

네트워크 기반 모듈라 로봇의 원격 제어

염동주¹, 이보희^{2*}

¹세명대학교 전기전자공학과 학생, ²세명대학교 전기공학과 교수

Remote Control of Network-Based Modular Robot

Dong-Joo Yeom¹, Bo-Hee Lee^{2*}

¹Student, Department of Electrical and Electronic Engineering, Semyung University

²Professor, Department of Electrical Engineering, Semyung University

요 약 동작을 기억하는 모듈라 로봇은 손으로 직접 표현하기 때문에 창의적 구조물을 쉽게 만들고, 동작시킬 수 있다. 하지만 사용자에게 의하여 만들어진 동작을 저장할 충분한 저장 공간이 모듈 내에 없어서 만들어진 동작을 재사용이 불가능하며 모듈라 로봇이 다시 동작을 기억할 시에 다른 동작으로 바뀌게 된다. 또한, 다수의 모듈라 로봇을 동시에 동작시킬 수 있는 주 제어가 없어서 직접적으로 사용자가 모듈라 로봇에 입력해야하는 단점이 있다. 이러한 단점을 극복하고자, 유선 및 무선 네트워크를 이용하고 웹 서버 및 컴퍼넌트 기반 소프트웨어를 설계하여, 주변의 스마트 기기에서 동작시킬 수 있는 원격제어기를 제안하였다. 그리고 제안된 제어기의 하드웨어 개념 및 소프트웨어의 연결 관계를 자세히 제시 하였다. 제안된 방식은 모듈라 로봇에 연결하여 다양한 형태의 구조물을 만들어 동작시키고 저장한 후 다시 재생 동작을 수행하여 동작의 재현성을 보였으며 기존의 저장된 동작을 효과적으로 재생함으로써 유용성을 확인하였다. 아울러 다운로드한 궤적 데이터를 도해적으로 표현하고 실제 동작된 궤적과의 차이를 분석하여 신뢰성을 확인하였다. 향후에는 원격제어기에 저장된 궤적을 인공지능 기법을 이용하여 표준화시켜 모듈라 로봇의 동작을 손쉽게 구현 시킬 예정이다.

주제어 : 모듈라 로봇, 창의적 구조물, 원격 제어, 웹 서버 및 컴퍼넌트 방식 소프트웨어, 동작 궤적

Abstract A modular robot that memorizes motion can be easily created and operated because it expresses by hand. However, since there is not enough storage space in the module to store the user-created operation, it is impossible to reuse the created operation, and when the modular robot again memorizes the operation, it changes to another operation. There is no main controller capable of operating a plurality of modular robots at the same time, and thus there is a disadvantage that the user must input directly to the modular robot. To overcome these disadvantages, a remote controller has been proposed that can be operated in the surrounding smart devices by designing web server and component based software using wired and wireless network. In the proposed method, various types of structures are created by connecting to a modular robot, and the reconstructed operation is performed again after storing, and the usefulness is confirmed by regenerating the stored operation effectively. In addition, the reliability of the downloaded trajectory data is verified by analyzing the difference between the trajectory data and the actual trajectory. In the future, the trajectory stored in the remote controller will be standardized using the artificial intelligence technique, so that the operation of the modular robot will be easily implemented.

Key Words : Modular robot, Creative structure, Remote controller, Web server and component-based software, Motion trajectory.

*This paper was supported by the support of research projects in the Semyung University in 2016.

*Corresponding Author : Bo-Hee Lee(bhlee420@nate.com)

