

<https://doi.org/10.7236/IIBC.2018.18.5.31>

IIBC 2018-5-5

MQTT 기반 IoT 시스템을 위한 대시보드 저작 및 뷰어의 설계 및 구현

Design and Implementation of Dashboard Author and Viewer for IoT Systems based on MQTT

황기태*

Kitae Hwang*

요약 많은 센서 장치로 연결된 IoT 시스템을 구축하면 센서들의 값을 모니터링하는 대시보드를 보통 제작하게 되며, 대시보드 소프트웨어의 개발 및 유지에는 많은 시간과 비용이 들어간다. 본 논문은 대시보드를 빠르고 쉽게 만들 수 있는 저작 도구 YellowPeach를 설계하고 구현한 내용을 소개한다. YellowPeach를 이용하여 사용자는 센서 값을 출력하기 위해 미리 준비된 다양한 차트들로 대시보드를 꾸미며, 완성된 대시보드는 JSON 파일로 저장된다. 또한 본 연구에서는 웹 브라우저에서 대시보드를 출력할 수 있는 웹 애플리케이션을 제작하였다. 대시보드 뷰어 웹 애플리케이션은 MQTT 기반의 IoT 시스템을 연결하여 센서로부터 실시간으로 값을 받아 화면에 출력한다.

Abstract When an IoT system is constructed, a dashboard is also developed to monitor the values of sensors in the IoT System. Much time and money, however, are spent in developing the dashboard software. This paper presents design and implementation details of a dashboard maker called as YellowPeach, which helps to create a dashboard easily and fast without programming. Using YellowPeach, the user can author a dashboard by putting predefined charts to display values of sensors in it, and store the dashboard as a JSON file. Also an web application has been developed to run the dashboard on the web browser. The dashboard viewer web application connects on MQTT-based IoT system, receives values from the sensors, and displays the values on charts in real-time.

Key Words : IoT, Dashboard, MQTT

1. 서론

사물인터넷(Internet of Things; IoT)이란 사물 혹은 인간이 네트워크를 통해 긴밀하게 상호 작용할 수 있도록 연결하는 기술을 뜻한다. 사물인터넷 기술은 스마트 팩토리, 스마트 가드닝, 스마트 홈 등 산업 전반에 걸쳐 도입되고 있다. 사물인터넷에 연결되는 장치들은 대형 컴퓨터나 PC에서부터 초소형의 센서에 이르기까지 다양

하다^[1,2].

이런 다양한 분야에서 사물인터넷을 구축하면 사물인터넷에 연결된 장치들의 상태를 모니터링하고 장치들에게 명령을 내리는 대시보드(dashboard)가 필연적으로 요구되며, 대시보드를 작성하는 데는 많은 시간이 소요된다.

본 논문에서는 코딩 없이 대시보드를 제작할 수 있는

*중신회원, 한성대학교 컴퓨터공학부(교신저자)
접수일자 : 2018년 8월 1일, 수정완료 : 2018년 9월 1일
게재확정일자 : 2018년 10월 5일

Received: 1 August, 2018 / Revised: 1 September, 2018 /
Accepted: 5 October, 2018
*Corresponding Author: calafk@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

소프트웨어 YellowPeach와 웹 브라우저를 통해 대시보드를 출력하는 웹 애플리케이션 뷰어를 설계하고 구현한 내용을 소개한다. IoT 시스템은 센서나 장치들이 데이터를 주고받을 수 있는 메시지 브로커가 필요한데, MQTT, HTTP REST, AMQP, STOMP 등의 여러 프로토콜이 있다^[3]. 본 연구에서 대시보드 저작 도구는 IoT 시스템이 채용하는 메시지 프로토콜에 종속되지 않지만, 대시보드 뷰어는 IoT 시스템 내의 센서들로부터 값을 직접 받기 위해 IoT 시스템이 채용하고 있는 메시지 프로토콜을 특정할 필요가 있다. 그러므로 본 연구에서는 현재 IoT 시스템에서 센서들과 컴퓨터들 사이에 메시지를 주고받는데 가장 많이 이용되는 MQTT 기반의 IoT 시스템을 대상으로 한다.

본 연구에서는 스마트팜, 스마트홈, 감시 카메라, 기타 센서 장치들로 구성된 실제 IoT 시스템을 구성하고, YellowPeach와 대시보드 뷰어의 활용성을 검증하였다.

