

무선 데이터 전송과 웹 로깅 시스템 설계

Design of Wireless Data Transmission and Web Logging System

노재성

서울대학교 정보통신공학과

Jae-sung Roh

Department of Information & Communication Engineering, Seoul University, Seoul, 02192, Korea

[요 약]

최근 수년간 사물인터넷 기술은 급격히 발달하고 있으며 많은 분야에서 활용되고 있다. 사물인터넷 디바이스의 기능은 다양한 환경에서 정보를 수집하는 것이며 사용자에게 유용한 정보를 제공하여야 한다. 최근에는 사물인터넷 디바이스의 종류가 증가하고 있으며 가격도 저렴해지고 있다. 또한 사용하기 쉬운 오픈 소스 플랫폼들이 많이 개발되어 사용자에게 편의를 제공하고 있다. 본 논문에서는 ESP-12E NodeMCU 모듈 기반의 무선 데이터 전송 및 클라우드 서버 시스템을 설계하였으며 사물인터넷 디바이스에서 수집된 센서 정보는 WiFi 무선통신을 통하여 클라우드 서버에 전송하였다. 수집된 센서 데이터는 클라우드 서버에서 채널과 필드를 생성하여 저장되며 각 필드의 데이터를 그래프로 시각화하였다.

[Abstract]

In recent years, the Internet of things has grown rapidly and is being used in many fields. The function of an Internet device is to collect information in various environments and to provide useful information to users. Due to recent developments in technology, the kinds of Internet devices are increasing and the prices are getting cheaper. In addition, many open source platforms that are easy to use have been developed and are providing convenience to users. In this paper, wireless data transmission and cloud server system based on ESP-12E NodeMCU module was designed and sensor information collected from Internet devices was transmitted to cloud server through WiFi wireless communication. The collected sensor data was saved by creating channel and field in the cloud server and the data of each field was visualized as a graph.

Key word : WiFi wireless communication, Visualization, Internet of things, Cloud server.

<https://doi.org/10.12673/jant.2018.22.6.636>



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 16 November 2018; Revised 27 November 2018

Accepted (Publication) 21 December 2018 (30 December 2018)

Corresponding Author; Jae-sung Roh

Tel: +82+2+490+7206

E-mail: jsroh@seoil.ac.kr

I. 서론

4차 산업혁명 시대에는 사물 사이의 인터넷 연결을 통하여 사물 주변에서 일어나는 상황을 관찰, 정보 수집, 제어 등을 할 수 있는 사물 인터넷 기술이 주목받고 있다. 사물인터넷은 사물이 센싱과 통신기능을 보유하고 유무선 인터넷에 연결되어 인간과 사물, 사물과 사물 간에 정보를 상호 교환하는 기능을 수행하며 서로 연결된 사물들의 상호작용을 통해 가상세계까지 연결범위를 확장시키고 있다 [1]-[3].

최근에 사용되는 사물인터넷 개발도구는 가격이 저렴하고 사용하기 쉬운 하드웨어와 소프트웨어 기반의 오픈 소스 플랫폼들이 개발되어 제공되고 있다. 현재 사물인터넷을 위한 대표적인 하드웨어로 아두이노, 라즈베리 파이 등이 있으며 아두이노 모듈은 오픈소스를 기반으로 마이크로 컨트롤러와 다양한 센서로부터 값을 받아 처리할 수 있는 기능 개발을 수행할 수 있는 특징이 있다. 특히, 외부 상황을 관찰하는 센서를 활용한 사물인터넷 디바이스는 다양한 환경에서 정보를 수집하고 분석하여 사용자에게 유용한 정보를 제공하는 역할을 수행한다 [4]-[6].

본 논문에서는 센서에서 데이터를 수집하고 이 정보를 시각화하기 위한 방안과 효율적으로 시스템을 구현하기 위한 기술을 연구하였다. 기존에는 센서 데이터의 저장을 ATmega128 보드와 SD(secure digital) 카드를 사용하여 데이터를 수집하였다. 또한 원격지에서 실시간으로 데이터를 확인하는 것이 어렵기 때문에 고가의 전용 서버를 구축하여 데이터 로깅 시스템을 구축하였다. 하지만 본 논문에서는 가격이 저렴하고 WiFi 기능이 있는 ESP-12E NodeMCU 모듈을 사용하여 작업공간의 온도 및 습도 데이터를 취득하고 이 정보를 AP(access point)를 통하여 ThingSpeak 사이트의 클라우드 기반 서버에 전송하였다. 서버에서는 채널에 따른 필드가 생성되며 시계열 데이터가 저장된다. 그리고 웹 브라우저를 통하여 데이터를 전송하기 위하여 공개 API(application program interface)를 사용하여 15초 간격으로 데이터를 저장하였다. 전송받은 센서 데이터를 이용하여 다양한 장치가 작동될 수 있으며 이 데이터를 사용자들이 볼 수 있도록 시각화하였다.

