 제거하거나, 특정 모듈을 강화하여, 경제성과 신뢰성을 고려한 최적화된 고장감내 시스템 구축이 가능하다. 또한 하드웨어는 기술에 대한 특허권이 없기 때문에, 교통 IoT 시스템의 구성과 기능에 따라, 적합한 형태의 이중화 하드웨어를 저비용으로 구현할 수 있다. (2) Linux HA[5]는 시스템 수준에서 고장을 감지할 수 있는 경량구조의 하트비트 메커니즘을 지원한다. 하트 비트는 시스템의 상태를 검사하기 위해 일정 간격마다 발생하는 관제 신호이다. 경량 구조의 하트비트 메커니즘 도입으로, 제한되어 있는 시스템 자원을 임

무수행에 집중할 수 있고, 시스템의 휴대성(portability)과 확장성(scalability)을 확보 할 수 있다[6]-[7].

본 연구는 교통신호 제어시스템의 데이터 수집 게이트웨이를 이중화 보드로 개발하고, 고장발생에 따른 장애극복(failover) 성능을 분석하였다. 이중화 게이트웨이 시스템은 액티브 모듈과 워 스탠바이 모듈로 구성한다. 각 모듈은 오픈 소스 하드웨어인 Raspberry Pi[8]를 사용하여 저비용 시스템으로 구축 하였다. 액티브 모듈과 워 스탠바이 모듈은 일정시간 간격으로 하트비트 신호를 주고받으며, 시스템 고장유무를 감시한다. 액티브 모듈의 고장이 발생하면, 워 스탠바이 모듈은 대기(idle) 상태에서 활성(active) 상태로 전환되어 임무를 수행한다. 교통신호 제어시스템의 장애로 발생하는 차량 대기시간을 분석하기 위한 방안으로 장애 극복 시간을 측정하였다. 차량의 대기시간은 도로의 환경 시나리오, 접근 차량 수, 신호 관리체계 등 다양한 요인에 의해 변화하므로, 정량적인 비교는 쉽지 않다. 그러나 최근 연구된 교통신호 시스템은 10-60 초의 차량 대기시간을 제시하고 있다[9-10]. 이를 근거로 교통신호 제어시스템의 장애에 따른 차량 대기시간은 10초 이내로 설정하였다. 교통신호 제어시스템의 기능을 검증하고, 장애 극복 시간을 측정하기 위해서 이중화 프로세스를 대상으로 임의적인 장애를 생성하였다. 시험결과 고장복구 시간은 평균9초 이내로 확인되어 신뢰성 목표시간을 만족하는 것을 확인 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 교통 IoT 시스템의 발전 방향과 고장감내 기술 동향을 설명하고, 3장에서는 고장감내 교통제어 시스템을 구축하기 위한 이중화 게이트웨이 시스템을 설명한다. 4장에서는 이중화 게이트웨이 시스템의 고장감내 기능을 확인하기 위한 시험환경 및 시험 방법, 그리고 고장감내 성능을 설명하고, 5장에서는 결론 및 향후 연구에 대해 기술한다.

## II. 교통 IoT 시스템 기술동향

교통신호 제어시스템의 기본 목적은 교통상황 검지정보를 활용하여 접근로별 차량의 수요를 적정하게 배분하고 차량의 지체시간 및 정지수를 최소화 하는 것이다. 교통신호 제어시스템은 효율적인 제어전략을 수립하기 위해, 교통상황을 인지하고, 도로의 포화도 및 대기차량을 고려하여 신호시간을 산정한다. 교통신호제어기는 (1) 일반신호제어기, (2) 전자신호제어기, (3) 실시간 신호제어기로 구분할 수 있으며, 각 신호제어기의 기능과 특성은 표1로 정리 하였다.

IT 기술의 발달로 다양한 형태의 교통정보 수집시스템이 구축되고 있다. 기존 차량검지기는 교통량, 차량 속도 및 도로 점유율과 같은 특정 지점의 정보만을 수집해 왔다. 근래에는 개별 차량의 구간통행시간 정보를 취득할 수 있는 UTIS, DSRC(하이패스)등의 교통정보 수집시스템이 광범위하게 구축되고 있다.

표 1. 교통신호제어기의 유형과 운용특성

Table 1. Traffic signal controller types and operational characteristic.

Traffic signal controller type	Operational characteristic
Normal traffic signal controller	Distributed and fixed time Control system ◦ These system can be operate in conjunction with single or adjoining intersection. ◦ These system can not be operated by a central computer. (Off-line network)
Electro-mechanical traffic signal controller	Central control system ◦ These system consist of intersection traffic signals, a communications network to tie them together, and a central computer or network of computers to manage the system. ◦ These system can be operated by a central computer. (On-line network)
Real-time traffic signal controller	Real time control System. ◦ These system is the most progressive system that operates in real-time, and adjusts signal timings based on detected data in real-time. ◦ These system can be operated by a central computer.

향후 교통운영체계는 도로와 자동차 및 보행자간의 다양한 수집 체계를 통한 실시간 구간검지 체계 중심의 차세대 지능형 교통체계(C-ITS : co-operative ITS)로 진화할 것으로 예상되며, 전 세계적으로 V2X 기반의 신호제어시스템 개발이 가속화 되고 있다. 그러나 차세대 지능형교통체계 기반 신호제어시스템 구축에 10년 이상이 소요될 것을 고려할 때, 기존 장비 및 기술을 활용한 신호운영시스템의 기능고도화가 필요한 상황이다.

기존의 교통 시스템의 성능을 개선하기 위하여 전자, 제어, 정보 통신기술을 융합한(ITS; intelligence transport system) 기술이 지속적으로 연구되고 있다[2]. 일례로 스마트신호제어시스템은 차량검출 센서, 무인항공기, 빅 데이터 기법을 사용하여 효율적으로 교통 상황을 관리한다[6]. 스마트신호제어시스템의 운용 환경은 그림1로 확인할 수 있다. 이 외에도 교통 상황을 감시 및 관리하기 위한 다양한 기능의 교통 임베디드 시스템이 개발되고 있다. 대표적으로 교차로 알리미, 과속경보 표지판, 무선차량 검출기 등이 배치되어 운용되고 있다.