2장에서 본 논문의 연구 배경을 설명하고, 3장에서 전체 시스템의 설계와 구현에 설명하고 4장에서는 YellowPeach와 대시보드 뷰어 웹 애플리케이션의 동작을 검증하고 5장에서 결론을 맺는다.

II. 연구 배경

1. 사물인터넷과 MQTT

스마트 팩토리, 스마트 가드닝 등 현재 사물인터넷 시스템은 전형적으로 인터넷이나 로컬 네트워크에 많은 센서들과 사용자들이 연결되고 이들 사이에는 메시지를 주고받는 방식으로 구축된다. MQTT는 사물인터넷 시스템에서 센서들과 사용자 컴퓨터들 사이에 데이터를 주고받는 표준 프로토콜로 많이 사용된다^[4]. MQTT는 주고받는 데이터를 메시지라고 부르며, 메시지를 기다리는 구독자(subscriber)와 메시지를 발송하는 발행자(publisher)로 구분 짓는 publish-subscribe 모델로 작동한다. 메시지 구독과 발행은 토픽(topic)을 기준으로 한다. 구독자와 발행자 사이에 메시지 전송을 중계하는 MQTT 브로커를 두고, 구독자는 여러 메시지 중에서 받고자 하는 메시지의 토픽(topic)을 MQTT 브로커에 등록하면, 발행자가 메시지를 토픽과 함께 MQTT 브로커에 전송할 때 MQTT 브로커는 등록된 가입자들 중 토픽을 기다리는 구독자들에게만 메시지를 전송한다. 즉 토픽을

기반으로 구독자와 발행자 사이에 메시지가 전송된다.

그림 1은 메시지 발행자인 센서들과 메시지 구독자인 컴퓨터 장치들이 연결된 전형적인 MQTT 기반의 응용 시스템을 보여준다.

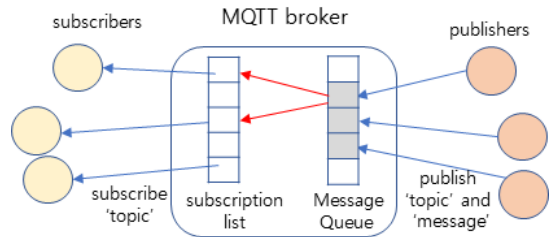


그림 1 MQTT 기반 IoT 시스템

Fig. 1. MQTT-based IoT System

2. 대시보드

대시보드는 차트, 표 등의 다양한 컴포넌트로 데이터를 시각적으로 보여주는 소프트웨어이다. 대시보드는 빅 데이터를 시각화하거나 사물인터넷의 사물로부터 발생하는 데이터를 시각화하기 위해 사용되고 있다.

대시보드를 만들기 위해서는 차트, 표 등의 시각적 컴포넌트들을 함께 개발하여야 하는 어려움이 있다. 한편 Echarts, Tau Charts, ChartJS, amcharts 등 웹 페이지에 다양한 차트를 그릴 수 있는 여러 자바스크립트 라이브러리들이 있다^[5]. 본 연구에서는 비교적 간단하고 다양한 기능의 ChartJS^[6]를 이용한다.

ChartJS 라이브러리를 이용한다고 하더라도 대시보드를 소프트웨어로 직접 개발하는 것은 많은 시간과 노력이 소요된다. 또한 사물인터넷 시스템의 경우 구성된 하드웨어 시스템이 설치 혹은 개발된 후에야 대시보드를 만들고 테스트할 수 있기 때문에, 대시보드 개발은 번거로운 과정이다.

본 논문에서 개발된 YellowPeach는 구축하고자 하는 사물인터넷 시스템의 토픽(topic) 설계 정보만 있으면 토픽을 하나의 차트로 나타내어 토픽에 연결된 센서 값을 출력할 수 있는 대시보드를 만든다. 즉, 사물인터넷 시스템이 개발되기 전에도 대시 보드를 빠르고 쉽게 저장할 수 있다.

III. 시스템 설계

1. 대시보드 저작 및 뷰어 시스템 구성

대시보드 저작 도구 YellowPeach와 뷰어가 작동하는 전체 시스템은 그림 2와 같이 크게 세 부분으로 구성된다.