II. 사물인터넷 플랫폼 동향

사물인터넷이 이슈가 되면서 리눅스나 기타 오픈 소스를 활용하는 다양한 하드웨어들이 등장하고 있으며 아두이노, 라즈베리 파이, 코드 버그, BBC micro:bit, ESP-12E NodeMCU, 리틀비즈, 인텔 갈릴레오, 에디슨 등 다양한 플랫폼이 교육 및 개발을 위해 사용되고 있다.

임베디드 보드를 사용하기 위해서는 디바이스 드라이버 설치와 터미널 설정 등의 복잡한 작업을 수행해야 했지만 아두이노 기반의 보드는 개발환경인 스케치만 설치하면 절차들이 자

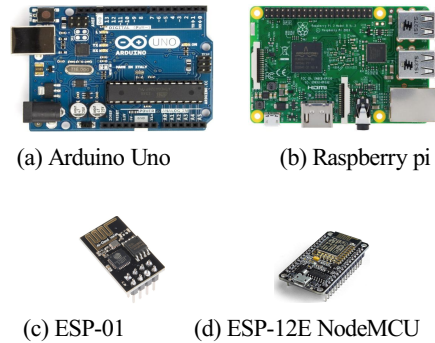


그림 1. 사물인터넷 플랫폼
Fig. 1. IoT platform.

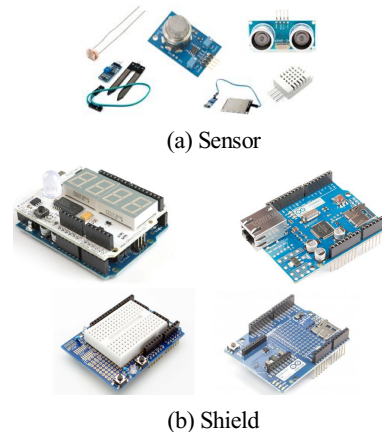


그림 2. 사물인터넷 플랫폼을 위한 센서와 쉴드
Fig. 2. Sensor and shield for IoT platform.

동으로 설정되어 쉽게 개발할 수 있는 장점이 있다. 그림 1은 사물인터넷 플랫폼(아두이노 우노, 라즈베리 파이, ESP-01, ESP-12E NodeMCU)을 제시하고 있다 [7]-[10]. 이중에서 아두이노는 다양한 모델을 제공하고 있어서 산업 현장에서도 많이 사용되고 있다. 또한 구글에서는 안드로이드의 사물인터넷 연결 기기로 아두이노 메가를 선정하고 안드로이드 NDK(native development kit)를 기반으로 개발환경을 지원하고 있다. 라즈베리 파이는 영국에서 컴퓨터 과학 교육을 목적으로 낮은 가격의 판매, ARM CPU의 빠른 속도, 리눅스 사용 등으로 많은 인기를 끌고 있으며 카메라와 7인치 터치 디스플레이를 제공한다. 다양한 하드웨어와의 연결을 위해 GPIO (general purpose input/output)이 제공되고 있어서 사물인터넷을 위한 개발용으로 널리 사용되고 있다. 그림 2는 사물인터넷 모듈을 효율적으로 사용하기 위한 다양한 센서와 쉴드를 나타낸다.

III. 시스템 구성 및 설계

3-1 시스템 구성

본 논문에서 제시하는 ESP-12E NodeMCU 모듈 기반의 무선 데이터 전송 및 클라우드 서버 시스템은 그림 3과 같은 구성으로 설계되었다. 사물인터넷 디바이스에서 수집된 센서 정보는 WiFi 모듈과 AP를 통하여 클라우드 서버에 전송되며 수집된 데이터는 클라우드 서버에 저장되어 그래프로 시각화된다. ESP-12E NodeMCU 모듈은 펌웨어에 따라서 작은 컴퓨터로 동작할 수 있어서 소형 및 저가 사물인터넷 장비를 설계할 수 있다. 그리고 이 모듈의 가장 큰 장점은 WiFi 통신 기능이 ESP-12E NodeMCU 모듈에 포함되어 펌웨어의 변경만으로 WiFi 통신 기능을 사용할 수 있다. ESP-12E NodeMCU 모듈의 입력 및 출력 핀 구성은 그림 4와 같다. 다양한 센서 데이터를 GPIO를 통해 제공받고, 아날로그 데이터도 받을 수 있으며 내부 Flash memory에 특정 펌웨어를 업로드하면 MCU(micro controller unit) 컨트롤 기능이 가능하다.