교통 임베디드 시스템의 임무는 교통 흐름을 감시하고 관리하는 것으로, 고장이 발생하면 교통장애 외에도 심각한 인적/물적 손실이 발생할 수 있다. 그러므로 교통 임베디드 시스템은 고 신뢰성 시스템 구조가 요구되며, 이 과정에서 높은 개발/제작, 운용비용이 예상된다. 그러나 교통 시스템을 담당하는 일부 공공기관에서는 예산부족을 이유로 시스템 설치를 지연하거나, 신뢰성을 보장할 수 없는 시스템을 설치하기도 한다. 이로 인해 교통시스템은 잦은 고장 및 사고 위험이 증가한다. 이러한 문제를 해결하기 위해서 저비용 고 신뢰성 시스템 기술을 연구하여, 제품의 가격은 낮추면서도 장기간 운영할 수 있는 교통신호 제어시스템 관련 플랫폼 기술의 확보가 필요하다. 본 연구에서는 서버 등의 장비에서 활용하는 이중화 기술을 open source 하드웨어를 이용하여 저비용 고 신뢰성 시스템 구현을 지원한다.

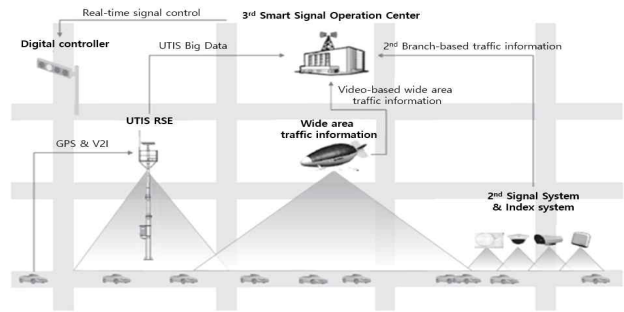


그림 1. 스마트 신호제어 시스템 운용환경[1]

Fig. 1. Smart signal control system operating environment.

### III. 고 신뢰성 교통 IoT 시스템

#### 3-1 고 신뢰성 교통 IoT 시스템 구성

교통 IoT 시스템은 현재의 신호제어시스템뿐만 아니라 차세대 지능형 교통체계(C-ITS: Co-operative ITS)의 일환으로, 실시간 교통상황 파악과 신호제어 기능을 수행한다. 또한 시스템에서 수집되는 도로의 교통정보를 중앙관제센터에서 통합하고, 도로 간 연계성을 고려한 광역 교통흐름 관리를 수행할 수 있다. 교통 IoT 시스템은 (1) 도로의 교통상황을 파악하는 검지 시스템, (2) 차량의 흐름을 제어하는 신호등 시스템, (3) 교통 혼잡도에 따른 신호를 생성하는 디지털 제어기, (4) 지역 도로의 교통상황을 중앙관제센터로 전송하는 게이트웨이로 구성된다. 교통 IoT 시스템의 세부 구성은 그림 2에서 확인할 수 있다. 도로상의 각 지점에 설치되어 있는 검지 센서는 교차로의 방향별 교통량과 접근속도를 측정하고, 그 결과를 디지털 제어기에 전달한다. 디지털 제어기는 각 검지 센서의 정보를 기반으로 도로의 혼잡도를 분석하고, 원활한 교통 흐름을 유도하기 위한 최적의 신호제어 정보를 산출한다. 산출된 신호제어 정보는 교통 IoT 시스템을 통해 신호등 시스템에 전달되어 실제 교통 흐름을 제어한다. 교통 IoT 시스템은 교통 흐름 정보와 신호제어 정보를 게이트웨이를 통해 중앙관리 센터로 전송한다. 중앙관제센터는 각 지역의 교통정보를 통합하여, 광역 교통상황을 파악하고 교통흐름을 제어한다.

교통 IoT 시스템의 열악한 운용 환경(온도, 습도, 진동 등)은 시스템의 수명을 저해하고, 빈번한 고장의 원인이 된다. 교통 IoT 시스템은 차량의 흐름을 실시간으로 관리하므로, 장애가 발생하면 교통상황에 혼란을 초래할 수도 있다. 그러므로 시스템의 신뢰성 및 가용성을 보장할 수 있는 고장감내 기법을 교통 IoT 시스템에 도입해야 한다. 교통 IoT 시스템에 다중화 기법을 적용하면, 고장 상태 검출과 자체적인 복구로 기능의 연속성을 보장할 수 있다. 다중화 기법은 시스템의 장애를 극복하기 위해서, 동일한 기능을 수행하는 복수의 시스템을 구축한다.

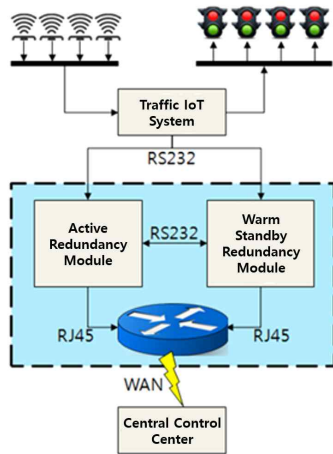


그림 2. 교통 IoT 시스템 구성도  
Fig. 2. Traffic IoT system configuration diagram.

그림2 에서는 교통 IoT 시스템과 중앙관제센터 사이의 게이트웨이가 이중화 되어 있는 것을 확인 할 수 있다. 교통 IoT 시스템을 구성하는 모든 하위 시스템은 고 신뢰성 기법 적용이 필요하지만, 본 연구에서는 게이트웨이를 대상으로 제한적으로 고 신뢰성 기법을 적용하였다.

### 3-2 이중화 게이트웨이 시스템

이중화 게이트웨이 시스템은 동일한 기능을 수행하는 두 개의 모듈이 액티브와 워 스탠바이 형태로 구성된다. 액티브모듈의 정상상태에서, 워 스탠바이는 액티브모듈의 상태를 백업하며 장애 상태를 감시한다. 액티브모듈에 장애가 발생하면, 워 스탠바이는 게이트웨이 기능을 구동한다.