Received August 27, 2018

Revised September 17, 2018

Accepted October 20, 2018

Published October 31, 2018

1. 서론

우리 주변에는 다양한 형태의 많은 로봇이 있지만, 일반적으로 형태가 정해져 있는 로봇의 경우에는 특정 작업만을 수행이 가능하다. 하지만 독립된 모듈라 로봇은 서로 결합하여 여러 가지 형태를 만들어내면서 산업이나 구조, 탐사, 교육 등 여러 분야에서 활용될 수 있다[1]. 또한, 그 중 요즘에는 아이들의 교육을 위한 일환으로 모듈라 로봇이 각광 받고 있으며, 이에 따른 창의력이나 코딩 교육에 대한 실험도 진행되고 있다[2-4]. 동작을 기억하여 실행하는 동작 기억형 모듈라 로봇의 경우에는 프로그래머가 아닌 손으로 직접 재현하는 코딩이기에 손쉽게 구조물을 표현할 수 있다[5]. 하지만, 이러한 모듈라 로봇을 손으로 직접 만드는 행동은 복잡한 동작 생성 시 어려움이 있다. 실제로 걷기운동과 같은 어려운 운동을 만드는 데에 있어서 동작 구현이 힘들기도 하며, 동작이 완성되었다 하더라도 저장할 기록장소가 없기 때문에 동작을 재현하기가 어렵다. 논문에서 제안되는 모듈라 로봇은 스위치의 조작에 따라 동작 기억-동작 재생-정지 순으로 이루어지기 때문에 동작 기억이 최우선적으로 되므로 사용자가 인식하지 않으면 동작된 궤적을 잃어버릴 수 있는 단점이 있다. 이에 따라서 본 논문은 모듈라 로봇의 기본적인 동작 기억, 동작 재생, 정지 명령을 PC, 스마트폰을 웹브라우저와 연계에 따른 제어를 하여 궤적을 기기에 업로드하고 또 필요할 경우 다운로드 할 수 있는 원격 제어 시스템을 제안한다. 이를 통하여 사용자가 편리하게 원격제어를 통하여 모듈라 로봇을 보다 쉽게 제어할 수 있으며, 사용자 인터페이스를 이용하여 제작을 하고 싶은 구조물의 궤적을 미리 저장 및 관리하여, 쉽게 구조물의 궤적을 분별할 수 있도록 제안하였다.

2. 모듈라 로봇의 구조

2.1 기구부 구조

모듈라 로봇의 구조물은 3D프린터를 이용하여 플라스틱 기반으로 제작하였으며 모듈간의 접합면과 접합 구조를 Fig. 1과 같이 설계하였다.

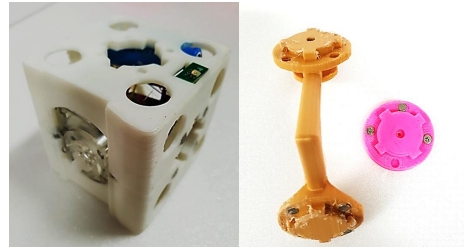


Fig. 1. Docking face and docking assembly

모듈라 로봇은 모터의 결합을 위해 도킹부분 제외하면 한 면이 70밀리미터 부피를 가진 큐빅 모양으로 만들어진다. 그리고 각 면은 다른 모듈의 면과 결합되기 위한 액세스리틀을 사용하였다. Fig. 2는 모듈 결합 모터 축과 결합구조의 체결부에 대한 상세내역을 보여주고 있다.

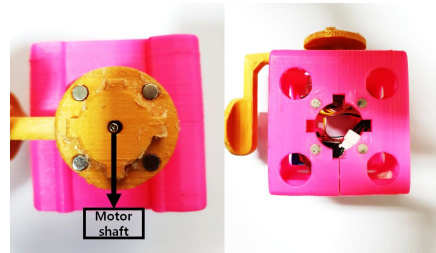


Fig. 2. Detailed joint assembly

이러한 구조를 이용하여 모듈간의 결합은 Fig. 3과 같은 형태로 수행된다. 한쪽 면은 결합 구조가 부착하여 움직임을 생성하는 부분이고 반대쪽 면은 결합구조가 결합될 수 있는 홈을 만들어 결합이 이루어지게 설계 되었다.



Fig. 3. Docking process of two modular blocks

그리고 모듈의 견고한 결합을 위하여 결합부는 십자형으로 설계하였으며 기본적으로 네오디움 자석을 통한 자력으로 보완하였다. 여기서 운동하는 회전부를 N극으로 설정하고 모듈라 로봇의 고정부분은 S극으로 설정하

였다. 이러한 기본 결합 방식을 이용하여 다양한 형태의 구조물을 생성할 수 있다. 그림 4와 같이 각 모듈들이 결합하여 탱크 움직임과 탐사 로봇 및 자전거 형태 및 애벌레 움직임과 같은 복잡한 구조물로 응용할 수 있다.



