

# 다중 연결이 가능한 터미널 에뮬레이터 개발

최홍석, 김윤수, 김진수, 주영관, 전중남\*  
 충북대학교 소프트웨어학과

## Development of Multi-connection Terminal Emulator

Hong-Soek Choi, Youn-Su Kim, Jin-Su Kim, Young-Kwan Ju, Joong-Nam Jeon\*  
 Department of Computer Science, Chungbuk University

**요약** 최근 IoT의 발전으로 다양한 통신방식을 지원하는 장치들을 서로 연결하여 데이터를 송·수신하는 사례가 증가하고 있다. 본 연구에서는 직렬통신, TCP, UDP, ZigBee, Bluetooth 등 여러 가지 통신 방식을 지원하는 멀티 터미널 에뮬레이터를 제안한다. 이 시스템은 각 통신 방식 별로 여러 개의 채널 연결을 지원하고, 데이터 전송을 예약할 수 있고, 통신 데이터 통계를 수집하여 그래프로 표현하고, 로그 기록을 저장하는 기능을 제공한다. 이 시스템을 사용함으로써 다중 장치간 통신 연결을 테스트할 수 있게 되어, IoT 시스템 개발에 편의성을 제공할 것으로 기대한다.

**키워드** : IoT, 다중 연결, 터미널 에뮬레이터, 직렬 통신, 개발도구

**Abstract** Recently, with the development of IoT, there have been increasing cases of transmitting and receiving data by connecting more than one devices supporting various communication methods. In this paper, we propose a multi-terminal emulator supporting various communication methods. The system supports multiple channel connections for each communication method, schedules data transmission, collects communication statistics, represents them into graphs, and stores log records. By using this system, it is possible to test the communication connection between multiple devices, and it is expected to provide convenience for IoT system development.

**Key Words** : IoT, multi-connection, terminal emulator, Serial Communication, Development Tools

### 1. 서론

최근 제4차 산업혁명의 시대가 도래 하면서, 그에 따른 정보통신기술의 발전과 함께 모든 사물이 통신으로 연결되는 IoT(Internet of Things)의 개념이 보편화 되었다. IoT의 발전으로 다양한 통신 방식을 갖는 다수의 장치들과의 연결 필요성이 증가함에 따라, 기존의 일대일 연결만을 지원하던 터미널 에뮬레이터는 이를 충분히 지원하지 못하는 한계점이 나타났다[1].

기존의 터미널 에뮬레이터는 직렬 통신 또는 TCP/UDP

통신 중 한 가지 방식의 연결을 지원한다. 본 논문에서는 이런 단점을 개선하기 위하여 IoT 환경에서 여러 가지 통신 방식을 갖는 장치들을 연결하여 즉각 데이터 통신을 시험할 수 있는 다중 연결 터미널 에뮬레이터를 연구한다. 본 연구에서 제안하는 에뮬레이터는 직렬 통신, TCP/UDP, Zigbee, Bluetooth 통신 방식을 지원하고, 각 통신 방식에 대하여 여러 개의 채널을 설정하여 선택적으로 데이터를 송·수신할 수 있고, 효율적인 테스트를 위하여 주기적 데이터 전송 및 시간 예약 기능을 지원하고, 채널별 통신량을 그래프 형태로 제공하고, 채널별 통

신 이력을 로그 파일로 저장하는 기능을 제공한다. 이런 기능을 갖춘 다중 연결 터미널 에뮬레이터는 IoT기반 서비스의 개발 편의성을 제공할 것으로 기대된다.

이 논문의 2장에서는 관련 연구를 기술하고, 3장에서는 다중 연결 터미널 에뮬레이터의 구조 및 기능을 자세히 설명하고, 실제로 구현한 다중 연결 터미널 에뮬레이터를 설명하며, 4장에서는 구현된 시스템에 대한 실험 결과를 제시한다. 마지막으로 결론에서는 본 논문에서 제안하는 다중 연결 터미널 에뮬레이터의 효용성 및 한계에 대하여 기술하였다.

## 2. 관련연구

이 장에서는 터미널 에뮬레이터 개발에 필요한 관련 연구를 살펴본다.

사물인터넷 기술은 인터넷 기술 표준화하는 IETF (Internet Engineering Task Force)에서는 표준 통신 프로토콜을 기반으로 유일하게 식별되고 상호 연결된 객체들간의 네트워크로 정의한다[2].