액티브모듈과 워 스탠바이 모듈은 하트비트 프로세스, 미션 프로세스, 컨트롤 프로세스로 구동된다. 첫 번째로 하트비트 프로세스는 모듈의 장애 발생 여부를 검출한다. 두 번째로 미션 프로세스는 디지털 제어기에서 전달받은 교통상황과 워 스탠바이 모듈의 동작 상태를 중앙관제 센터로 전달한다. 세 번째로 컨트롤 프로세스는 미션 프로세스의 활성화(active)/활성대기(standby) 전환을 관리하고, 데이터 마이그레이션(migration)을 수행한다. 액티브모듈과 워 스탠바이 모듈의 프로세스 구성과 기능이 동일하므로, 서로의 임무를 교차하여 운용 할 수 있다. 전체 시스템과 프로세스의 구성은 그림 3에서 확인 할 수 있다. 그림은 이중화 게이트웨이 시스템의 동작절차를 설명한다.

이중화 게이트웨이는 수행 내역에 따라 (1) 데이터 수집, (2) 고장 검출, (3) 임무 수행, (4) 고장 복구 기능으로 구분한다. 첫 번째로 데이터 수집은 디지털 제어기로부터 검지 센서와 신호 등 상태정보를 수집하는 단계이다. 게이트웨이는 지역 교통 상황과 신호제어 정보를 실시간으로 수집하고, 중앙관제 센터로 전달한다(①).

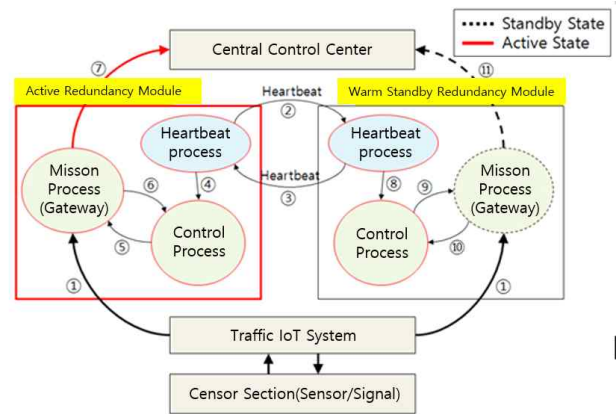


그림 3. 이중화 게이트웨이 시스템  
Fig. 3. Dual redundant gateway system.

이때, 교통정보는 이중화 시스템의 액티브와 워 스탠바이 모듈에 동시에 전달된다. 두 번째로 고장 검출은 하트비트 프로세스와 공유메모리 기능을 사용하여, 각 모듈의 고장유무를 확인한다. 이중화 게이트웨이는 하드웨어 고장과 프로세스 고장을 검출할 수 있다. 하드웨어 고장은 하트비트를 사용하여, 각 모듈이 상대방 모듈의 상태를 확인한다(②, ③). 이때 하트비트의 발행 간격은 100 ms부터 100 ms 간격으로 설정할 수 있다. 하트비트 프로세스는 특정 모듈에 장애가 발생하여 하트비트 발행이 수신되지 않으면, 고장발생으로 판정하고, 그 결과를 컨트롤 프로세스에 통보한다(④, ⑧). 프로세스 고장은 각 모듈 내부의 프로세스의 동작 상태를 확인하는 것으로, 각 프로세스가 공유 메모리에 남긴 풋프린팅(foot printing) 정보를 분석한다(⑤, ⑥, ⑨, ⑩). 세 번째로 임무 수행은 액티브모듈이 정상 상태에서 미션 프로세스가 교통정보를 관제 센터로 전송하는 한다(⑦). 고장 복구는 액티브 모듈의 장애가 발생하면, 워 스탠바이 모듈의 컨트롤 프로세스가 미션 프로세스를 활성화 하여 교통 관제 서비스를 지속적으로 유지한다(⑪).

#### 1) 이중화 게이트웨이 하드웨어

이중화 게이트웨이의 하드웨어는 액티브모듈, 워 스탠바이 모듈, router로 구성된다. 임무를 수행하는 액티브모듈과 워 스탠바이 모듈은 오픈 소스 하드웨어인 Raspberry Pi B+ 보드를 사용하였고, router는 ipTIME네트워크 공유기를 사용하였다. Raspberry Pi B+는 700 Mhz ARM1176jzf-S싱글코어를 탑재하고 있으며, 512MB의 SDRAM, LAN9514 이더넷/USB2.0 컨트롤러, USB 4포트, 이더넷 1포트, 17개의 GPIO 포트가 구성된다. Raspberry Pi B+는 Raspbian LINUX O/S 탑재하고 있다. Raspberry Pi B+보드를 사용한 이중화 게이트웨이 하드웨어 구성은 그림 4로 확인 할 수 있다.



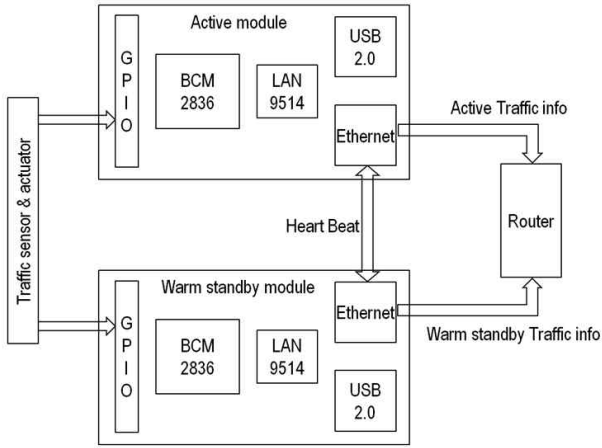


그림 4. 이중화 게이트웨이 하드웨어 구성도  
 Fig. 4. Dual redundant gateway hardware configuration.

