

IoT 홈 허브 및 장치에서 무선 공유기 네트워크 설정을 위한 반응형 웹 사용자 인터페이스 서비스 제공 방법

손장우*, 홍승모^o

Provision of Responsive Web User Interface Service for Wireless Router Network Setting in IoT Home Hubs and Devices

Jang Woo Shon*, Seung Mo Hong^o

요 약

사물 인터넷 (Internet of Things: IoT) 장치는 인터넷에 연결되어 서버와 통신을 하게 된다. 사물에 부착된 센서로부터 정보를 얻기 위해 고사양의 와이파이, 하드웨어와 소프트웨어로 구성된 라즈베리 플랫폼을 사용하는 것이 일반적이다. 본 논문에서는 가격이 저렴하고 대중적인 WiFi system on chip (SoC)에서 사용자가 손쉽게 IoT 장치를 공유기에 접속하는데 hypertext markup language version 5 (HTML5), IoT oriented representational state transfer (REST) protocol, Inline JavaScript, document object model (DOM) 및 Inline cascading style sheets (CSS)를 이용한 반응형 웹 사용자 인터페이스 서비스 제공 방법을 제안한다. 이 방법은 IoT 시스템을 구성하는데 있어 최소 비용으로 사용자 편의 인터페이스 제공하는 장점이 있으며, artificial intelligence (AI) IoT 서비스를 위한 제어기와 센서에 최적의 장치 설계 방법이 될 수 있다.

Key Words : WiFi module, hypertext markup language version 5 (HTML5), IoT oriented protocol, representational state transfer(REST) protocol, document object model (DOM)

ABSTRACT

Internet of things (IoT) devices are connected to the internet and communicate with servers. It is common to use the raspberry platform, which consists of high-end WiFi, hardware and software to obtain information from sensors attached to objects. In this paper, the user can easily connect the IoT device to the router in the WiFi system on chip (SoC). We propose a method for providing a responsive web user interface service using hypertext markup language version 5 (HTML5), IoT oriented representational state transfer (REST) protocol, Inline JavaScript, document object model(DOM), and Inline cascading style sheets (CSS). This method has the advantage of providing a user-friendly interface at a minimum cost in configuring an IoT system, and can be an optimal device design method for controllers and sensors for artificial intelligence (AI) IoT services.

* First Author : Soongsil University Department of Smart System Software, jwshon@ssu.ac.kr, 정회원

^o Corresponding Author : Soongsil University Department of Telecommunication Engineering, omnu@ssu.ac.kr, 정회원
논문번호 : 201911-289-D-RU, Received November 7, 2019; Revised November 22, 2019; Accepted December 2, 2019

I. 서론

사물 인터넷은 사물과 인터넷이 결합한 기술로서, 센서 네트워크와 임베디드 프로세서와 같은 몇 가지 초기 기술에 뿌리를 두고 있다. 시스템을 구현하기 위해서는 하드웨어와 소프트웨어를 구성하고 기능에 따른 최적의 시스템을 구축하는데 어려움이 있다. 쉽게 구입 가능하면서 비용을 최소화시켜서 투자비용과 확장성을 고려해야 한다. 따라서 IoT 시스템에서 사용할 디바이스는 소량 구입에 제약이 없고 가격이 저렴해야 한다. WiFi SoC 칩은 설계 및 제조에 많은 비용과 시간이 요구되지만 대량 생산되어 가격이 저렴하다. 최근의 인공지능 사물인터넷 시스템 구성은 IoT 장치, IoT 허브를 최초 설치하거나 장소 이동할 경우 wide area network (WAN) 접속을 해야 하는데, 초기에 공유기의 service set identifier (SSID)와 password 및 기타 설정을 하는데 어려움이 많이 발생한다. 가격이 저렴하고 IoT 하드웨어로 많이 사용되는 WiFi SoC, 그림. 1은 사물 인터넷에 여러 가지 인터페이스 방법으로 사용되나 편리한 사용자 요구 기능을 구현하는데 어려움이 있다.

