

<https://doi.org/10.7236/IIBC.2018.18.4.211>

IIBC 2018-4-29

MQTT와 Node-RED를 이용한 설비 모니터링 시스템의 구현

Implementation of factory monitoring system using MQTT and Node-RED

오세춘*, 김태형**, 김영곤***

Se-Chun Oh*, Tae-Hyung Kim**, Young-Gon Kim***

요약 근래에는 IoT 및 스마트팩토리 산업분야의 확산으로 인하여 IIoT와 관련한 다양한 기술들이 지속적으로 소개되고 있다. 본 논문에서는 이러한 다양한 기술들을 이용한 공장 설비 감시용 양방향 무선 네트워크 시스템의 구축을 제안한다. 본 논문에서 사용되는 주요 기술들은 다양한 현장의 설비상태를 모니터링하기 위한 초소형 센서 노드의 설계 기술, 개별 센서 노드에서 처리된 다양한 정보를 무선 시스템을 이용해 서버로 전송하고 또한 서버에서 새롭게 설정된 각종 센서의 임계치들을 개별 센서 노드로 무선 전송하기 위한 Wi-Fi 기반의 MQTT 기술, 수집된 데이터를 저장하고 유무선으로 사용자에게 손쉽게 표시할 수 있는 Node-RED 기반의 설계 기술 등이다. 아울러 개별 센서 노드에서의 이상상태의 판단시에 이를 상황실로 알리고 상황실의 무선조정에 의해 현장의 화면영상을 상황실에서 확인할 수 있는 무선 양방향 카메라 시스템도 함께 구현하였다.

Abstract Recently, various technologies related to IIoT are introduced continuously due to the spread of IoT and smart factory industries. This paper proposes the construction of a two-way wireless network system for monitoring plant equipment using these various technologies. The main technologies used in this thesis are design techniques for micro sensor nodes to monitor facility conditions at various sites, MQTT technology for wireless communication between local server and sensor nodes and Node-RED based design technologies, which store data collected and can be easily presented to users via wired and wireless wires. In addition, a wireless two-way camera system was also implemented in which the screen images of the site can be viewed in the situation room according to the instructions of the situation room when determining abnormal conditions.

Key Words : MQTT, Node-RED, NodeMCU, Monitoring system

1. 서론

근래에는 IoT(Internet of Things)와 스마트팩토리(SmartFactory) 산업의 대두로 특히 IIoT(Industrial IoT)

분야와 관련된 다양한 형태의 기술들이 각종 산업분야에 파급을 미치고 있는 추세이다. 기존의 공장 내부 설비 모니터링 및 제어 기술들은 대부분 유선 네트워크를 기반으로 구축되어 있었으나 근래에는 유선과 무선을 포함한

*정희원, 한국산업기술대학교 컴퓨터공학과(교신저자)

**준희원, 한국산업기술대학교 컴퓨터공학과

***정희원, 한국산업기술대학교 컴퓨터공학과

접수일자 2018년 6월 25일, 수정완료 2018년 7월 25일

게재확정일자 2018년 8월 10일

Received: 25 June, 2018 / Revised: 25 July, 2018 /

Accepted: 10 August, 2018

*Corresponding Author: sc.ohl@kpu.ac.kr

Dept. of Computer Engineering, Korea Polytechnic University, Korea

다양한 형태의 혼합형태의 네트워크가 사용되고 있는 추세이며[1] 이를 위한 무선기술로는 Wi-Fi, Bluetooth, UWB(Ultra Wide Band), ZigBee 등의 다양한 무선 네트워크들이 검토되고 있다[2]. 본 논문에서는 고정된 넓은 범위 내에서 사용되며 상대적으로 빠른 데이터 전송속도를 유지할 수 있는 Wi-Fi와 이를 기반으로 동작되는 MQTT09(Message Queuing Telemetry Transport)[3]를 사용한 양방향 공장 설비 모니터링 시스템을 설계하고 구현하였다.