3-2 무선 데이터 전송/클라우드 로깅 시스템 설계

ESP-12E NodeMCU 모듈을 사용한 웹 클라이언트는 웹 서버로부터 데이터를 송수신할 수 있는 기능을 보유하고 있다. 웹 서버로 데이터를 전송하기 위해서는 웹 서버에서 데이터를 받아서 처리할 수 있는 웹 페이지와 데이터를 저장할 수 있는 데이터베이스 등이 필요하다. 웹 클라이언트의 기능은 로그인 정보의 전송, 게시판 글 작성, 파일 업로드 등이 있으며 HTTP (hypertext transfer protocol)을 사용하고 있다.

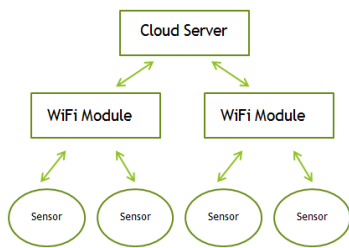


그림 3. WiFi 기반 통신 네트워크 구성도
 Fig. 3. WiFi-based communication network configuration diagram.

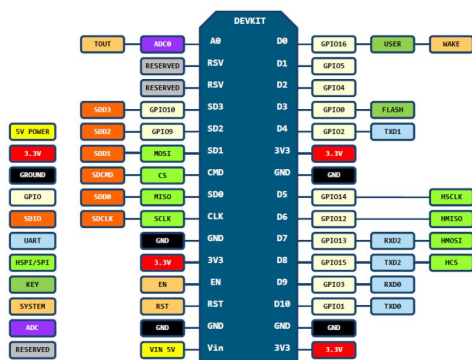


그림 4. ESP-12E NodeMCU 핀 구성
 Fig. 4. ESP-12E NodeMCU pin configuration.

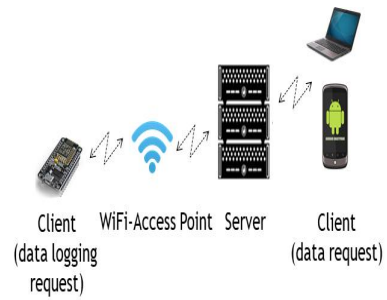


그림 5. 무선 데이터 전송 및 서버 로깅 시스템 구성
 Fig. 5. Wireless data transfer and server logging system configuration.

그림 5는 무선 데이터 전송 및 서버 로깅 시스템의 구성을 나타낸다. 다양한 센서 데이터를 기록하고 저장하는 데이터 로깅은 사물인터넷에서 반드시 필요한 부분으로 전용 웹서비스를 사용하여 전송된 센서 데이터를 저장하고 데이터를 그래프로 모니터링이 가능하다. 그림 6은 온습도 정보 모니터링을 위한 회로를 나타낸다. 회로에서 상태 LED와 디지털 온습도 센서를 기본으로 장치를 구성하였으며 데이터 핀에 5.1[kΩ]의 풀업 저항을 사용하였다. 디지털 온습도 센서는 그림 7의 DHT 22 모듈을 사용하였다. DHT22 모듈은 습도와 온도를 측정할 수 있는 디지털 센서로 3.3[V]에서 동작하며 온도와 습도를 함께 측정할 수 있다. 또한 하나의 데이터 디지털 핀만을 사용하므로 다른 센서부품에 비해 편리한 장점이 있다. DHT22 센서는 4개의 핀으로 구성되며 전원 핀(VCC, GND) 2개, 데이터 출력 핀 1개, 사용하지 않는 핀 1개이다. 1개의 데이터 핀만을 필요로 한다는 점에서 1-와이어 통신과 유사하며 DHT 시리즈 센서에서는 1:1 연결만 지원하는 단순화된 방법을 사용하고 있다.