Fig. 4. Combination examples of modular robot

2.2 단위 모듈 제어기 구조

전체적인 모듈라 로봇 제어기는 Fig. 5와 같이 설계되며 모듈라 로봇이 움직이고 사용자 인터페이스를 위한 버튼스위치 및 모터를 구동하는 제어부를 통제하기 위한 하드웨어를 배치하였다.

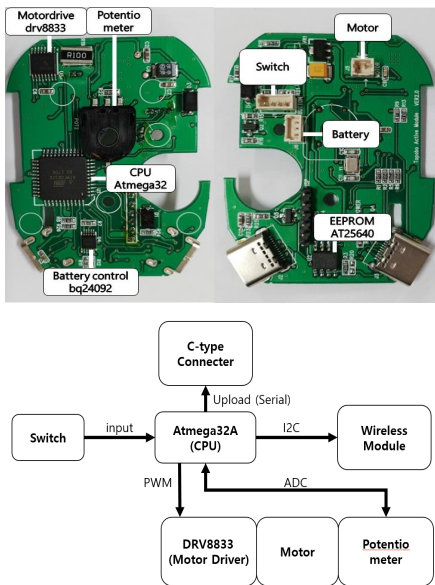


Fig. 5. Control board assembly and block diagram

여기서 CPU는 Atmega32A를 사용했으며 동작을 제어하기 위한 버튼스위치 1개와 LED를 2개 장착하여 만들었으며, 구동을 위한 모터의 속도를 제어하는 모터드라이브가 설계되었다. 동작의 저장 및 동작의 재생을 위해 위치정보를 저장하는 EEPROM도 사용하였다. 모듈라 로봇 간에 원격 동기화를 위하여 NRF24L01P 무선 모듈을 설계하였고 3.7V, 800mAh의 용량의 배터리의 충방전 회로를 설계 및 배치하였다. 그리고 서로 다른 모듈의 결합 시 통신위한 유, 무선 통신 회로를 설계 하였다 [6]. 무선통신은 모듈라 로봇과 무선모듈 간의 I2C 통신 방식을 채택하여 모듈라 로봇의 버튼 값을 읽어 무선모듈로 보낸다. 그 후 무선모듈의 CPU와 NRF24L01P간에 SPI 통신을 통해서 모듈간의 통신을 하였다[7]. 모듈의 동작은 모듈에 결합된 버튼을 이용하여 한번 터치 시 동작을 기록하고 다시 한 번 터치하면 동작을 재생하는 단순한 구조로 되어 있으며 결합된 모듈은 같은 방식으로 동기에 맞추어 동작한다.

3. 원격 제어기 설계

3.1 하드웨어 구조

단위 모듈의 개별적인 동작은 버튼으로 통제되어 저장 및 재생이 되지만 또 다른 동작을 만들면 기록이 지워지고 만다. 이를 극복하기 위해 모듈라 로봇을 원격 동작 제어기를 구성한다. 이러한 제어기는 기본적으로 리눅스와 같은 OS를 담고 있어야 하며 NanoPi Neo Air를 기반으로 그림 6과 같이 원격제어기를 제안하였다. 기본적인 구성은 리눅스 모듈을 기본으로 하여 LED와 버튼을 이용한 사용자 인터페이스 환경을 구축하였고 외부에서 와이파이를 이용하게 하기 위한 와이파이 통신 인터페이스를 설계 하였다[8-10]. 아울러 리모콘과의 접속 및 입출력 하드웨어 확장을 위한 보조 제어장치인 Arduino mini와의 결합한 형태로 구현되었다. 원격 제어기는 만들어진 구조에 연결하여 개별 모듈에 저장된 동작을 업로드할 수 있게 설계 하였다. 또한 동작의 재생을 하기 위해 무선 통신을 이용하여 동작 데이터를 다운로드 할 수 있으며 다운로드된 데이터를 재생하는 명령도 수행할 수 있다. 원격 제어기는 독립동작을 수행할 수 있게 하였으며, 리튬 이온 전지를 장착하였으며 충·방전되는 회로를 첨가하였다.