종래의 편리한 사용자 요구 기능을 구현하는데 고사양의 하드웨어 라즈베리파이를 이용하여 WAN 접속 시 공유기 SSID와 password 등 기타 설정을 통하여 사용자 편의 기능을 구현하고 있다. 종래의 동작 방법은 IoT 장치 또는 IoT 허브가 wireless access point (WAP)로 동작해야 하는데 라즈베리파이에서 access point (AP) 모드를 지원하지만, 동작하는 방법이 일반 사용자가 사용하기에는 매우 복잡하다. 설정 방법은 라즈베리파이 콘솔 터미널에 접속하고 hostapd 폴더에서 AP 모드를 위한 복잡한 스크립트를 수정하고 재 부팅하는 순서로 사용자 웹브라우저가 1:1 접속할 수 있다. 다시 설정된 SSID와 password를 사용하여 다시 네트워크 관련 스크립트 수정 방법으로 station (STA) 모드로 WAN에 접속된 다른 무선 AP에 접속해야 한다. 이러한 기능은 최근에 인공지능

IoT 장치 또는 IoT 허브가 인터넷 접속하는 과정에서 꼭 필요한 기능이다.

일반 사용자를 위한 웹 사용자 인터페이스 기능을 고사양의 하드웨어, 라즈베리파이에서 IoT 장치나 IoT 허브를 구현하는 것은 소모 전력이 크고 불필요한 하드웨어와 소프트웨어 기능이 있다.^[1-4]

본 논문에서는 가격이 저렴하고 소모 전력이 적으며 작은 크기의 WiFi SoC에서 사용자가 손쉽게 IoT 장치를 공유기에 접속하는 서비스를 제공하는데 WiFi AP 모드에서 HTML5, IoT oriented REST protocol, Inline JavaScript, DOM 및 Inline CSS를 이용한 반응형 웹 사용자 인터페이스 동작 방법과 간단히 WiFi STA 모드로 전환하는 방법을 소개한다.

II. 연구배경

IoT 시스템은 센서 소자에서부터 AI 클라우드 서비스까지 많은 구성 요소로 구성되어 진다. 구성 요소인 IoT 장치와 IoT 허브는 표준화된 프로토콜을 사용해야 하며 가격이 매우 저렴해야 되는 필요성이 대두되고 있다. standalone 환경에서 필요한 기능 모두 동작하기 위해서는 웹서비스를 손쉽게 제공하고 확장이 가능한 데이터와 이벤트 중심의 프로그램 제어가 필요하다.

2.1 고사양 하드웨어 라즈베리파이

IoT 시스템에 많이 사용되는 그림. 2의 고사양 하드웨어 라즈베리파이는 32bit 또는 64bit micro controller unit (MCU), 고용량, 고속의 외부 메모리 그리고 전용 WiFi 모듈로 구성되어 있다.

일반 사용자를 위한 웹 사용자 인터페이스 기능 구현은 그림. 3의 소프트웨어 구성이 필요하다. Web server는 소프트웨어와 하드웨어로 구분된다. 하드웨어는 고사양의 라즈베리파이가 필요하며, 소프트웨어는 Linux operating system (OS), File system, Web

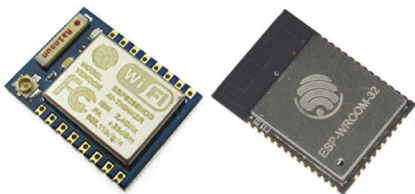


그림 1. WiFi SoC
Fig. 1. WiFi SoC

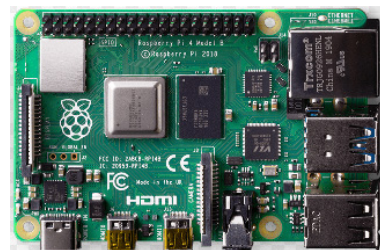


그림 2. 고사양 라즈베리파이 보드
Fig. 2. High spec. raspberry pi board

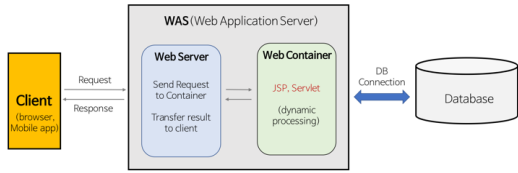


그림 3. Web server와 web application server (WAP)
Fig. 3. Web server and web application server(WAP).