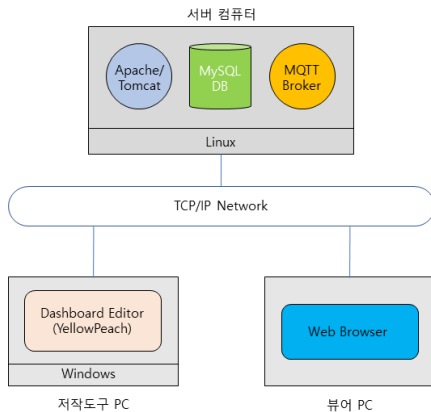


그림 2. 대시보드 저작 및 뷰어 시스템
Fig. 2. System View

첫째는 MQTT 브로커, 웹 서버, DBMS가 작동하는 서버 컴퓨터이다. MQTT 브로커는 MQTT 프로토콜이 구현된 어떤 것도 가능하지만, 본 연구에서는 eclipse에서 개발한 mosquitto^[7]를 이용하였다. 실제 작동하는 IoT 시스템은 그림 2에 센서들이 덧붙여지고, 이들이 MQTT 브로커에 데이터를 보내게(publish) 될 것이다. 또한 대시보드 뷰어는 MQTT 브로커에 토픽을 등록(subscribe)하여, MQTT 브로커로부터 센서 값을 받게 된다.

서버 컴퓨터에는 아파치/톰캣 웹 서버가 설치되고 본 연구에서 개발된 JSP 응용프로그램들이 실행되어 MySQL 데이터베이스에 저장된 대시보드 파일 리스트와 대시보드 파일을 관리한다. 웹 브라우저로부터 요청을 받으면 대시보드 파일 리스트나 대시보드 파일을 전송한다.

서버에 설치된 마지막 요소는 MySQL 데이터베이스이다. 저작된 대시보드는 JSON 파일로 저장되며, 이들은 MySQL 데이터베이스에 저장된다. 뿐만 아니라 MySQL 데이터베이스는 MQTT 브로커를 이용하여 IoT 센서들이 주고받는 각 토픽에 대해, 토픽마다 DB 테이블을 만들고 센서 값들을 모두 저장한다.

둘째는 본 연구에서 YellowPeach로 이름붙인 대시보드 저작 도구이다. YellowPeach는 별도의 PC에서 자동으로 구현하였다. JDBC를 이용하여 MySQL 데이터베이스에 저장된 토픽 리스트를 가지고 와서, 사용자에게 보여 주어 대시보드 상에 차트를 삽입할 수 있도록 한다. 그리고 저작된 대시보드는 JSON^[8] 파일로 저장한 뒤 MySQL 데이터베이스에 저장한다.

셋째는 대시보드 뷰어이다. 대시보드 뷰어는 웹 브라우저가 설치된 어느 컴퓨터에서나 작동시킬 수 있도록, HTML5 웹 애플리케이션으로 설계하였다. 사용자는 HTML5 표준을 따르는 아무 웹 브라우저를 이용하여 서버에 접속하면 된다. 웹 브라우저는 JSP 응용프로그램을 통해 웹 페이지와 자바스크립트 코드, 그리고 JSON 파일을 넘겨받고, JSON 파일을 파싱하여 화면에 대시보드를 그리고, 대시보드 속에 담긴 토픽을 MQTT 브로커에 subscribe하여 센서들로부터 값을 받아 실시간으로 차트를 그린다.

2. 대시보드 저작

YellowPeach의 구조와 동작 과정에 대한 설계는 그림 3과 같다. YellowPeach은 JDBC 드라이버 외 3개의 모듈로 구성된다.

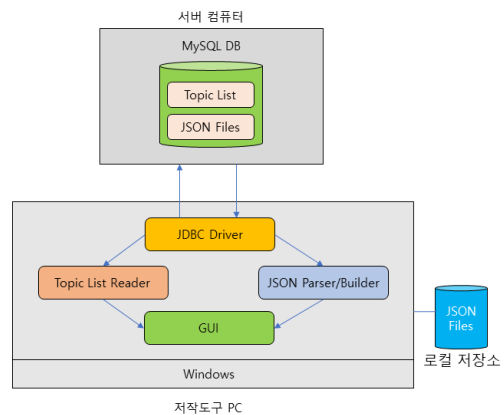


그림 3. YellowPeach의 구조
Fig. 3. Structure of YellowPeach

YellowPeach의 Topic List Reader 모듈은 JDBC 드라이버를 호출하여 서버 컴퓨터의 MySQL에 저장된 토픽 리스트를 읽어 오며, 토픽 리스트는 필요할 때마다 GUI 모듈을 통해 화면에 출력된다. 각 센서마다 혹은 동일한 의미를 지닌 센서들이 동일한 토픽으로 데이터를 전송한

다. GUI 모듈은 JavaFX 라이브러리를 이용하여 구현하였다.