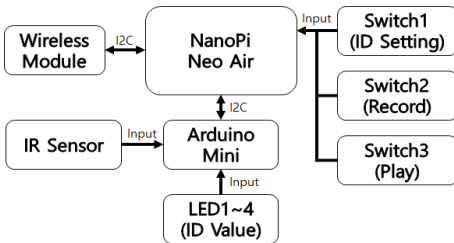
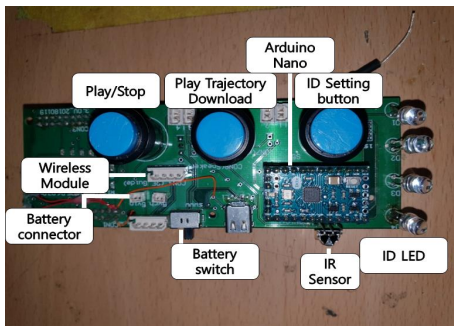


Fig. 6. Control board assembly & block diagram of remote controller

3.2 원격 제어 소프트웨어

원격제어기는 기본적으로 원격에서 다양한 스마트 기기에서 통제해야 하므로 스마트 기기와 원격 제어기를 연결시킬 수 있는 구조가 설계 되어야 한다. Fig. 7은 이러한 소프트웨어를 나타낸 그림이며 사용자는 다양한 사용자 환경에서 원격 제어기를 통제 하게 된다.

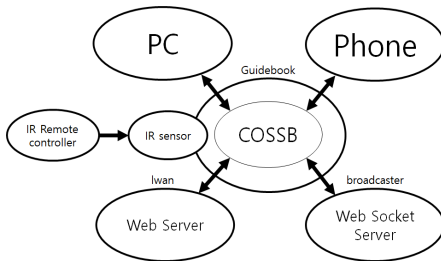


Fig. 7. Relation diagram of remote controller

이때 원격 제어기내의 다양한 하드웨어를 통제하고 효율적인 제어를 하기 위해 사용자가 바로 프로그램을 수행하지 않고 시스템 차원에서 하드웨어를 통제하는 시스템 소프트웨어를 설계하고 메시지를 이용하여 통제하는 방식이 필요하다. 사용자는 이러한 소프트웨어 기반으로 웹 서버 및 웹 소켓 서버를 통하여 외부매체와 통신할 수 있기 때문에 PC 또는, 스마트폰으로 웹 브라우저와 연계하여 사용할 수 있다[11-13]. 이 경우, 센서, 제어

모듈로부터의 데이터 수집, 웹 서버와 같은 여러 시스템 소프트웨어의 재사용과 같은 다양한 기능을 통합해야 하는 시스템에 적합한 소프트웨어 프레임 워크가 필요하다. 이 때 필요에 따라 새로운 보다 복잡한 기능이 추가되면 구조를 재구성하거나 레거시를 분해하지 않고 쌓아두기가 어렵다. 따라서, 분산 환경에서 여러 서비스 간 상호 운영을 목표로 하는 COSSB (Component-based Open & Simple Service Broker)라는 구성 요소 기반의 소프트웨어 프레임 워크를 그림 8과 같이 설계하였다.

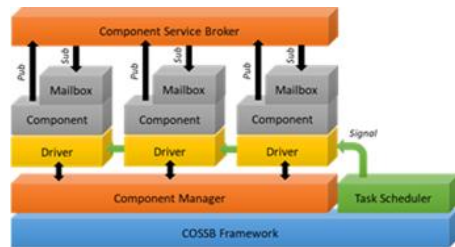


Fig. 8. COSSB structure

COSSB를 기반으로 한 소프트웨어 아키텍처는 여러 서비스 블록으로 구성되며 Linux 기반 임베디드 시스템에서 실행되며 구성 요소가 명령과 응답을 동기화해야 할 경우 publish / subscribe 방식을 사용하여 시스템은 모든 구성 요소가 독립적으로 실행되고 구성 요소 드라이버 및 관리자에 의해 제어되므로 특정 서비스 구성 요소의 오류에 유연하게 대응할 수 있다. 전체 동작은 프로필이 로드되고 각 구성 요소의 서비스는 작업 스케줄러에 의해 수행된다. 각 서비스 구성 요소에는 서비스 브로커를 통해 배달되는 비동기식으로 수신된 메시지를 보관할 수 있는 메일박스를 사용한다[14]. 이 소프트웨어 프레임 워크를 사용하는 궁극적인 목적은 사용자가 필요할 때 시스템을 재구성하거나 재구성하지 않고 간단한 서비스 구성 요소를 추가하여 기능을 확장할 수 있다. 액티브 모듈을 처리하기 위해 GPIO 구성 요소인 4 가지 기본 구성 요소, 기능 데이터에서 GPIO 출력으로 인코딩하는 응용 프로그램 구성 요소, 클라우드 기반 REST API 및 시스템 GPIO 입출력 제어 구성 요소와 같은 모션 재생 및 기록 구성 요소가 등록된다[15-17].