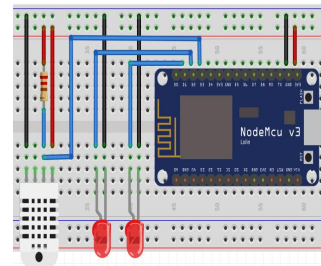


그림 6. 환경 정보 모니터링을 위한 ESP-12E NodeMCU 회로
 Fig. 6. ESP-12E NodeMCU circuit for environmental information monitoring.

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



그림 7. 온도/습도 모니터링을 위한 DHT22 핀 구성
 Fig. 7. DHT22 pin configuration for temperature/humidity monitoring.

DHT22 센서로부터 데이터를 얻기 위하여 5초 이상의 간격으로 ESP-12E NodeMCU 모듈의 데이터 핀으로 센서 값을 읽어 들이며 2바이트의 습도 데이터, 2바이트의 온도 데이터, 1바이트의 패리티 데이터로 구성된다. ESP-12E NodeMCU 보드의 경우 USB-UART 변환 장치뿐만 아니라 아두이노 환경에서 펌웨어를 업로드할 때 리셋 버튼과 GPIO 0 핀에 연결된 버튼을 누르지 않고도 자동으로 펌웨어 업로드를 진행시켜 주는 회로가 포함되어 있다. 그림 8은 아두이노 IDE 환경에서 제공하는 시리얼 플로터 화면을 나타낸다. 시간이 흐름에 따라서 그림 6의 ESP-12E NodeMCU 회로를 통하여 수신된 온도와 습도 데이터가 각각 시각화된다.

ESP8266 컨트롤러와 WiFi 통신 기능을 사용하기 위해 전용 라이브러리인 ‘ESP8266WiFi.h’를 사용하여 AP와 통신을 진행한다. AP에 연결하기 위해서는 AP 이름인 SSID와 패스워드가 필요하며 패스워드가 설정되어 있지 않은 공개된 AP의 경우에는 패스워드 없이 연결이 가능하다. 프로그램에서 ESP-12E NodeMCU 모듈과 연결하는 AP는 연결되는 스테이션에 자동으로 IP(internet protocol) 주소를 할당할 수 있는 DHCP(dynamic host configuration protocol) 기능을 지원한다. DHT22 센서 프로그램을 작성하기 위해서는 DHT 라이브러리 Adafruit Unified Sensor Library2가 필요하며 다양한 센서들의 출력 단위를 통일시켜 센서 사용의 편의성을 제공한다. 프로그램에서는 ‘DHT.h’ 파일을 포함시키고 전역 객체를 생성해야 하며 객체 생성 시 DHT22 센서가 연결된 핀과 센서의 종류를 지정해야 한다. 웹 브라우저를 통해 서버측 채널의 필드에 데이터를 전송하는 경우에는 그림 9와 같이 입력하면 데이터가 15초 간격으로 기록된 시간과 함께 해당 채널에 데이터가 자동으로 저장된다.

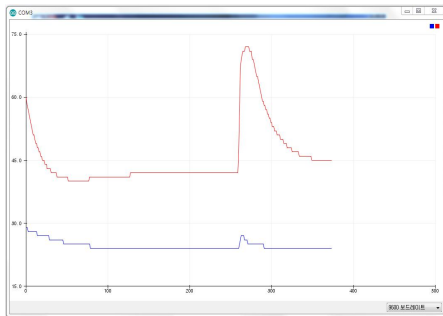


그림 8. 아두이노 IDE의 시리얼 플로터 시각화
Fig. 8. Serial plotter visualization of Arduino IDE.

```
http://api.thingspeak.com/update?key=write_key&field1=record_value
http://api.thingspeak.com/update?key=write_key&field2=record_value
http://api.thingspeak.com/update?key=write_key&field1=record_value1
&field2=record_value2
```

그림 9. 웹 브라우저를 통한 데이터 전송 형식
Fig. 9. Data transmission format via web browser.

데이터를 서버에 입력하는 방법은 그림 9와 같이 웹 브라우저를 사용하여 데이터를 입력할 수 있으며 ESP-12E NodeMCU 모듈을 사용하여 일정한 시간 간격으로 자동으로 데이터를 로깅할 수도 있다. 자동으로 데이터를 입력하기 위해서는 ESP-12E NodeMCU 모듈을 웹 브라우저의 역할을 수행할 수 있는 웹 클라이언트로 사용한다. 이를 위해 웹 클라이언트로 사용할 수 있는 클래스 중 HTTPClient를 사용하여 DHT22 센서를 통해 온도와 습도 정보를 20초 간격으로 받으며 데이터 로깅을 위한 ThingSpeak 사이트 채널의 필드에 저장한다. 본 논문에서는 사용한 HTTPClient 클래스는 웹 페이지의 주소를 사용하여 데이터를 로깅하며 HTTP 프로토콜에 특화되어 있어서 통신 절차가 직관적이고 간단하다. 그림 10은 HTTPClient 클래스로 구현한 코드를 나타낸다. 그림 9와 비교하면 유사한 구조를 가지고 있으나 통신모듈 내부에 소스코드가 내장되어 수행되기 때문에 자동적으로 서버측 채널의 필드 1과 필드 2에 데이터 값 1과 데이터 값 2가 저장된다.