개별설비의 상태를 측정하기 위해서는 Wi-Fi 기능을 내장하고 있는 Espressif사의 ESP8266 MCU(Micro Controller Unit)를 사용하였고 여기에서 측정된 각종 센서 데이터를 MQTT 프로토콜을 이용해 무선으로 로컬 서버로 전송을 하고, 로컬 서버에서는 Node-RED[4] 미들웨어 소프트웨어를 이용해 센서 노드에서 수집된 정보를 실시간으로 상황실 컴퓨터나 기타 휴대용 단말기에 유무선으로 시각화하는 기능을 담당하도록 설계했다. 또한 시스템 관리자가 Node-RED 인터페이스를 이용해 각 센서 노드의 개별 센서들의 이상여부 판단용 센서 임계치를 상황에 맞게 새롭게 설정할 수 있으며, 필요시에는 실시간 현장의 영상도 원격으로 확인할 수 있는 양방향 무선통신 기능도 추가적으로 구현하였다.

II. 관련 연구

1. MQTT

MQTT[3]는 사물 인터넷(IoT)과 같이 제한된 대역폭을 갖는 통신 환경에 적합하도록 개발된 푸시 기술(push technology) 기반의 메시지 전송 프로토콜으로써 적은 코드로도 구현이 가능한 특징이 있다. 기존의 푸시 기술은 클라이언트/서버의 구조로 되어 있으나 MQTT는 메시징(messaging) 개념을 이용한 푸시 기술을 구현하고 있다. 즉 MQTT 브로커(broker)를 중심으로 정보의 생성 및 송신을 담당하는 publisher와 정보를 수신만 하게되는 subscriber로 구성이 되며, publisher에서 생성된 모든 메시지는 MQTT 브로커에 의해 취합이 되고 MQTT 브로커는 특정 메시지의 수신을 희망하는 모든 subscriber들에게 해당 메시지를 전송해주는 구조이다.

모든 메시지들은 토픽(topic)이라는 형태의 메시지 정보와 실제 데이터 값(payload)으로 구성이 된다. 토픽은

‘/’ 기호를 사용하여 계층구조로 형성할 수 있다. 그림 1은 MQTT의 동작 구조이며 좌측의 publisher 노드들에서 생성된 ‘sensor/temp’ 메시지 및 ‘sensor/humi’ 메시지는 MQTT 브로커로 모아지며 브로커에서는 우측의 subscriber 노드들 중에서 특정 메시지(토픽+데이터)의 수신을 원하는 subscriber 노드로 해당 메시지를 전달하게 된다. 그림 1의 우측 하단 노드의 경우는 publisher와 subscriber의 기능을 하나의 노드에서 모두 가지고 있는 경우이며 우측 상단의 subscriber 노드는 ‘sensor’ 토픽을 수신하는 것으로 등록이 되어 있으므로 모든 센서 메시지 정보(‘sensor/*’)들을 수신한다.

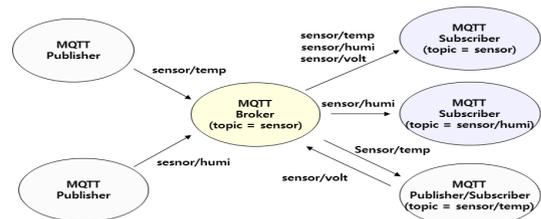


그림 1. MQTT 동작 구조

Fig. 1. Example of MQTT operation

MQTT는 기존의 HTTP 방식과 비교하여 상대적으로 낮은 소비전력과 적은 네트워크 부하를 갖는 구조로 되어 있는데 이러한 장점에 대한 많은 연구[5],[6]들이 진행되고 있으며, 또한 이러한 장점으로 인해 최근에는 다양한 분야에서 MQTT가 적극적으로 적용이 되고 있는 추세이다[7]~[14].