액티브모듈과 워 스탠바이 모듈은 각각 GPIO와 이더넷을 사용하여 외부 장치와 연동하도록 개발하였다. 첫 번째는 차량 검지기의 센서 정보와 신호상태를 수신 받는 GPIO 이다. 차량 검지기는 도로의 차량 유무를 검지하는 센서로 차량이 검출되면 'high', 미 검출 되면 'low' 값이 발생토록 설정하였다. 신호 상태는 차량제어를 위한 값으로 차량 통과 신호(청색)는 'high', 차량 정지 신호(적색)는 'low' 값을 출력한다. 이중화 게이트웨이는 GPIO에서 입력된 교통정보를 통합하여, 관제센터로 전달한다. 두 번째는 하트비트 신호를 주고받는 이더넷 이다. 이중화 게이트웨이는 하트비트 신호를 주고받으면서, 각 모듈의 고장 상태를 확인한다. 하트비트 신호는 UART 프로토콜을 사용하여, 빠르고 반복적인 송수신이 가능하도록 개발하였다. 마지막으로 교통정보를 관제센터로 전달하는 이더넷 이다. 교통정보는 TCP 프로토콜을 사용하여, 데이터 전송의 신뢰성을 보장하였다. Raspberry Pi B+는 한 개의 이더넷 포트를 지원하므로, 하트비트 신호와 교통정보는 같은 이더넷 포트를 사용한다. 그러므로 router를 사용하여 LAN(local area network)을 구성하고, 액티브 모듈과 워 스탠바이 모듈, 그리고 관제센터의 이더넷 포트를 연결한다.

2) 이중화 게이트웨이 소프트웨어

이중화 게이트웨이 시스템은 하트비트 프로세스, 미션 프로세스, 컨트롤 프로세스로 구동된다. 각각의 프로세스는 공유메모리를 사용하여, 상호간의 임무 상태를 확인 할 수 있다. 이중화 게이트웨이의 액티브 모듈은 미션 프로세스가 구동중인 모듈로써, 액티브 모듈과 워 스탠바이 모듈을 교차하며 운용할 수 있도록 개발하였다. 각 프로세서의 기능과 운용 절차는 다음과 같다.

1	Start Heart-Beat Process
2	Initialize Linux HA, shared memory, UDP
3	While(End of HB process) begin
4	Send to heart beat to (warm standby/active)
5	Listen to heart beat form (warm standby/active)
6	If (normality Heart Beat) than
7	Write normality heart beat state to shared memory
8	Else (Time Out Heart Beat) than
9	Write failure to Shared memory
10	If (failure module == active) than
11	Start failover procedure
12	Wait(200ms) //next heart beat term
13	end

그림 5. 하트비트 프로세스 유사코드  
 Fig. 5. Heartbeat process pseudo code.

(1) 하트비트 프로세스

하트비트 프로세스는 이중화 게이트웨이 시스템의 모듈 상태를 확인하고, 고장을 판정한다. 하트비트 프로세서는 R. Alan의 Linux-HA 프로젝트를 기반으로 개발하였다[5]. Linux-HA는 클러스터 인프라를 구축하기 위한 서비스 데몬으로 시스템 점검 기능과 IP takeover 기능을 탑재하고 있다. 본 연구에서는 Linux-HA를 사용하여 이중화 게이트웨이의 모듈 동작상태를 모니터링 한다. Linux-HA는 자체적인 설정파일을 사용하여, (1) 하트비트 신호간격, (2) 고장판정 시간, (3) 라우터 IP, (4) 통신 방식, (5) IP와 통신 포트를 설정할 수 있다. 그림 5은 하트비트 프로세서의 구동절차를 유사코드로 설명한다. 하트비트 프로세스는 시작과 동시에 시스템 구동에 필요한 운용 서비스와 메모리를 초기화 한다. 구체적으로는 (1) 하트비트 서비스를 위한 Linux-HA 서비스 데몬 구동, (2) 하트비트에 의한 모니터링 정보를 기록하기 위한 shared memory 할당, (3) 하트비트 신호를 주고받을 UDP 소켓을 생성한다(line 2). 모든 초기화 작업이 정상적으로 완료되면, 프로세스가 종료 시까지 하트비트 신호의 송수신을 반복한다. 하트비트 신호는 비동기식의 UDP 프로토콜을 사용하고, 전송용 포트와 수신용 포트를 별도로 할당하여, 이중화 보드 간의 하트비트 신호 충돌을 방지한다. 하트비트 신호의 전송과 수신이 정상적으로 이루어지면, 공유 메모리에 기록하여, 모듈의 상태를 유지한다. 그러나 하트비트 신호가 수신되지 않는 경우, 상대방 모듈에 기계적 또는 전자적인 장애가 발생한 것으로 판정하고 공유 메모리에 고장 상태를 기록한다. 고장 정보는 컨트롤 프로세스에게 전달되어 장애 극복 절차를 진행한다.

(2) 미션 프로세스

미션 프로세스는 도로 교통상황을 수집하고, 통합하여 관제센터로 전달하는 기능을 수행한다. 미션 프로세스의 구동 절차는 그림 6의 유사코드로 설명한다. 미션 프로세스는 시작과 동시에 프로세스 구동에 필요한 서비스와 메모리를 초기화 한다. 초기화는 (1) 관제 센터와 데이터 통신하기 위한 TCP 소켓 생성

1	Start Mission Process
2	Initialize shared memory, TCP, GPIO
3	While(End of Mission process) begin
4	Receive traffic data from sensor & actuator
5	If (Active module) than
6	Transmit traffic data to monitor center
7	Else (Warm standby module) than
8	IF (active module failure) than
9	Transmit traffic data to monitor center
10	Else
11	Discard traffic data from sensor
12	Write normality Mission process state
13	end

그림 6. 미션 프로세스 유사코드

Fig. 6. Mission process pseudo code.