GUI 모듈은 사용자가 대시보드를 제작하도록 돕는다. 사용자에게 토픽 리스트를 알려주어, 사용자가 토픽의 데이터를 어떤 차트를 이용하여 그릴 것인지 판단하게 한다.

JSON Builder 모듈은 사용자의 대시보드 만들기가 끝나면 대시보드 정보를 JSON 파일로 저장한다. JSON 파일에는 각 토픽의 값들을 어떤 차트로 어느 위치에 그릴 것인지 정보가 들어 있다. JSON 파일은 역시 JDBC 드라이버를 이용하여 MySQL 데이터베이스에 바로 저장된다.

JSON (JavaScript Object Notation)은 경량 데이터 교환 형식이다^[8]. 이 형식은 사람이 읽고 쓰기에 용이하며, 기계가 분석하고 생성함에도 용이하여 시스템 간의 데이터를 주고 받는 형식으로 많이 사용되므로 본 논문에서도 이 형식을 사용한다.

대시보드를 수정할 때, MySQL 서버로부터 JSON 파일을 읽고 JSON Parser 모듈을 이용하여 파싱하고, 그 결과를 GUI 모듈이 화면에 출력한다.

3. JSON 파일 구조

YellowPeach에서 제공하는 차트의 종류는 bar, line, value, video, camera, gauge, text, donut, image, pie 등 총 10개이다. 이 차트들은 ChartJS가 제공하는 차트 종류와 동일하다. 각 차트는 이름, 출력 위치, 차트에 출력될 토픽 등으로 구성되며 사례는 그림 4와 같다.

4. 대시보드 뷰어와 실시간 차트 그리기

대시보드 뷰어는 JSP 프로그램과 웹 브라우저에서 실행되는 자바스크립트로 구현하였고 실행되는 과정은 그림 5와 같다.

```
{
  "content": [{
    "0": {
      "row": "1",
      "chart": "histogram",
      "control": "HSFarm/mirae/ctrtemp",
      "chartName": "nn",
      "firstName": "",
      "firstTopic": "HSFarm/mirae/temp",
      "secondName": "null",
      "secondTopic": "null"
    },
    "1": {
      "row": "1",
      "chart": "streamming",
      "control": "false",
      "chartName": "streamming tt",
      "firstName": "null",
      "url": "https://www.naver.com/",
      "secondName": "null",
      "secondTopic": "null"
    },
    "2": {
      "row": "2",
      "chart": "value",
      "control": "HSFarm/mirae/ctrlrd",
      "chartName": "va",
      "firstName": "",
      "firstTopic": "YPSmartHome/room1/humid",
      "secondName": "null",
      "secondTopic": "null"
    },
    "3": {
      "row": "2",
      "chart": "image",
      "control": "HSFarm/mirae/IMG",
      "chartName": "va",
      "firstName": "",
      "url": "https://www.naver.com/",
      "secondName": "null",
      "secondTopic": "null"
    }
  ]},
  "dashboardName": "DEF"
}
```

그림 4. 대시보드 정보를 담은 JSON 샘플

Fig. 4. JSON sample file representing a dashboard

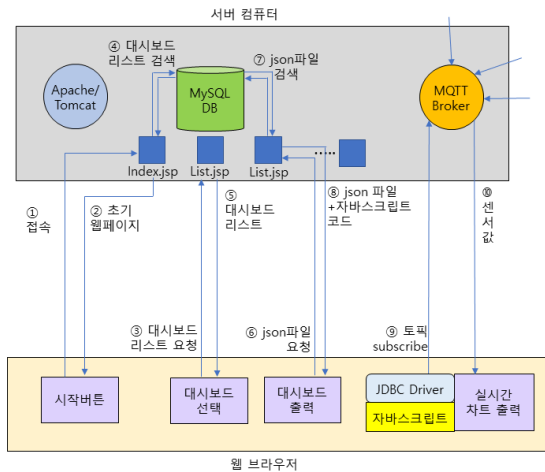


그림 5. 대시보드 뷰어의 실행 과정
Fig. 5. Running process of Dashboard viewer

JSP 프로그램들은 웹브라우저가 웹 서버로부터 JSON 파일을 다운받는 과정동안 실행되며, 과정8에서 JSON 파일과 함께 전달된 자바스크립트 코드가 JSON 파일을 파싱하여 대시보드를 웹 페이지의 캔버스 영역에 그린다. 대시보드를 그릴 때 ChartJS를 이용하였다.