사물인터넷 (IoT) 네트워크는 일반적으로 저비용, 저전력, 다기능의 센서 노드들로 구성되고, 이러한 센서들은 유무선 전송장치, 데이터 처리를 실현하는 마이크로 프로세서로 구성된다[3].

사물인터넷 (IoT) 구현을 위한 4가지 기술 요소인 스마트디바이스, 진화된 네트워크, 클라우드 컴퓨팅, 빅데이터 분석이 최근에 활용되고 있다[4].

다양한 IoT 장치들을 모니터링하기 위해, 서버보다 지연 시간이 짧은 Smart Fog와 같은 구성을 사용하여 실시간 인터랙션을 구현하고 있다[5]. 그리고 IoT 장치들의 다중 연결 서비스를 제공하기 위하여 Open Hardware 기반의 게이트웨이를 활용한다[6].

IoT 네트워크 구성은 근거리 통신, WiFi, LTE 등이 최근에 많이 사용되고 있다. 서버나 중간 규모의 디바이스는 인터넷 연결을 기반으로 정보를 주고받는 TCP/IP가 사용되고, 이외의 소규모 디바이스는 블루투스, Zigbee, RFID 등의 직렬통신을 센서 노드연결에 활용하고 있다[7].

사물인터넷의 정보들은 각 노드간의 호환성을 위해 정해진 프로토콜을 이용해 정보를 서버로 전송하고, 서버는 수신한 정보를 가공해 필요한 노드로 정보를 전송하는 구조를 갖고 있다[8].

IoT 통신 장치 개발에서 주로 쓰이는 상용 프로그램으로는 대표적으로 Putty, TeraTerm, Xshell 등이 있다[9].

시리얼 통신은 거의 모든 PC에서 표준으로 사용되는 디바이스 통신 프로토콜이며, 정보의 바이트를 한 번에 한 비트씩 순차적으로 송수신한다[10].

TCP/IP는 컴퓨터와 컴퓨터간의 LAN 혹은 WAN에서 원활한 통신을 가능하도록 하기 위한 통신규약이다[11].

## 3. 제안 프로그램

이 장에서는 본 논문에서 제안하는 다중 연결이 가능한 터미널 에뮬레이터는 시스템 구성과 기능에 대해 기술한다.

### 3.1 시스템 설계

이 절에서는 다중 터미널 에뮬레이터의 전체 시스템 구성도에 대해서 기술한다. Fig. 1은 다중 터미널 에뮬레이터의 전체 시스템 구성도를 나타낸 것이다.

통신 연결 선택부는 통신 프로토콜인 TCP/UDP와 직렬 통신 방식 중에서 선택하여 설정한다. 그리고 단일 통신뿐만 아니라 여러 대의 통신 장치를 연결을 설정할 수 있다.

송·수신 기능부는 각 연결 장치들에 명령전송이나 장치가 전송한 결과를 화면에 표시하여 장치에 송·수신한 내역을 확인할 수 있는 기능이다. 그리고 연결된 장치를 선택적으로 송·수신하도록 옵션을 제공하여 편의성을 갖추었다.

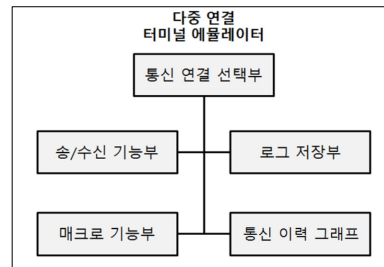


Fig. 1. Structure of System

매크로 기능부는 연결된 장치와의 송·수신 주기를 설정하여 정해진 데이터를 설정한 주기마다 재전송하는 기능이다.

로그 저장부는 프로그램의 송·수신 기록을 저장하여 로그로 보관한다.

통신이력 그래프는 통신 장치가 연결된 이후로 에뮬레이터와 통신 장치간의 송·수신 기록을 그래프로 표현해주는 기능이다.

### 3.2 프로그램 구현

이 절에서는 다중터미널 에뮬레이터를 구현한 프로그램에 대해서 기술한다.