server (APACHE, PHP5, MySQL)을 설치해야 한다. 또한 TCP-IP 네트워크 설정이 필요하다. 웹브라우저 클라이언트로부터 hypertext transfer protocol (HTTP) 요청을 받아 정적인 콘텐츠 (HTML5, CSS, JavaScript)를 제공하는 컴퓨터 프로그램이다. [2]

2.2 IoT oriented 프로토콜

IoT oriented 프로토콜은 bluetooth low energy (BLE), long range (LoRa), message queuing telemetry (MQTT), extensible messaging and presence protocol (XMPP) 및 REST 등이 있다

MQTT는 발행(Publish)과 구독(Subscribe) 구분 (Semantics)이 있는 IoT oriented 프로토콜이며 오버헤드가 적도록 설계되었고 데이터 payload와 무관하다. MQTT는 3가지 수준의 서비스 품질을 제공한다. 최대한 적은 번의 서비스를 제공하고, 한 번은 배달을 보장하지만 중복이 발생할 수 있으며, 정확히 한 번은 복제 없이 메시지가 배달되도록 보장한다. 또한 REST 프로토콜은 웹 서비스에 널리 사용되며 일부 IoT 서비스 모델로 사용된다. REST는 상태 비저장 HTTP 전송을 위한 디자인 패턴이며 디렉터리 구조 양식 자원 표시기를 노출할 수 있다. extensible markup language (XML)와 JavaScript object notation (JSON)은 데이터를 전송하는 데 사용할 수 있다. 클라이언트는 GET, PUT, POST 및 DELETE 방법을 사용하여 자원에 액세스한다.

이러한 방법은 HTTP 프로토콜로 데이터를 전달하는 데이터와 이벤트 중심의 프레임워크에 적합하며 클라이언트/서버 간의 구성 요소를 엄격히 분리하여 구현은 단순화시키고 확장성과 성능을 높이는 아키텍처이다. 따라서 사용자 스마트 폰과 서버 간 자원 분배 관리가 효율적이고 관리가 편리한 마이크로 서비스를 제공한다. 이러한 서비스는 WiFi SoC에서 구현하는데 매우 효율적이다. [3]

III. WiFi SoC에서 반응형 웹 서버 UI 구현 방법

3.1 반응형 웹 서버 UI 구현 시스템

WiFi SoC를 사용한 인터페이스 웹 서비스 전체 시스템의 구성도 그림. 4와 같다.

기본적인 동작 방식은 라즈베리파이에서는 Linux OS, File system, Web server (APACHE, PHP5, MySQL) 소프트웨어들을 설치해야 하지만, WiFi SoC에서는 이러한 소프트웨어들을 설치하지 않고 stand-alone API를 사용하여 웹브라우저 클라이언트로부터 HTTP 요청 받아 정적인 콘텐츠(HTML5, Inline JavaScript, Inline CSS등) 서비스를 제공한다. 따라서 WiFi SoC는 사용자 편이성과 조작성의 기능을 갖추며 웹브라우저 클라이언트는 Point to Point AP 모드로 IoT oriented REST protocol을 사용한 1차 웹 서비스 접속이 가능하다.

웹브라우저 클라이언트와 WiFi SoC의 통신은 인간과 기계가 읽을 수 있는 개방형 표준 데이터 형식 및 객체 표현을 통한 인터넷에서 자료를 주고받는 방법으로 JSON 데이터 쌍을 사용한다. 이러한 방법은 JSON application program interface (API) 함수를 사용하여 구현이 가능하다. 사용자가 스마트폰에서 문자 입력과 버튼 조작을 손쉽게 구현할 수 있다.