(2) 교통센서의 정보를 수집하기 위한 GPIO 포트 설정, (3) 미션 프로세서의 동작 상태를 기록하기 위한 공유 메모리 할당 작업을 수행한다. 모든 초기화 작업이 정상적으로 완료되면, 프로세스가 종료될 될 때까지 교통관제 정보를 관제센터로 전송하는 기능을 수행한다. 미션 프로세스는 GPIO 포트에 연결되어 있는 다수의 교통 센서의 정보를 라운드 로빈 방식으로 수집한다. 모든 교통 센서의 정보가 수집되면, 액티브 모듈은 센서 정보를 관제 센터로 전송한다. 반면, 워밍 스탠바이 모듈은 센서 정보 수신 후, 컨트롤 프로세스의 공유 메모리를 확인하여 액티브 모듈의 고장 유무를 확인한다. 액티브 모듈이 정상상태이면, 수신된 센서 정보는 폐기된다. 그러나 액티브 모듈이 고장상태이면 교통관제 정보를 관제 센터로 전달하는 작업을 수행한다. 미션 프로세스는 매 회 교통정보를 관제센터로 전달한 후, 공유 메모리에 자신의 상태를 기록한다. 컨트롤 프로세스는 미션 프로세스가 공유 메모리에 기록한 정보를 기반으로 고장유무를 판정한다.

**(3) 컨트롤 프로세스**

컨트롤 프로세스는 하트비트 프로세스와 미션 프로세스의 상태를 확인하고, 장애 발생 시 장애 극복 절차를 수행한다. 컨트롤 프로세스의 구동 절차는 그림 7의 유사코드로 확인 할 수 있다. 컨트롤 프로세스는 시작과 동시에 프로세스 구동에 필요한 메모리를 초기화 한다. 초기화는 (1) 하트비트 프로세스의 상태를 확인하고, 제어하기 위한 공유 메모리 할당, (2) 미션 프로세스의 상태를 확인하고 제어하기 위한 공유메모리 할당작업을 수행한다. 모든 초기화 작업이 정상적으로 완료되면, 프로세스가 종료될 될 때까지 하트비트 프로세스와 미션 프로세스를 모니터링하고, 고장을 검출한다. 하트비트 프로세스와 미션 프로세스의 공유 메모리에는 각 프로세스의 동작 이력이 기록되어 있다. 동작 이력에는 마지막 기록 시간과 수행 내역이 저장되어 있다. 동작 이력 정보를 분석하면, 세 가지 고장 상황을 검출 할 수 있다. 첫 번째는 하트비트프로세서의 이력 정보로 이중화 보드 모듈(active or warm standby)의 하드웨어적인

1	Start Control Process
2	Initialize shared memory
3	While(End of Control process) begin
4	Read shard memory about Heartbeat Process
5	Read shard memory about Mission Process
6	If(failure Heartbeat or Mission Process) than
7	If (Active module) than
8	Notify failure to Heartbeat and Mission process
9	Else (warm standby) than
10	Active Heartbeat and mission process.
11	end

그림 7. 컨트롤 프로세스 유사코드

Fig. 7. Control process pseudo code.

고장 상태를 파악할 수 있다. 두 번째는 모듈 내부의 프로세스들의 dead lock 또는 live lock에 의한 고장상태를 검출 할 수 있다. 마지막으로 미션 프로세스의 교통관제정보를 분석하여 미션 프로세스 또는 교통 센서부의 고장을 검출 할 수 있다. 컨트롤 프로세스는 고장이 검출되면 공유메모리를 사용하여 하트비트 프로세스와 미션 프로세스에게 장애 극복 기능 활성화를 통보한다.

**IV. 이중화 게이트웨이 성능평가**

이중화 게이트웨이의 성능은 장애 극복 시간으로 측정 할 수 있다. 장애 극복 시간은 시스템이 고장을 인식하고, 복구하기까지 소요되는 평균 시간이다. 본 연구에서는 고장상황을 정의하고, 각 고장 상황에서 시스템의 장애 극복 시간을 측정하였다. 이를 위해 첫 번째로 이중화 게이트웨이 보드의 안전성을 시험하기 위해 모의 교통관제 시스템을 구축하였다. 두 번째로 이중화 보드에서 발생할 수 있는 고장 상황을 정의하고, 시험하였다. 마지막으로 각 고장상황에서 장애 극복 시간을 측정하여 이중화 게이트웨이 시스템의 성능을 평가하였다.

**4-1 시험환경 구성**

모의 교통관제 시스템은 차량검지기, 신호등, 신호제어기(base station)로 구성된 사거리 모형으로 제작하였다. 차량검지기는 자성을 감지하는 홀 센서를 사용하였다. 홀 센서는 모형 도로로 위에 RC 자동차가 지나가면 자기장 변화를 감지한다. 홀 센서의 아날로그 신호를 디지털 신호로 변조하기 위하여 ADC(analog to digital) 보드를 개발하였다. ADC 보드는 디지털 신호로 변환된 차량검지기의 정보를 신호제어기로 전달한다. 신호제어기는 차량검지기의 정보를 사용하여 신호등 제어 신호를 생성한다. 신호제어기는 교통제어 신호를 신호등으로 전달하여 LED의 점멸을 제어한다. 교통 IoT 시스템의 신호제어기(Base Station)는 실시간으로 센서/신호 정보를 이중화 게이트웨이 시스템으로 전달한다. 교통 IoT 시스템의 전체 구성도 그림 8에서 확인 할 수 있다.

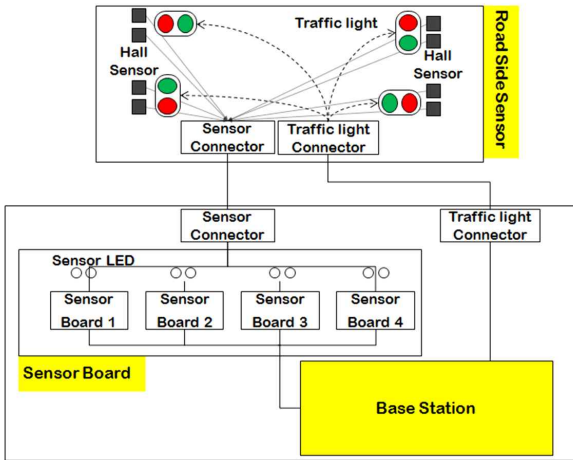


그림 8. 모의 교통관제 시스템 구성도  
 Fig. 8. Traffic control simulation model diagram.