Fig. 2는 다중 연결 터미널 에뮬레이터를 구현한 프로그램을 캡처한 것이다. C#에서 제공하는 SerialPort 클래스를 이용해 직렬 통신을 구현하였고, TCP/UDP의 구현은 소켓 클래스를 사용하여 구현하였다.

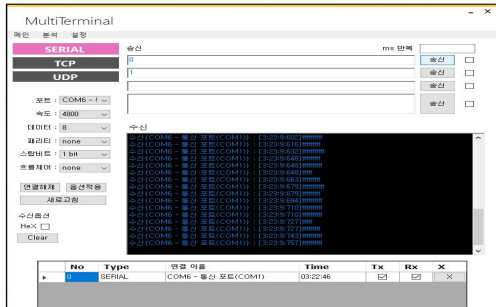


Fig. 2. Implementation of Multi Terminal

#### 3.2.1 연결 설정 기능

Fig. 3은 통신 프로토콜을 선택하는 옵션을 나타낸 화면이다. 선택 가능한 프로토콜 최근에 IoT에 가장 많이 활용되는 직렬 통신, TCP, UDP를 선택해 구현하였다. 직렬 통신 프로토콜은 포트, 속도, 데이터 크기, 패리티, 정지비트, 흐름제어 옵션을 사용자가 설정하여 연결하도록 구성하였다. 직렬 통신의 포트에 연결 커넥터인 동글을 활용하면 블루투스 외의 다양한 직렬 통신 지원 IoT 모듈들을 활용할 수 있다. 직렬 통신 장치에는 NFC, Zigbee, 거리센서, 빛 센서 등이 있다.

TCP와 UDP의 연결은 포트번호와 아이피 주소를 입력하고, 이 에뮬레이터를 서버역할 또는 클라이언트 역할 여부를 체크하여 설정을 완료한다.

이 에뮬레이터는 시험하기 원하는 직렬 통신 또는 TCP/UDP 장치를 동시에 연결에 추가할 수 있다. 통신의 종류가 달라도 에뮬레이터를 통해 서버 클라이언트 역할을 부여해 다양한 송·수신 시험이 가능하다.

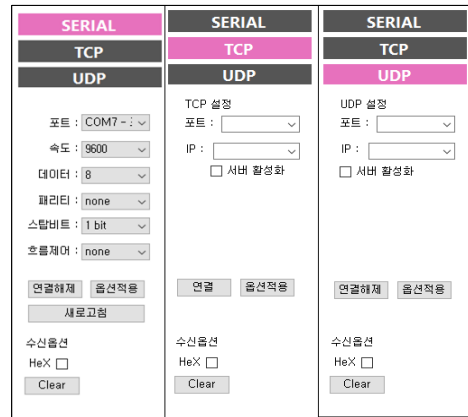


Fig. 3. Connection setting function

#### 3.2.2 데이터 통신 기능

Fig. 4는 개발된 프로그램에서 직렬 통신, TCP 프로토콜 2가지가 동시에 연결된 상태를 나타낸다.

No	Type	연결 이름	Time	Tx	Rx	X
2	SERIAL	COM3	17:10:58	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	SERIAL	COM4	17:11:02	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	TCP Client	127.0.0.1	17:13:12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. 4. Multiple Connection Example

이 화면에서 각 연결은 독립된 객체로 각 연결이 독립되어 연결되어 동작하도록 구현되어 있기 때문에 각 연결 간 간섭 없이 데이터 송·수신이 가능하다. 뿐만 아니라 Tx, Rx 체크박스 옵션을 통해 각 연결에 대한 선택적 송·수신이 가능하다. 이 기능을 통해 각 장치 간에 독립된 데이터 송·수신을 보장할 수 있다.

Fig. 5는 제안한 에뮬레이터에 연결된 장치에 전송 주기를 설정하는 화면을 표시한 것이다. 주기설정은 Fig. 5와 같이 'ms 반복' 오른쪽 입력창에 주기를 입력한다. Fig. 5의 예제에는 1000을 입력하였고, 이는 1초 마다 데이터를 전송하라는 의미이다.