그림 11은 ESP-12E NodeMCU 모듈을 웹 브라우저로 사용하여 ThingSpeak 사이트 채널의 필드에 데이터를 전송/로깅하는 절차를 나타낸다. 구현한 데이터 전송/로깅 프로그램은 크게 setup()과 loop() 부분으로 구성된다. setup() 부에서는 센서 초기화, WiFi 모드 및 기존 연결 종료, AP 접속을 시도한다. loop() 부에서는 온도 및 습도 데이터 읽기, DHT 센서값 검사, HTTPClient 기능 구현 및 웹서버 응답 기능을 구현하였다. 최근 클라우드를 이용한 데이터 로깅을 지원하는 웹 서비스가 증가하고 있는 추세이며 ThingSpeak 사이트에서는 데이터 전송을 위한 공개 API와 전송된 데이터를 저장하는 공간을 제공하고 있다. ThingSpeak 서비스에서 데이터 로깅은 채널 단위로 이루어지며 새로운 채널을 생성하여 데이터를 로깅할 수 있다. 또한, 저장된 데이터를 시각화할 수 있는 다양한 기능을 제공하여 스마트폰의 앱을 통해 로깅된 데이터를 확인할 수 있다. ThingSpeak 서비스에서 채널은 최대 8개의 필드를 가질 수 있으며 각 필드에는 다른 종류의 데이터를 기록할 수 있다. 그림 12는 필드 1과 필드 2에 저장된 로깅 데이터를 시각화하고 있다. 필드에 기록되는 모든 데이터는 시간 정보와 함께 기록되므로 시간에 따른 데이터의 변화를 쉽게 확인할 수 있다.

```
HTTPClient httpClient;

String host = "http://api.thingspeak.com";
String url = "/update?key=";
String key = "write_key";
String field1 = "&field1=";
String field2 = "&field2=";

httpClient.begin(host + url + key + field1 + record_value1 + field2
+ record_value2);
```

그림 10. HTTPClient 부분의 코드
Fig. 10. Code of HTTPClient part.

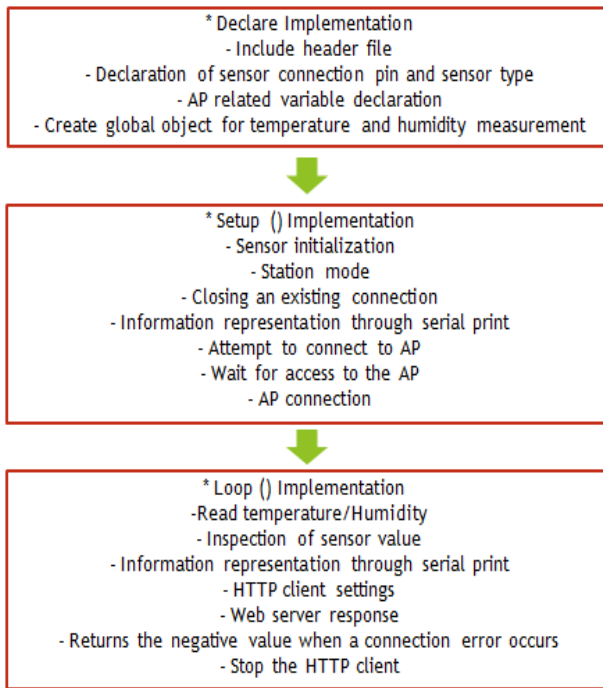


그림 11. 데이터 전송/로깅 프로그램 흐름도
Fig. 11. Data transmission/logging program flow chart.