중앙관제센터의 역할을 하는 프로그램은 이중화 게이트웨이 시스템으로부터 센서 및 신호 정보와 고장유무 상태 정보를 전달받는다. 모의 교통관제 시스템의 센서와 신호 상태 정보는 중앙관제센터의 프로그램에 실시간으로 갱신되어 관리자가 확인할 수 있도록 구성하였다. 그림 9은 교통 IoT 중앙관제 프로그램의 GUI이다. 그림 9의 (a)는 이중화 게이트웨이 시스템 모듈의 미션 수행상태를 나타낸다. (b)는 두 모듈 간의 하트비트 감시상태를 나타낸다. 모의 교통관제 시스템에 설치되어 있는 센서와 신호등의 상태는 (c)와 (d)에서 확인할 수 있다. (c)는 차량감지 센서의 상태로, 차량이 검출되면 ‘Detected’, 차량이 검출되지 않으면 ‘Nothing’으로 출력된다. (d)는 모의 사거리에 설치되는 신호등의 제어상태를 출력한다. (e)는 교통 IoT 시스템에서 전달받은 데이터의 수신 시간과 값을 실시간으로 기록하며, (f)는 신호등의 제어상태가 변경될 때마다 신호등 색을 변경하여 신호등 제어상태를 시각적으로 표현한다. 모의 교통관제 시스템의 실제 시험환경은 그림 10에서 확인할 수 있다.

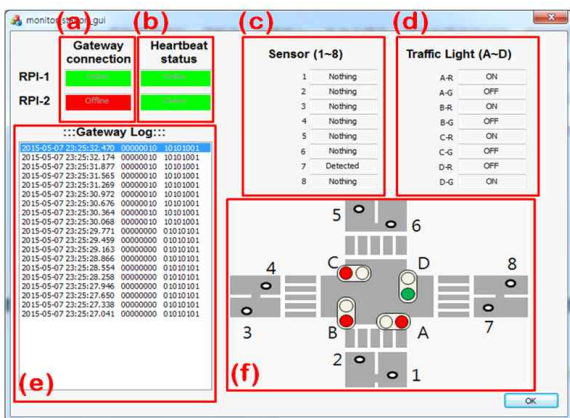


그림 9. 중앙관제 소프트웨어 GUI  
 Fig. 9. Central control software GUI.

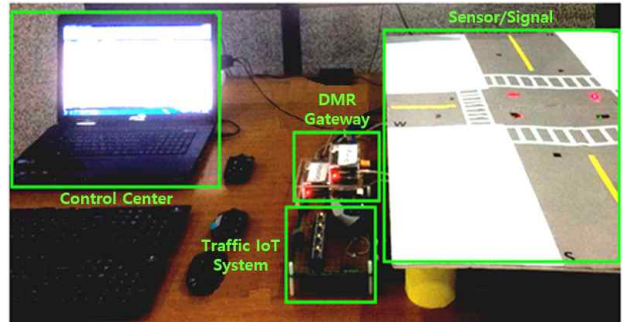


그림 10. 모의 교통관제 시스템 환경  
 Fig. 10. Traffic control simulation system.

#### 4-2 결함주입 시험절차

이중화 게이트웨이 시스템의 장애 극복 시간 측정은 액티브 모듈에 고장이 발생 한 후, 웹 스탠바이 모듈에서 임무를 재개 하는데 걸리는 시간을 측정한다. 장애 극복 시간을 측정하기 위한 절차는 다음과 같다.

##### (Step 1): 결함모델 선정

이중화 게이트웨이 시스템에 주입할 결함모델을 선정한다. 이중화 게이트웨이의 고장검출 및 복구 성능을 관찰하기 위해서는, 시스템의 기능을 지속할 수 없는 고장이 발생해야 한다. 이를 위해, 액티브 모듈의 전원 고장과 미션 프로세스의 dead lock 고장 시험을 결함 모델로 선정하였다. 액티브 모듈에 전원 고장이 발생하면 시스템의 모든 프로세스가 중단된다. 이 때 웹 스탠바이는 하트비트 신호로 액티브 모듈의 고장을 검출할 수 있다. 이러한 특성을 고려하여 액티브 모듈의 하트비트 고장 상황을 액티브 모듈의 전원고장 상황으로 대신할 수 있다. 프로세스의 고장은 “kill -9” 명령어를 사용하여 고장이 발생하는 프로세스를 강제 종료하는 방식으로 구현하였다.

##### (Step 2): 결함주입 스크립트 작성

결함주입을 실행하고, 고장 검출과 복구 시간을 측정하는 스크립트 파일을 작성한다. 스크립트 파일은 (1) 결함주입 직전 시간을 로그 파일에 저장, (2) kill -9 명령어를 사용하여, 하트비트 프로세스 혹은 미션프로세스를 강제로 종료하는 명령어를 포함한다. 고장검출시간은 웹 스탠바이 모듈의 컨트롤 프로세스가 기록한다.

##### (Step 3): 모의 교통관제 시스템 구동

모의 교통관제 시스템 시험환경을 구동하고, 결함주입 대기 상태를 유지한다. 이중화 게이트웨이는 컨트롤 프로세스, 하트비트 프로세스, 미션 프로세스 순으로 구동한다. 관제 센터의 관제 프로그램을 실행한 후, 교통검출기 센서와 신호 정보가 정상적으로 수신되는 지를 확인한다.

##### (Step 4): 결함 주입

모의 교통관제 시스템이 구동중인 상태에서, 결함주입 스크립트를 실행한다. 결함주입 스크립트가 프로세스를 강제로 종료하면, 고장 검출 및 복구 절차에 의해 시스템이 장애 극복 되고, 고장복구 시간은 웹 스탠바이 모듈에 기록된다.