#### 3.2.3 매크로 송신 기능



Fig. 5. Macro transmission function

### 3.2.4 통신 빈도 그래프 기능

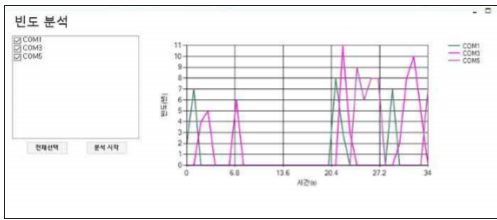


Fig. 6. Communication frequency graph function

Fig. 6은 연결된 장치 간에 통신 빈도를 그래프로 나타낸 화면이다. 통신 빈도 그래프는 에뮬레이터와 연결된 장치들이 연결이 된 시점 이후로 데이터를 송·수신한 빈도를 출력한다. 빈도를 확인하려는 장치의 체크 박스를 선택하면 해당 연결에 대한 조회가 가능하다.

단일 장치와의 연결에서는 통신한 시점을 쉽게 확인할 수 있지만, 여러 장치와의 연결에서는 파악하기 어려우므로, 통신 빈도 그래프를 이용하면 여러 연결 장치의 송·수신 빈도의 비교를 손쉽게 그래프로 확인할 수 있다.

### 3.2.5 로그 저장 및 출력



Fig. 7. Log storage and output function

Fig. 7은 에뮬레이터가 기록한 로그를 저장 및 출력하는 화면이다. 로그 저장 및 출력은 지금까지 장치들과 통신한 내용을 외부파일로 출력할 수 있는 기능이다. 외부 파일은 .Log 확장자로 생성되며, 이미 저장된 로그 파일을 프로그램에서 불러와 조회하는 것도 가능하다.

로그기능은 장치들과의 통신 이력을 보존할 수 있어, 에뮬레이터를 다시 실행하지 않아도 이전의 로그 파일을 확인하여 시험 과정을 추적할 수 있다.

## 4. 다중 연결 실험 및 결과

이 장에서는 여러 IoT 통신 장치와 프로그램을 연결한 실험과정과 실험 결과에 대해서 기술하고 있다.

### 4.1 장치 스펙 및 연결 옵션

이 절에서는 실험에 사용한 장치들의 스펙과 연결 옵션에 대해서 기술한다.

표 1은 실험에 사용된 장치의 하드웨어적 스펙을 나타내고 있다. 선정된 통신 장치는 실제 IoT 관련 프로그램 개발에 많이 쓰고 있는 아두이노 3대와 직렬 통신으로 사용가능한 블루투스 모듈인 HC-06 3개를 각각 연결하여 사용한다. 아두이노에서는 타이머 인터럽트를 통해 정확히 10ms 주기로 데이터인 "L1" 문자 데이터를 자동으로 전송하도록 시험용 프로그램이 설치되어 있다. 아두이노와 에뮬레이터의 직렬 통신 연결 설정 옵션을 9600 8N1을 사용한다.

Table 1. Experiment environment

Device	Type of device	contents
Arduino x 3ea	Board Name	Arduino UNO
	Communication module	HC-06 x 3ea
	Data transmission cycle	10ms
	Size of transmitted data	"L1" (2byte)
PC	CPU	i3-4030U (1.90GHz)
	RAM	8Gb
	OS	Windows 10

### 4.2 다중 연결 구성 환경

이 절에서는 다중 연결을 위한 구성 환경에 대해 기술한다.

#### 4.2.1 장치 구성 및 연결

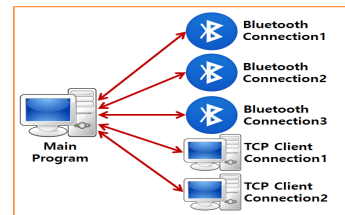


Fig. 8. Experiment environment

Fig. 8은 실험에 쓰인 장치의 구성이다. 장치구성을 위한 연결 설정은 4절에서 제시한 연결 절차를 따라 각 장치와 연결을 수행한다.

3개의 블루투스 모듈은 9600 8N1 통신 설정에 맞춰 프로그램에 각각 연결 설정한다. TCP 연결은 메인 프로그램

램을 서버로 설정한 뒤, 나머지 2개의

프로그램은 서버의 주소인 127.0.0.1을 입력하여 클라이언트로 설정하여 연결 설정한다.

### 4.2.2 실험 절차

Case 1 : 다중 연결 및 송·수신 테스트

Main Program과 연결된 장치들에 데이터를 송신할 때 모든 장치가 동일한 데이터를 동시에 받을 수 있는 지 테스트한다.