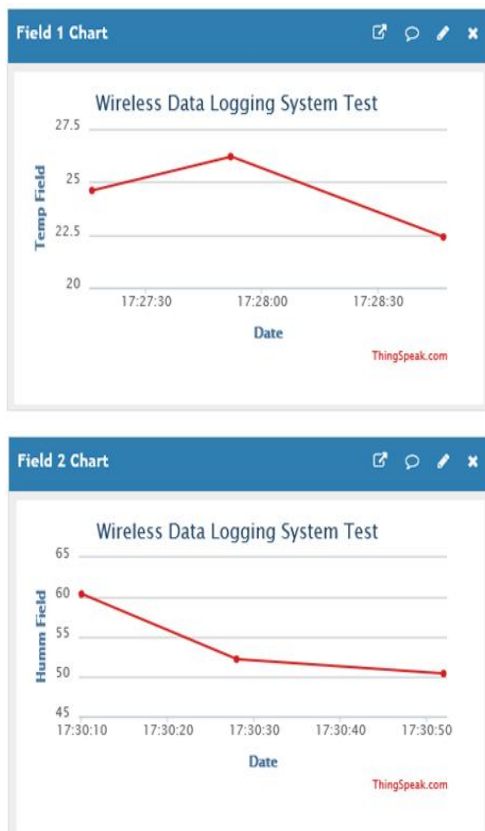


그림 12. 로깅 데이터 시각화
Fig. 12. Visualization of logging data.

IV. 결론

본 논문에서는 ESP-12E NodeMCU 모듈 기반의 무선 데이터 전송 및 클라우드 서버 시스템을 설계하였으며 디바이스에서 수집된 센서 정보는 WiFi 통신 기능을 통하여 클라우드 서버에 전송하였다. 수집된 센서 데이터는 클라우드 서버에서 채널과 필드를 생성하여 저장되며 각 필드의 데이터를 그래프로 나타내었다. 설계된 무선 데이터 통신장치는 안드로이드 앱이나 웹을 통해서 사용자에게 실시간으로 다양한 정보를 전달할 수 있으며, 각종 데이터를 서버에 전달함으로써 하나의 플랫폼으로 여러 데이터를 처리하는데 유용하게 활용할 수 있을 것이다. 향후 연구로는 다수의 사물인터넷 보드에서 받은 센서 데이터 정보를 보다 효율적으로 클라우드 서버에 등록할 수 있는 구조에 대한 연구를 진행하고자 한다. 또한 다양한 센서의 적용과 확보된 데이터를 가치 있는 정보로 표현하는 방법을 더욱 연구해야 할 것으로 사료된다.

Acknowledgments

본 논문은 2018년도 서일대학교 학술연구비에 의해 연구되었음.

References

- [1] S. H. Kim, "Internet of things technology," *The Institute of Electronics and Information Engineers*, Vol. 43, No. 3, pp. 64-71, 2016.
- [2] D. Zeng, S. Guo, and Z. Cheng, "The web of things : a survey," *Journal of Communications*, Vol. 6, No. 6, pp. 424-438, 2011.
- [3] M. K. In, K. C. Lee, and S. Y. Lee, "Web of things (WoT) standardization trends," *Journal of the Korean Society of Mechanical Engineers*, Vol. 138, pp.79-84, 2011.
- [4] Y. Fan, Y. Yin, L. Xu, Y. Zeng, and F. Wu, "IoT-Based smart rehabilitation system", *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1568-1577, 2014.
- [5] L. Jiang, L. Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT - oriented data storage framework in cloud computing platform," *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1443-1451, 2014.
- [6] G. Y. Heo, *ESP8266 Programming*, 1st ed. Seoul, Korea: Jpub Publishing, 2017.
- [7] D. S. Lee and J. S. Lim, "Design of compact data integration and convergence device using esp8266 module," *Journal of the Korean Convergence Society*, Vol. 8, No. 2, pp.15-20,

- 2017.
- [8] C. S. Jeong and M. R. Kwon, "A study on safety management of day care center using disaster management system," *Journal of the Institute of Internet, Broadcasting and Communication*, Vol. 18, No. 1, pp.29~35, 2018.
- [9] M. Margolis, *Arduino Cookbook*, 2nd ed. California, CA: O'Reilly Media, 2012.
- [10] M. Schwartz, *Internet of Things with the Arduino Yun*, 1st ed., Birmingham, United Kingdom: Packt Publishing, 2014.



노재성 (Jae-Sung Roh)

2000년 8월 : 한국항공대학교 정보통신공학과 (공학박사)
2000년 9월 ~ 현재 : 서일대학교 정보통신공학과 교수
※관심분야 : 무선 임베디드 시스템 설계, 사물인터넷 S/W 응용