자바스크립트 코드가 대시보드를 그리고 나면, JSON 파일 안에 나열된 모든 토픽을 MQTT 브로커에게 subscribe 한다. 그 후 MQTT 브로커로부터 실시간으로 날아오는 센서 값을 해당하는 차트에 실시간으로 출력 한다.

IV. 테스트 및 실행

1. 테스트 시스템 구성

본 연구에서 개발한 대시보드 저작 도구인 Yellow Peach와 뷰어의 작동을 검증하기 위해 본 연구팀은 그림 6과 같은 테스트 IoT 시스템을 구성하였다.

테스트 IoT 시스템은 그림 2의 기반 시스템에 스마트팜(라즈베리파이3 1개), 스마트홈(라즈베리파이3 1개), 그리고 감시 카메라 2개(라즈베리파이3 1개, 라즈베리파이ZeroW 1개), 독립 센서(라즈베리파이ZeroW 1개)를 구성하였다. 각 임베디드 컴퓨터 장치에는 여러 개의 센서와 카메라 등을 구축하였다. 스마트팜과 스마트홈 IoT 시스템의 실제 모습은 그림 7과 같으며, 스마트팜은 이전 연구^[9]를 통해 완성된 IoT 시스템을 그대로 사용하였다.

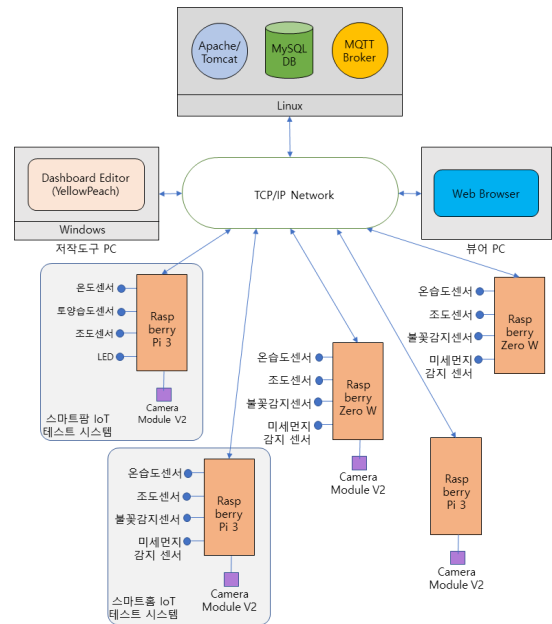


그림 6. 테스트 IoT 시스템과 대시보드 저작/뷰어
Fig. 6. Test IoT system and YellowPeach/dashboard viewer

2. 시스템 테스트

그림 7은 전체 시스템이 실제 작동하는 모습이다. 그림에서 왼쪽 하단은 대시보드 저작 도구인 YellowPeach가 실행되는 화면이고, 그 오른쪽은 대시보드가 출력되는 뷰어 화면이다.



그림 7. 테스트 IoT 시스템을 포함한 전체 시스템
Fig. 7. Total System including the test IoT system

그림 8은 YellowPeach로 대시보드를 저작하는 화면이며, 그림 9는 브라우저 상에서 대시보드가 작동되는 화면이다.