Case 2 : 다중 연결 부하 테스트

Main Program에 연결된 3개의 블루투스 장치에 설치된 테스트 프로그램에서 10ms 주기로 데이터를 동시에 전송해도 프로그램이 부하 없이 사용가능한지 테스트한다.

## 4.3 다중 연결 실험 결과

이 절에서는 다중 연결 실험 결과에 대해서 기술한다.

### 4.3.1 Case 1 실험 결과

Fig. 9를 통해서 메인 프로그램에 2개 이상의 연결 방식의 통신 장치들이 동시에 연결될 수 있음을 확인할 수 있다.



Fig. 9. 메인 프로그램에서의 전송

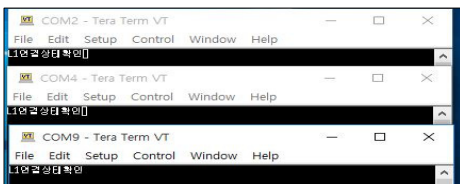


Fig. 10. 직렬 통신 수신 성공

Fig. 10은 에뮬레이터에서 메시지를 전송하면 동시에 3개의 직렬 통신이 문제없이 수신한 상태를 표시한 것이다. Fig. 11은 에뮬레이터에서 메시지를 전송하면 동시에 2개의 TCP 연결에 올바른 데이터가 Fig.과 같이 출력되었다.



Fig. 11. TCP 수신 성공

### 4.3.2 Case 2 실험 결과

Fig. 12를 통해 블루투스 장치 3개 모두 메인 프로그램과 연결되어 있음을 확인할 수 있다. 그리고 3개 장치에서 10ms 주기로 보내는 데이터를 문제없이 처리하여 출력하고 있다. 뿐만 아니라 3개의 장치에서 데이터 전송을 오랜 시간 지속하여도 프로그램이 멈추거나 종료되는 일은 발생하지 않는다.

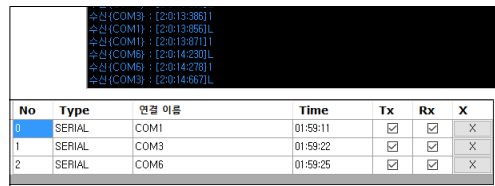


Fig. 12. 다중 연결 부하 테스트 결과

## 5. 결론

본 논문에서 제안한 다중 연결이 가능한 터미널 에뮬레이터는 직렬 통신, TCP, UDP 연결 프로토콜을 지원하며, 두 개 이상의 연결을 동시에 지원하는 터미널 에뮬레이터이다.

실험에서는 구현한 에뮬레이터를 통해 직렬 통신과 TCP/UDP 통신으로 구성된 다중 연결 환경을 구성하였고, 연결된 장치들과의 다중 데이터 송·수신을 성공적으로 수행하였다. 또한 시험용 프로그램을 이용해 다중 연결 환경에서 발생할 수 있는 부하에서도 프로그램이 원활하게 사용 가능함을 확인하였다.

구현한 에뮬레이터는 다수의 연결 장치에 동시에 같은 형식의 데이터를 전송하고, 각 장치로부터의 데이터를 수신할 수 있다. 이 에뮬레이터는 다수의 장치를 운용할 때 발생할 수 있는 여러 가지 불편함을 고려해 매크로 송신 기능, 통신 빈도 그래프 출력 기능, 로그 저장 및 출력 기능을 추가하였다.

이처럼 이 논문에서 제시하는 프로그램을 활용한다면 IoT 시스템에 사용할 수 있는 다양한 장치들을 통합 및

관리할 때 유용할 것이다. 향후에도 이 에뮬레이터가 IoT 개발 도구로서의 충분한 개발 지원이 되도록 지속적으로 기능에 대한 연구를 수행할 것이다.

## ACKNOWLEDGMENTS

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 서울어코드 활성화 지원사업(IITP-2017-2012-0-00598)의 연구결과로 수행되었음.

이 논문은 2017년 추계학술대회 학부생 발표 논문 중 우수논문으로 선정되었습니다.