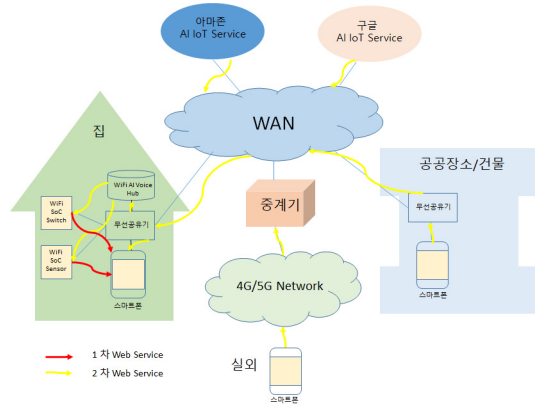


그림 4. WiFi SoC 인터페이스 웹서비스 전체 구성도
Fig. 4. web service configuration of WiFi SoC air interface

3.2 반응형 웹 서버 UI 서비스 동작 순서

그림. 5의 동작 순서에 따라서 WiFi SoC는 공장에서 초기에 설정된 SSID와 password 값을 갖게 된다. 웹브라우저 클라이언트는 2차 web service 접속하기 위하여 WiFi SoC에서 STA 모드로 무선 공유기와 일

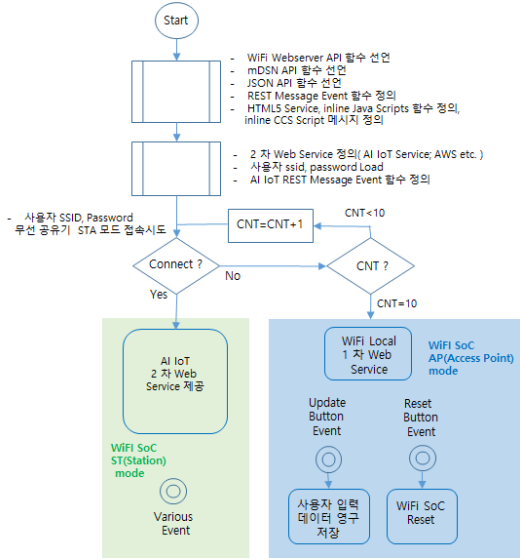


그림 5. Standalone 반응형 웹 서버 동작 순서
Fig. 5. Standalone responsive web server operating sequence

정시간 접속 시도한다. 공장에서 초기에 설정된 SSID와 password는 가정에서 설치된 무선 공유기의 SSID와 password가 다르기 때문에 접속 실패한다. 여러 횟수 접속 실패 후 그림. 6와 같이 WiFi SoC는 AP 모드 1차 web service 기능으로 동작한다.

사용자는 스마트폰의 웹 브라우저를 사용하여 point-point uniform resource locator (URL) 주소 접속과 WiFi SoC의 local domain name service (DNS) 기능을 사용하여 사용자는 알고 있는 무선공유기의 SSID와 password를 입력하고 WiFi SoC에 영구 저장한다. 또한 사용자 편의 리셋 터치 버튼기능을 이용하여 WiFi SoC는 다시 부팅하여 2차 web service 접속을 시도한다.

WiFi SoC는 무선 공유기의 dynamic host configuration protocol (DHCP)로부터 internet protocol (IP) 주소를 얻고 WAN 접속이 이루어진다.

```
Set Default. 10 초 이후, Network 연결 실패
Read Data: default,12345678
1234567890
Connection Failed.
AP Mode Activated.
AP IP : 192.168.4.1
Please connect to this address.
-- AP mode --
```

그림 6. 1차 WiFi SoC 접속 시험 메시지
Fig. 6. Primary WiFi SoC connection message

그림. 7, 8, 9는 WiFi SoC AP 모드 1차 웹서비스 기능으로 동작하고 사용자는 스마트폰의 웹 브라우저를 통하여 point to point URL 접속 및 local DNS 기능을 이용하여 사용자가 알고 있는 무선 공유기의 SSID와 password를 입력하고 WiFi SoC에 영구 저장한다. 이후 다시 사용자 편의 리셋 터치 버튼을 누르고 WiFi SoC를 다시 부팅시킨다.