3.3 사용자 인터페이스 설계

모듈라 로봇의 동작을 원격에서 통제하는 수단으로 웹브라우저가 이용된다. 원격에서 통제하기 위한 프로

그랩은 원격에서 접속하여 궤적을 저장하고 필요에 따라 접속된 모듈라 로봇에 저장된 궤적을 송부하여 동작하게 하는 방법을 되풀이 한다. Fig. 9는 기본적인 화면 인터페이스로, 테스트를 위한 테스트창과 사용자가 그림으로 보기 쉽도록 하는 유저 인터페이스를 설계하였다.

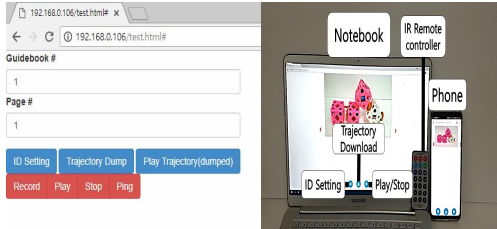


Fig. 9. GUI of remote controller

사용자 인터페이스는 Fig. 9와 같이 웹 기반으로 설계 되었지만(좌측 그림) 사용자의 편의성을 위한 Text 기반으로 동작할 수 있게 하는 방식과 그래픽을 이용하는 방식으로(우측 그림) 나눠 설계 되었다. 웹 화면은 모듈라 로봇의 동작 기억, 동작 재생, 정지 기능을 이용하여 모듈라 로봇의 기본적인 제어를 수행할 수 있게 설계하였으며, 동작 저장(trajectory dump)를 통하여 궤적을 업로드 하거나, 동작 재생(trajectory play)를 실행하여, 궤적을 모듈라 로봇에 다운로드하게 된다. 그리고 유저인터페이스의 상단에는 저장된 구조물이 표시 될 수 있게 만들어 어떠한 구조물이 동작하는지를 알 수 있게 하였으며 좌우측 화살표 버튼을 통하여 다른 모듈라 로봇 예제 샘플로 바꿀 수 있다. 궤적을 옮길 경우 내부의 메모리에 궤적이 저장되며, 이 궤적을 이용하여 모듈라 로봇에 궤적을 넣을 수 있다. 또한 PC에서는 그림 9와 같이 작동하며, 스마트폰에서 또한 동작하며, 적외선 리모컨의 상단 3개 버튼을 이용하여 리모컨 상단의 좌측부터 동작 기억, 동작 재생, 정지 명령을 내릴 수 있게 설계하였다.

4. 동작실험

원격으로 동작함을 검증하기 위해 소프트웨어 동작을 실험하였다. 임베디드 기반 제어기의 동작은 웹브라우저를 이용하여 각각 기록, 재생, 정지 명령과 궤적 저장 명령을 설정하면 원격제어기에서 동작하고 있는 COSSB가 명령을 수신하여 해독하여 각 모듈라 로봇에게 전송하게 된다.

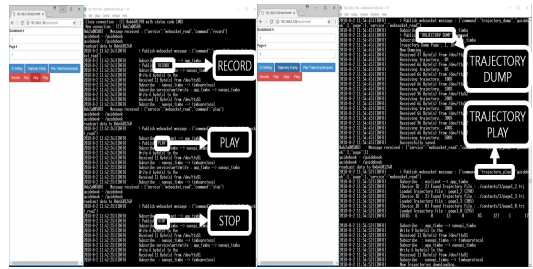


Fig. 10. COSSB operation history

Fig. 10은 소프트웨어 알고리즘의 이력 메시지로 명령에 따라 잘 수행됨을 알 수 있다. 아울러 만들어진 동작이 원격제어기에서 제대로 저장되며 재생이 됨을 알 수 있었다. 이때의 구조물의 동작은 Fig. 11과 같이 애벌레 모양이며 전진 운동을 표현하고 있다.