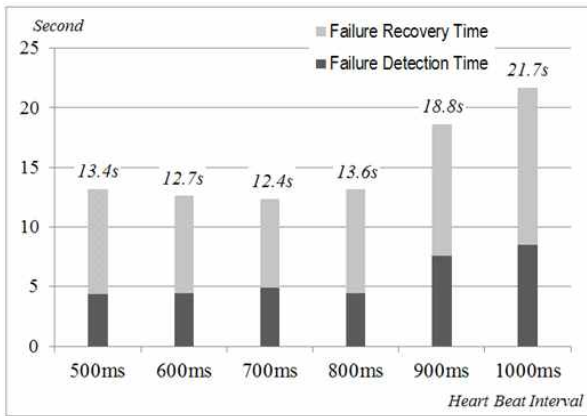


그림 11. 액티브 모듈의 전원고장에 의한 fail-over 측정  
**Fig. 11.** Fail-over measurement due to power failure of active module.

(Step 5): 결함주입 시험 결과 확인

실험 관리자는 액티브 모듈과, standby 모듈에 저장된 로그 파일을 분석하여, 고장 검출 및 복구 시간을 확인 할 수 있다.

4-3 결함주입 시험결과 분석

이중화 게이트웨이의 성능을 평가하기 위해서, 모의 교통관제 시스템에 임의적인 고장을 발생 시키고, 고장검출 시간과 고장복구 시간을 측정하였다. 임의적인 고장은 (1) 이중화 게이트웨이의 액티브 모듈의 전원 부 고장에 의한 하트비트 신호 단락과 (2) 액티브 모듈의 미션 프로세스 고장에 의한 게이트웨이 기능 중지로 구성한다. 액티브 모듈에 고장이 발생하면, 고장 유형에 따라 고장 검출과 복구 절차를 수행한다. 게이트웨이의 기능복구(failover) 성능은 고장 검출/복구 절차에 따라 차이가 발생한다. 특히, 하트비트 신호 주기는 웹 스탠바이 모듈의 고장검출 시간에 영향을 줄 수 있다. 본 연구에서는 각 고장 시나리오 별로 100회 시험을 실시하고, 그 결과를 통계적으로 정리하였다. 시험 결과는 (1) 각 고장 유형에 따른 기능복구(고장검출/고장복구) 성능과 (2) 하트비트 신호 주기에 따른 고장검출 시간을 확인하였다.

첫 번째로 이중화 게이트웨이의 액티브 모듈 전원 부 고장에 따른 기능복구 성능을 평가하였다. 시험 결과는 그림 11로 확인할 수 있다. 그림은 하트비트 신호 주기에 따른 고장검출 및 고장복구 시간을 설명한다. 하트비트 신호주기를 500ms로 설정하면, 고장검출에 소요되는 시간은 4.7초이며, 고장 복구에 걸리는 시간은 8.7초 측정되었다. 즉, 고장이 발생한 후, 게이트웨이가 정상적인 동작을 수행하는데 까지 소요되는 시간은 13.4초이다. 그림11을 보면, 이중화 게이트웨이의 기능복구 성능은 하트비트 신호주기에 영향을 받는 것을 확인할 수 있다. 하트비트 신호 주기가 500ms에서 1000ms로 증가하면, 기능복구 시간은 약 61% 증가하는 것을 확인할 수 있다. 하트비트 신호 주기가 길어지면 (1) 하트비트 프로세스의 하트비트 신호 대

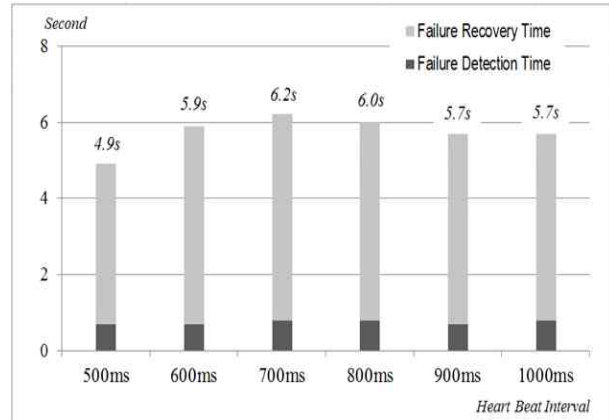


그림 12. 액티브 모듈의 미션 프로세스 고장에 의한 fail-over 측정  
**Fig. 12.** Fail-over measurement due to mission process failure of active module.

기 시간과, (2) 모듈의 고장 판정시간이 증가한다. 그러므로 대기시간의 증가는 고장검출과 고장복구 시간 증가의 원인이 된다.

두 번째로 이중화 게이트웨이의 액티브 모듈 미션 프로세스 고장에 따른 기능복구 성능을 측정하였다. 시험 결과는 그림 12로 확인할 수 있다. 그림은 하트비트 신호 주기에 따른 고장검출 및 고장복구 시간을 설명한다. 하트비트 신호주기를 500ms로 설정한 경우에, 고장검출에 소요되는 시간은 0.7초이며, 고장 복구에 걸리는 시간은 4.2초 측정되었다. 즉, 고장이 발생한 후, 게이트웨이가 정상적인 동작을 수행하는데 까지 소요되는 시간은 4.9초이다. 그림 12를 보면, 이중화 게이트웨이의 기능복구 성능은 하트비트 신호주기에 거의 영향을 받지 않고 있다. 미션 프로세서의 고장은 컨트롤 프로세서에 의해 검출되므로, 하트비트 신호 주기는 영향을 받지 않는다는 것을 확인할 수 있다.