그림. 9의 시험 메시지는 2차 웹서비스를 제공하는 것으로서 무선 공유기의 DHCP 기능을 통하여 IP 주소를 취득하고 TCP-IP WAN 접속을 나타낸다.

2차 웹서비스는 IoT oriented REST protocol을 이용하여 JSON 형식 데이터 교환과 클라우드 AWS 음성인식 단말 액츄에이터 및 센서 등 여러가지 서비스를 제공할 수 있다.



그림 7. 1차 WiFi SoC 접속 시 스마트폰 WiFi 네트워크 상태 화면
Fig. 7. Smart phone WiFi SoC network status when accessing primary web service

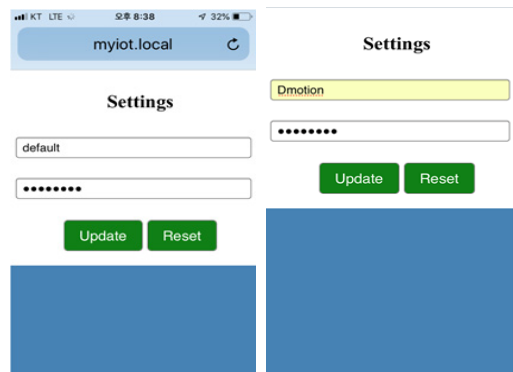


그림 8. 1차 web service 접속
Fig. 8. Primary web service connection

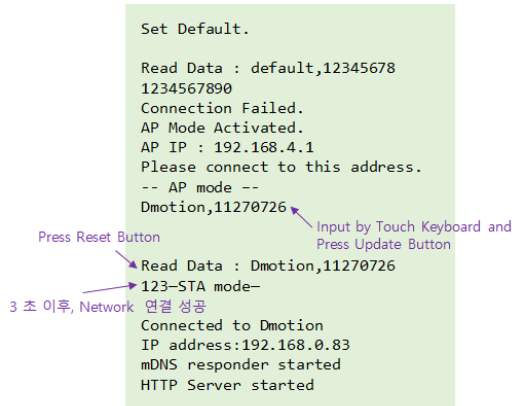


그림 9. 2차 WiFi SoC 접속 시험 메시지
Fig. 9. Secondary WiFi SoC connection message

3.3 반응형 웹 서버 UI 구현 프로그램

WiFi SoC에서 standalone API 함수, HTML5, Inline JavaScript 와 Inline CSS 서비스를 스트링 문자열로 제공하는 방식을 사용하였다. 종래의 사용자의 기능을 구현하기 위해서는 고 성능의 하드웨어와 Linux OS, File system, Web server (APACHE, PHP5, MySql) 소프트웨어들을 설치하고 index.html, style.css, serive.js 등의 여러 개의 파일로 웹 서비스를 제공해야 가능하였다. 그러나 WiFi SoC에서는 구성하기 어렵고 좋은 방법이 아니다.

본 논문에서는 아두이노와 같은 프로그램 개발 환경에서 사용 가능한 웹 서버 API 및 JSON API 함수들을 선언하여 운영체제를 사용하지 않고 standalone 방식으로 동작하는 것이 가능하다. 웹 서비스 중에서 Inline CSS는 사용자 스마트폰의 스크린 크기에 따라서 보기 편한 반응형 웹페이지 동작 구현을 한다.