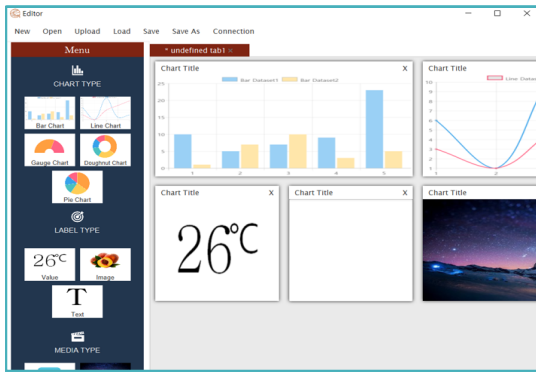


그림 8. YellowPeach의 대시보드 저작 과정
Fig. 8. Authoring a dashboard using YellowPeach

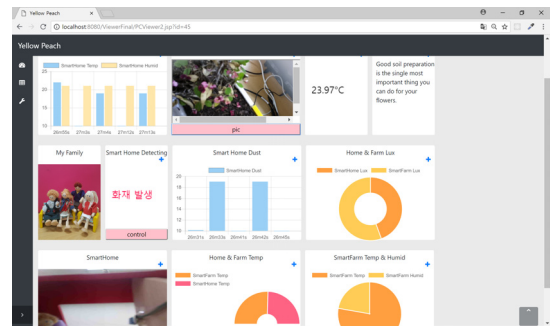


그림 11. 대시보드에 "화재 발생" 메시지 출력
Fig. 11. Display "Flame detection" message on the dash board

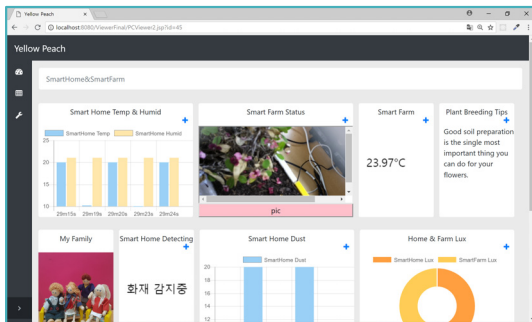


그림 9. 웹브라우저로 대시보드 출력
Fig. 9. Display a dashboard on web browser

그림 10은 스마트홈에서 불꽃 감지 센서에 라이터를 켜서 불꽃을 일으키는 모습을 보여준다. 이때 불꽃 감지 센서는 바로 “화재 발생” 메시지를 전송하며, 그림 11에서 이 메시지가 실시간으로 대시보드에 전송되어, “화재 발생”이 출력되는 사례를 보여준다.



그림 10. 라이터로 불꽃 감지 센서 작동
Fig. 10. Making Flame detection sensor work using a lighter

V. 결론

본 논문에서는 MQTT 기반 IoT 시스템의 센서 값을 실시간으로 보여주는 대시보드를 코딩 없이 간단히 만들 수 있는 대시보드 저작 도구 YellowPeach와 대시 보드를 웹브라우저에 출력하는 뷰어 웹 애플리케이션을 개발한 내용을 소개하였다. 또한 5개의 라즈베리파이 컴퓨터에 총 16개의 센서와 4개의 카메라를 가진 MQTT 기반의 테스트 IoT 시스템을 개발하고, 이 시스템을 대상으로 본 논문에서 개발한 시스템을 검증하였다.

본 논문에서 개발한 YellowPeach와 뷰어를 이용하면, 대상 IoT 시스템을 모니터링할 수 있는 대시보드를 만드는 데 최대 몇 시간이면 가능하며, 대시보드의 수정 및 관리가 매우 쉽다.

References

- [1] SeoHyung Kim, IoT : Internet of Things Technology, IEIE Magazine, Vol. 43, No.3, pp.64-71, 2016
- [2] Dongha Shim, Ji-Hoon Yang, Jeungki Son, Seung-Han Han, Hyounmin Lee, Smart Parking Guidance System based on IoT Car-stoppers, The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 17, No. 3, pp.137-143, Jun. 2017
- [3] Jasenka Dizdarevic, Francisco Carpio, Admela Jukana, Xavi Masip-bruin, A Survey of

Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration, ACM Computing Surveys, Vol. 1, No. 1, April 2018.

- [4] <https://en.wikipedia.org/wiki/MQTT>
- [5] <https://hackernoon.com/9-best-javascript-charting-libraries-46e7f4dc34e6>
- [6] <https://www.chartjs.org/>
- [7] <https://mosquitto.org/>
- [8] <https://www.json.org/json-ko.html>
- [9] Kitae Hwang, Heyjin Park, Jisu Kim, Taeyun Lee, Inhwan Jung, An Implementation of Smart Gardening using Raspberry pi and MQTT, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 18, No. 1, pp.151-157, 2018

저자 소개

황 기 태(정회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 1994년 III 현재 한성대학교 컴퓨터공학과 교수
- 경력 : University of Florida 방문 교수

<주관심분야 : 모바일 시스템, IoT>

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음.