## REFERENCES

- [1] S. T. K, W. Y Lee, S. J. H & I. W. Joe. (2017). *ABi-directional Communication Scheme based on BLE for IoT Networks*. KOREA SCIENCE. [http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBGHKW\\_2016\\_v43n6\\_710](http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBGHKW_2016_v43n6_710)
- [2] Postscapes. (2015). *Best Internet of Things Definition*. Postscapes. <https://www.postscapes.com/internet-of-things-definition>
- [3] Jun Zheng & Abbas Jamalipour. (2009). *Wireless Sensor Networks*. New York : John Wiley & Sons.
- [4] LG CNS. (2015). *What does it take to implement the IoT*. LG CNS. <http://blog.lgcns.com/818>
- [5] G. h. Hong, E. S. Park, S. h. Choi & D. K. Shin. (2016). *Smart Fog : Advanced Fog Server-centric Things Abstraction Framework for Multi-service IoT System*. KOREA SCIENCE. [http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBGHKW\\_2016\\_v43n6\\_710](http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBGHKW_2016_v43n6_710)  
DOI : 10.5626/JOK.2016.43.6.710
- [6] W. Y. Chang, H. S. Kim, Y. C. Lee & S. J. Yoon. (2016). *Implementation of the Open Hardware-based Gateway for providing Multiple Connections IoT Services*. IEIE. <http://www.ieek.or.kr/>
- [7] ITU. (2012). *Overview of the Internet of things*. International Telecommunication Union. <https://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth>
- [8] H. B. Jo & M. H. Kim. (2014). *Into information : IoT technology and domestic and overseas service development status*. 84, 92-99. Gyeonggi : Koreanstudies

Information Service System.

- [9] Wikipedia. (2017). *Terminal emulator list*. Wikipedia. [https://ko.wikipedia.org/wiki/%EB%8B%A8%EB%A7%90\\_%EC%97%90%EB%AE%AC%EB%A0%88%EC%9D%B4%ED%84%B0\\_%EB%AA%A9%EB%A1%9D](https://ko.wikipedia.org/wiki/%EB%8B%A8%EB%A7%90_%EC%97%90%EB%AE%AC%EB%A0%88%EC%9D%B4%ED%84%B0_%EB%AA%A9%EB%A1%9D)
- [10] National Instruments. (2008). *Serial Communication Overview*. National Instruments. <http://www.ni.com/white-paper/2895/ko>
- [11] G. H. Park. (2013). *Data communication and computer networks*. Tistory. <http://janlssary.tistory.com/55>

## 저 자 소 개

최 흥 석(Hong-Soek Choi)

[학생회원]

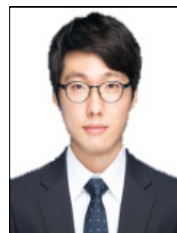


- 2017년 12월 : 충북대학교 소프트웨어학과 재학중

<관심분야> : EPL, 아두이노, 컴퓨터교육

김 윤 수(Youn-Su Kim)

[학생회원]

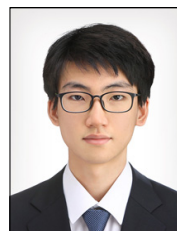


- 2017년 12월 : 충북대학교 소프트웨어학과 재학중

<관심분야> : 펌웨어, 스마트팩토리, 알고리즘

김 진 수(jin-Su Kim)

[학생회원]



- 2017년 12월 : 충북대학교 소프트웨어학과 재학중

<관심분야> : 유니티, 닌텐도, 게임기획

주 영 관(Youngkwan Ju)

[정회원]



- 1999년 2월 : 청주대학교 컴퓨터정보공학과 학사
- 2004년 2월 : 충북대학교 전자계산학과 석사
- 2009년 2월 : 충북대학교 전자계산학과 박사
- 2009년 4월 ~ 2011년 7월 : (주)보승지아이에스 연구소장
- 2009년 3월 ~ 현재 : 충북대학교 전자정보대학 강사  
<관심분야> : 임베디드 시스템, 안드로이드, 클라우드 컴퓨팅 등

전 중 남(Joongnam Joen)

[정회원]



- 1990년 2월 : 연세대학교 전자공학과 박사
- 1996년 8월 ~ 1998년 7월 : 미국 Texas A&M 연구교수
- 2008년 3월 ~ 2010년 2월 : 충북대학교 전자정보대학 학장
- 1990년 9월 ~ 현재 : 충북대학교 전자정보대학 교수  
<관심분야> : 컴퓨터구조, 임베디드 시스템