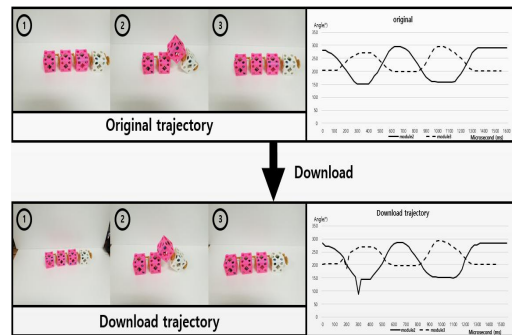


Fig. 11. Larva operation and motion trajectory

애벌레 모양을 동작 시키면서 궤적을 업로드한 결과를 그래프로 표현 하였다. 동작이 이뤄지는 두 개의 모듈의 궤적 그래프를 얻을 수 있었다. 궤적 데이터의 경우에는 애벌레 형으로 만들 경우, 모듈라 로봇이 4개를 사용되지만 실질적인 궤적 이동은 2개의 관절만 사용하므로 도식화하기 편하기 위해서 2개의 모듈만 궤적데이터를 예를 들었다. 이러한 동작을 이용하여 이번에는 업로드한 동작을 원격에서 수행시켜 보았다. 그림 11과 같이 궤적데이터가 다운로드 되는 모습을 볼 수 있었다. 그리고 이때의 궤적을 다시 읽어 비교해본 결과, 동작은 저장된 동작과 같이 제대로 수행됨을 알 수 있었다. 하지만 궤적에서 중간부분에 노이즈가 발생한 현상을 볼 수 있었는데 이 부분은 모듈라 로봇에 장착된 무선 모듈과 원격제어기의 거리가 떨어져 있어 나타나는 현상으로 기인한다. 하지만 이러한 노이즈를 나타내는 구간이 짧기 때문에 모듈라 로봇의 동작에는 영향을 크게 주지 않았고,

이를 보완해주기 위하여 무선모듈 및 원격제어기에 소형 외장 안테나를 추가하여 송·수신 감도를 올려서 실험하여 노이즈에 둔감한 동작을 할 수 있었다.

5. 결론

모듈라 로봇의 동작의 개별적인 동작의 저장 및 재생을 확장시켜 다수의 모듈라 로봇의 동작을 저장하고 재생시키며 재연성을 만들기 위한 원격 제어를 설계하였다. 이를 위한 모듈라 로봇의 기본 구조 및 결합 알고리즘을 제안하였으며, 동작 데이터를 원격에서 통제하기 위한 리눅스 기반 전송 구조를 제안하였다. 또한 원격에서 웹을 이용하여 이를 통제하는 웹 기반 사용자 인터페이스를 설계 구현하였다. 그리고 이를 실제 애벌레 동작에 적용하여 제안한 방식이 효과적으로 동작함을 알 수 있었다. 향후 인공지능 기법을 사용하여 원격제어기에서 모듈라 로봇의 동작을 표준화시켜 사용자가 좀 더 간편하게 하기 위한 동작을 재생시킬 수 있는 방식에 대한 연구가 요구된다.