Inline JavaScript 함수는 WiFi SoC와 스마트폰 사이에 데이터를 JSON 문자열로 교환하는 기능을 한

```
String WebService1 = "\n
<DOCTYPE html>\n
<html>\n
<head>\n
<meta charset='UTF-8' />\n
<title>1st Web Service</title>\n
<script>\n
document.addEventListener('DOMContentLoaded', () => {\n
  document.getElementById('update').addEventListener('click', function(){\n
    fetch('/update', {\n
      method: 'POST',\n
      headers: { 'Content-Type': 'application/json' },\n
      body: JSON.stringify({\n
        ssid: document.getElementById('ssid').value,\n
        pwr: document.getElementById('password').value\n
      })\n
    })\n
  })\n
  document.getElementById('reset').addEventListener('click', function(){\n
    fetch('/reset', {method: 'GET'})\n
  })\n
})\n
</script>\n
</head>\n
<body>\n
<body style='background: steelblue'>\n
  <div style='margin: 48px auto; padding: 24px;\n
    display: flex; flex-direction: column;\n
    background: white' id='main'>\n
    <h1 style='align-self: center'>Settings</h1>\n
    <div style='margin-top: 12px'>\n
    <label style='display: inline-block; margin: 3px 0; font-size: 24px;\n
      width: 240px; text-align: right' for='ssid'>SSID: </label>\n
    <input type='text' name='ssid' id='ssid' style='width: 360px;\n
      font-size: 20px; margin: 12px 12px' value='\n
    </div>\n
    <div style='margin-top: 12px'>\n
    <label style='display: inline-block; margin: 3px 0; font-size: 24px; \n
      width: 240px; text-align: right' for='password'>Password: </label>\n
    <input type='password' name='password' id='password'\n
      style='width: 360px; font-size: 20px; margin: 12px 12px' value='\n
    </div>\n
    <div style='align-self: center; margin-top: 24px'>\n
    <button type='button' id='update' style='font-size: 24px;\n
      padding: 12px 24px; border-radius: 6px;\n
      background: green; color: white'>Update</button>\n
    <button type='button' id='reset' style='font-size: 24px; padding: 12px 24px;\n
      border-radius: 6px; background: green; color: white'>Reset</button>\n
    </div>\n
  </div>\n
</body>\n
</html>"
```

그림 10. Inline JavaScript, Inline CSS 및 HTML 5가 문자열로 변환된 1차 웹 서비스
Fig. 10. Primary web service converted to string with Inline JavaScript, Inline CSS and HTML5

다. 이러한 방법들은 이벤트 중심의 웹 서비스를 제공

표 1. High end embedded 하드웨어와 WiFi SoC 모듈의 성능 및 가격 비교
Table 1. Performance and price comparison of high end embedded hardware and WiFi SoC module

H/W Spec. Low/High	High End embedded		WiFi SoC module	
	Raspberry Pi Zero	Raspberry Pi3B	ESP-8266	ESP-32
CPU Clock	1G	1.2G	80M	160M
ROM	micro-SD	micro-SD	512Kb	16Mb
RAM	1Gb	2Gb	160Kb	512Kb
Boot Time	15~70 sec	15~70 sec	<1sec	<1sec
WiFi AP/STA mode 전환 시간	15~70 sec	15~70 sec	3 sec	3 sec
소모 전류	100~240mA (5V)	100~350mA (5V)	0.5~200mA (3.3V)	0.5~200mA (3.3V)
구입 가격	>\$5	>\$35	>\$0.9	>\$1.2

하기 위한 IoT oriented REST 프로토콜에서 효율적이다.

사용되어지는 스크립트들은 사용자 편의 기능, 반응형 웹페이지, 터치 버튼 그리고 문자열 입력을 위한 스크린 터치 키보드 입력 동작 기능을 위한 것이다. 또한 HTML5 웹페이지에 Inline 방법으로 구현하였다. 그림. 10은 WiFi SoC에서 Inline CSS와 Inline JavaScript를 포함하는 HTML5 웹페이지에서 하나의 문자열(String)로 표현하였다. 하나의 문자열로 표현하기 때문에 standalone 환경에서 WiFi SoC는 쉽게 동작한다.