이중화 게이트웨이의 액티브 모듈 전원 부와 미션 프로세스 고장에 대한 기능복구 성능시간은 표에 정리하였다. 고장유형에 따른 기능복구 성능은 하트비트 신호주기를 500ms를 기준으로 비교하였다. 표를 보면 전원 부 고장에 따른 기능복구 시간은 평균 8.7초인 반면, 미션 프로세스 고장에 따른 기능복구 시간은 평균 4초로 산출되었다. 전원 부 고장을 복구하기 위해 소요되는 시간은 미션 프로세스 고장에 비하여 45% 이상 증가하는 것을 확인할 수 있다. 그 이유는 두 고장의 고장 검출 방식에 차이가 있기 때문이다. 액티브 모듈의 미션 프로세서에서 발생하는 고장은 동 모듈의 컨트롤 프로세스에 의해 검출된다. 컨트롤 프로세서는 미션 프로세서의 공유메모리 값을 확인하여, 즉각적인 이상을 확인하고, 그 상태를 웹 스탠바이 모듈로 전달된다. 반면, 액티브 모듈의 전원 부 고장은 웹 스탠바이 모듈의 하트비트 프로세스에 의해 검출된다. 액티브 모듈에 전원이 단락 되면, 이중화 게이트웨이의 프로세스가 모두 종료된다. 그러므로 웹 스탠바이 모듈은 액티브 모듈의 고장을 바로 확인할



**표 2.** 이중화 게이트웨이의 fail-over 결과  
**Table 2.** Fail-over results of dual redundant gateway.

Failure type	Performance measurement	Min(sec)	Max(sec)	Avg.(sec)
Mission process failure	Failure detection time	2.261	4.775	3.267
	Failure recovery time	0.273	1.463	0.832
	Fail over time	2.534	6.238	4.099
Power supply failure	Failure detection time	4.924	6.424	5.921
	Failure recovery time	1.783	3.669	2.784
	Fail over time	6.707	10.093	8.705

수 없으며, 하트비트 신호의 대기 종료시간 정보를 사용하여 자체적으로 판정해야 한다.

### V. 결 론

본 연구에서는 교통신호제어(시스템)의 게이트웨이를 위한 저비용 이중화 시스템을 구현하고 장애 극복 성능을 측정하였다. 저비용으로 게이트웨이를 이중화하기 위해, 오픈소스 하드웨어인 라즈베리 파이 기반의 이중화 하드웨어를 구성하고, Linux HA를 사용하여 고장 검출 및 복구 기능을 개발하였다. 이중화 하드웨어의 고장복구 성능을 확인하기 위하여 하트비트 프로세스와 미션 프로세스의 고장 복구 시간에 대하여 측정하였다. 개발한 이중화 게이트웨이 시스템은 (1) 기존의 고비용 이중화 시스템에 비하여 H/W 부분의 비용 절감이 가능하고, (2) 경량의 하트비트 프로세스를 사용하여 시스템을 저 전력 및 소형으로 제작할 수 있으며, (3) 성능 측정 시험을 통하여 최적의 하트비트 메시지 교환 주기와 정확한 고장복구 성능에 대하여 확인 할 수 있었다.

교통신호, 자동차, 무인기 등 상용 안전필수 분야의 제품을 위한 저비용 고 신뢰성 플랫폼의 요구가 증가하고 있다. 본 연구 결과를 기반으로 다양한 분야의 고 신뢰성 제품을 상용화 할 수 있을 것으로 사료된다. 향후 지속적으로 연구를 수행하여 고장 복구시간을 줄이고 Real-Time으로 동작 가능한 플랫폼 개발이 필요하다.

### Acknowledgement

본 연구는 중소기업청 혁신기업기술개발사업의 연구비 지원(s2411346:적재중량 50kg 이상의 화물 운송이 가능한 대형 회전의 드론 핵심기술 개발)에 의해 수행되었습니다.

### References

- [1] G. Y. Go, "The status of development of smart signal control system and future of intersection," *TTA Journal*, no 160, pp.44-50, 2015
- [2] Intelligent Transport Society of Korea, "2013 Knowledge Sharing Program of Korea:Establishment of intelligent transport systems(ITS)," Government publications created by Ministry Of Strategy and Finance(MOSF) 11-1051000-000442-01, pp.83-95, 2013
- [3] J. O. Choi, J. W. Jeong, "The Internet of Things: The Future of Adoption," National Information Society Agency (NIA), pp. 16-18, 2014
- [4] J. P. Noh, J. H. Park, K. S. Son, and D. H. Kim "Reliability analysis of redundant architecture of dependable control system," *Journal of Institute of Control, Robotics and Systems*, vol 19. No 4, pp.328-333, 2013
- [5] [http://www.linux-ha.org/wiki/Main\\_Page](http://www.linux-ha.org/wiki/Main_Page)
- [6] J. Y. Jeong, S. J. Park, J. S. Lim, and C. H. Lee "Implementation and performance analysis of high-availability system for mission computer," *The Journal of the Korea Contents Association*, vol 8. No 8, pp.47-56, 2009
- [7] J. E. Kim, "A High Availability System Design for Enhancing Business Continuity Planning", A master's thesis from the Chung-ang University, 2005
- [8] [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [9] M. Collotta, L. L. Bello, and G. Pau "A novel approach for dynamic traffic lights management based on wireless sensor networks and multiple fuzzy logic controller," *Expert System with Applications*, vol 42. No 13, pp.5403-5788, 2015
- [10] S. S. Chavan, R. S. Deshpande, and J. G. Rana, "Design of Intelligent Traffic Light Controller using Embedded System," *Emerging Trends in Engineering and Technology(ICETET)*, pp.1086-1091, 2009



**이 동 우 (Dong-Woo Lee)**

2014년 8월: 한국항공대학교 항공전자공학과 (공학박사)  
2014년 ~ 현재: 한국대학교 항공전자연구소 연구원  
※ 관심분야: 고신뢰성 시스템, 영상처리, 항공전자, 안전설계 및 검증



**나 종 화 (Jong-Whoa Na)**

1995년 2월: 아리조나대학교(미) 컴퓨터공학과 (공학박사)  
2005년 ~ 현재: 한국항공대학교 항공전자 및 정보통신공학부 교수  
※ 관심분야: 컴퓨터 시스템, 영상처리, 고신뢰성 시스템



**김 남 선 (Nam-Sun Kim)**

2005년 2월: 아주대학교 박사(교통공학 전공)  
1999년 2월: 아주대학교 석사(교통공학 전공)  
2015년 6월 ~ 현재: 경찰청 경찰대학 치안정책연구소 책임연구원  
※ 관심분야: 지능형교통체계(ITS), C-ITS, 자율주행, 드론