REFERENCES

- [1] K. Gilpin & D. Rus. (2010). Modular Robot Systems, *IEEE Robotics & Automation Magazine*, 7, 38-55.
DOI : 10.1109/MRA.2010.937859
- [2] V. T. Le & T. D. Ngo. (2017). moreBots: System Development and Integration of An Educational and Entertainment Modular Robot. *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 74-80.
DOI : 10.1109/IRIS.2017.8250101
- [3] K. Mitsuhashi & Y. Ohyama. (2017). Suggestion and Verification of the Modular Robot Education. *11th Asian Control Conference (ASCC)*, 1379-1383.
DOI : 10.1109/ASCC.2017.8287373
- [4] S. M. Kim & K. S. You (2015). The Effect of Robot-Based STEAM Class on the Korean Learning of Multicultural School Children - Focusing on After School Learning of Elementary School -. *Journal of Digital Convergence*, 13(8), 1-8.
DOI : 10.14400/jdc.2015.13.8.1
- [5] M. M. Santin, S. S. C. Botelho, G. P. Dimuro & C. L. L. Rodrigues. (2011). Hand Skill Programming: Using Fuzzy Sets to Program Topobo Kinectis Memory Devices. *Workshop-School on Theoretical Computer Science*, 67-73.
DOI : 10.1109/WEIT.2011.30
- [6] M. Huasong, L. Zhenglin, W. Hongxing & W. Tianmiao. (2010). Control of a self-assembly modular robot system over a wireless ZigBee network. *8th World Congress on Intelligent Control and Automation*, 1057-1062.
DOI : 10.1109/WCICA.2010.5554865
- [7] J. T. Kim, J. Y. Park, B. H. Lee, M. Q.-H. Meng, Chao Hu & Jie Sheng. (2017). Implementation of Functional Blocks of Modular Toy for Creative Education. *Journal of Convergence for Information Technology*, 7(5), 95-102.
DOI : 10.22156/CS4SMB.2017.7.5.095
- [8] P. X. Liu, M. H. Meng, C. Hu & J. Sheng. (2003) A modular structure for Internet mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 2, 1111 - 1116.
DOI: 10.1109/IROS.2003.1248793
- [9] C. Chengyi, Z. Guannan & J. Minglu. (2010). A ZigBee based embedded remote control system. *2nd International Conference on Signal Processing Systems*, 3, 373-376.
DOI: 10.1109/ICSPS.2010.5555814
- [10] B. C. Kim. (2015). A Internet of Things(IoT) based exploration robot design for remote control and monitoring. *Journal of Digital Convergence*, 13(1), 185-190.
DOI: 10.14400/jdc.2015.13.1.185
- [11] B. G. Jun. (2015). Implementation of the Centralized Control System for Swarm Robots using Multi-Threading method. *Journal of Digital Convergence*, 12(6), 349-354.
DOI: 10.14400/jdc.2014.12.6.349
- [12] G. Mohanarajah, D. Hunziker, R. D'Andrea & M. Waibel. (2014). Rapyuta: A Cloud Robotics Platform. *IEEE Transactions on Automation Science and Engineering*, 15, 481-493.
DOI: 10.1109/TASE.2014.2329556
- [13] B. C. Chung & W. S. Na. (2016). A Study on the Smart Fire Detection System using the Wireless Communication. *Journal of Convergence for Information Technology*, 6(3), 37-41.
DOI : 10.22156/cs4smb.2016.6.3.037
- [14] Y. T. Poornima, S. R. Kalathuru & P. G. Poddar. (2017). Mailbox based inter-processor communication in SoC. *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication*

Technology (RTEICT), 1033 - 1037.

DOI : 10.1109/RTEICT.2017.8256756

- [15] F. Haupt, F. Leymann, A. Scherer & K. V. Haupt. (2017). A Framework for the Structural Analysis of REST APIs. *IEEE International Conference on Software Architecture (ICSA)*, 53-58.
DOI : 10.1109/ICSA.2017.40
- [16] L. Turnbull & B. Samanta. (2013). Cloud robotics: Formation control of a multi robot system utilizing cloud infrastructure. *Proceedings of IEEE Southeastcon*, 1-4.
DOI : 10.1109/SECON.2013.6567422
- [17] S. H. Kim & G. J. Kim. (2014). Intelligent Safe Network Technology for the Smart Working Environments based on Cloud. *Journal of Digital Convergence*, 12(12), 345-350.
DOI : 10.14400/jdc.2014.12.12.345

염 동 주(Dong-Joo Yeom) [학생회원]



- 2011년 2월 : 세명대학교 전기공학
학과 학사
- 2018년 2월 ~ 현재 : 세명대학교
전기전자공학과 석사
- 관심분야 : 로봇, 임베디드, 인공
지능, 드론

· E-Mail : ydj1023@nate.com

이 보 희(Lee-Bo Hee) [정회원]



- 1985년 2월 : 인하대학교 전자공
학과 공학사
- 1996년 8월 : 인하대학교 자동화
공학과(제어 및 로봇) 공학박사
- 1997년 ~ 현재 : 세명대학교 전기
공학과 교수

· 관심분야 : 로봇, 임베디드, 인공지능, 드론

· E-Mail : bhlee420@nate.com