또한, WiFi SoC에서 JSON API 함수를 사용할 때 사용자 스마트 폰에서 보내온 Base64 엔 코딩된 데이터 분류와 변환이 간단해지며, 데이터와 이벤트 중심의 standalone 프로그래밍이 쉽게 구현 가능하다.

IV. 결 론

본 논문에서는 AI IoT 시스템에서 센서와 웹브라우저 클라이언트 사이의 효율적인 통신 구현 방법을 제시하였다. 대중적인 WiFi SoC를 사용한 저 비용 경량화 구성, HTML5 웹 서비스와 IoT oriented REST protocol, Inline JavaScript, DOM 그리고 Inline CSS를 사용한 반응형 브라우저를 구현하였다. 따라서 사용자가 편리한 웹브라우저 클라이언트에서 직관적으로 표현되고 제어 될 수 있는 것을 확인하였다. 또한, 사용자 편의 공유기 정보 입력을 위하여 WiFi AP 모드 접속과 간편한 WiFi STA 모드 전환 동작이 표 1에서와 같이 기존의 라즈베리파이에 비하여 매우 빠르며 안정적이다. 향후 AI IoT 시스템은 센서, 허브 그리고 구글 및 아마존과 같은 AI 클라우드 서비스를 통합하기 위한 응용 프로그램 개발하는데 기능적으로 꼭 필요하다. 사용자 편의 기능의 반응형 웹 서버 UI 동작 기능은 꼭 필요한 최적화 기능이며 추가적인 다양한 서비스를 개발하는데 필수적이다. WiFi SoC에서 사용자 편의 기능 구성은 소형, 저 전력 그리고 클라우드 AI IoT 서비스를 위한 제어기와 센서에 많이 활용될 것으로 생각되어진다.

References

[1] S. Park, *Make raspberry pi an access point* (2017), Retrieved Sep. 04, 2017 from <https://fishpoint.tistory.com/2159>

[2] *Understanding difference between Web Server*

and WAS(Web Application Server), from <https://gmlwj9405.github.io/2018/10/27/webserver-vs-was.html>

[3] D. Serpanos and M. Wolf, "Internet of things System," *Springer*, pp. 27-86, Sep. 2018.

[4] J. H. Bae and J. T. Kim, "Distributed remote IoT sensor data monitoring and control through the ARTIK cloud" *J. KICS*, vol. 43, no. 12, pp. 2100-2106, Dec. 2018.

[5] B. K. Tripathy and J. Anuradha, "Internet of things technology, application, challenges and solution," *CRC Press*, pp. 17-99, 2005.

[6] H. Lee, "Embedded system framework and its implementation for device to device intelligent communication of manufacturing IoT device considering smart factory," *J. KITS*, vol. 27, no. 5, pp. 459-465, Oct. 2017.

[7] *HTML and CSS, JavaScript*, 10. 15. 2019, from <http://www.w3schools.com.js>

[8] S. I. Hwang, S. Y. Joo, and J. M. Ju, "A study of IoT platform for the smart farm factory," in *Proc. KICS Symp.*, pp. 520-521, Jan. 2016.

손 장 우 (Jang Woo Shon)



1989년 : 숭실대학교 전자 공학과 학사
 1996년 : 숭실대학교 전자 공학과 석사
 1996년 : (주)LG전자 연구소 회로설계팀, 주임연구원
 2018년~현재 : 숭실대학교 IT학부 스마트시스템 소프트웨어학과 부교수

<관심분야> AI IoT, 임베디드 프로세서, 임베디드 모션제어

[ORCID:0000-0002-1128-7145]

홍 승 모 (Seung Mo Hong)



1999년 : 숭실대학교 정보통신
공학과 학사

2001년 : 숭실대학교 정보통신
학과 석사

2008년 : 숭실대학교 정보통신
학과 박사

2018년~현재 : 숭실대학교 IT

학부 전자정보공학부 조교수

<관심분야> 무선통신시스템, 신호처리, 고속 신호처
리 회로 설계

[ORCID:0000-0003-2578-0974]