2. Node-RED

Node-RED[4]는 IBM사에서 개발한 IoT 분야의 대표적인 front-end 개발용 미들웨어 툴이다. Node-RED는 흐름 기반(flow-based)의 시각화 소프트웨어이며 그림 2에서 보이듯이 많은 디바이스와 프로토콜이 입출력을 갖는 노드들로 추상화되어 있어서 사용자는 웹을 이용하여 이러한 노드들을 서로 연결하여 주고 필요한 주요 변수들을 설정함으로써 새로운 서비스 또는 데이터 처리 흐름을 쉽게 구성할 수 있다. Node-RED는 Node.js 기반으로 개발이 되어 Node.js가 제공하는 많은 라이브러리를 사용할 수 있으며 또한 새로운 노드를 쉽게 추가할 수도 있다.

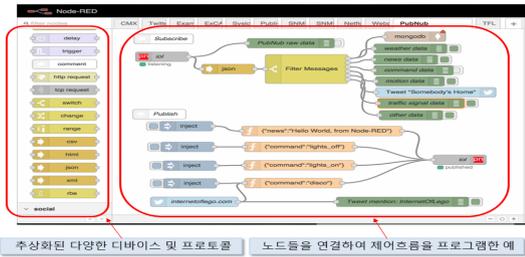


그림 2. Node-RED 프로그래밍 화면
 Fig. 2. Programming screen of Node-RED

Node-RED를 설치하게 되면 설치 중에 특정 IP가 제공되는데 이 IP 주소를 웹 브라우저를 통해 접근하게 되면 그림 2에서 보이는 Node-RED 프로그래밍 화면이 나타난다. 그림 2의 좌측 메뉴에서는 추상화된 다양한 디바이스와 프로토콜들을 볼 수 있는데 사용자는 이를 화면 우측으로 드래그한 후 각각의 입출력을 연결해 주고, 필요시 기본적인 변수 값들을 설정해 주면 간단히 제어 흐름에 대한 프로그래밍이 완료된다.

Node-RED의 또다른 편리한 기능은 대시보드(dashboard) 기능이며 여기에는 다양한 GUI용 도구들을 제공해 주고 있어서 사용자는 이를 이용하여 기존에 설계된 다양한 제어용 흐름들을 쉽게 대시보드용 화면으로 구성할 수 있다. 그림 3은 대시보드의 실제 구현 예이다. 그림 2의 프로그래밍 화면에서 사용하는 다양한 입출력 변수들을 기본으로, 제공되는 다양한 템플릿을 이용해 쉽게 그림 3과 같은 대시보드 GUI 화면을 설계할 수 있다. 설계된 대시보드 GUI 화면은 외부의 어느 곳에서도 지정된 IP 주소를 통해 쉽게 접근할 수 있기 때문에 유무선 기기를 통해 원격으로 접근이 가능하다.



그림 3. Node-RED 대시보드 구현 화면 예제
 Fig. 3. Example screen of dashboard implementation

Node-RED는 라즈베리파이3용 라즈비안(raspbian) 운영체제에서의 기본적인 제공, BeagleBone Black과 인

도우에서도 사용이 가능하고 최근에는 아두이노 보드 등을 포함한 다양한 플랫폼에서 지원뿐만 아니라 편의성 및 확장성으로 인해 많은 산업분야에서 관련 연구와 적용이 적극적으로 진행되고 있다[15],[16].

3. NodeMCU

NodeMCU[17]는 IoT용 오픈 소스 펌웨어(firmware)와 개발 키트(development kit)를 의미한다. NodeMCU는 Espressif Systems사의 ESP8266 32비트 RISC MCU를 사용하는데 이 MCU에는 TCP/IP 스택을 지원하는 IEEE802.11 b/g/n Wi-Fi 모듈, SRAM 모듈, PWM 모듈, GPIO(General Purpose Input/Output) 모듈, 10비트 해상도의 1채널 ADC 모듈, SPI, I²C 및 UART 등의 다양한 직렬통신 모듈들을 내장하고 있다.

NodeMCU의 기본적인 프로그래밍 언어로는 Lua script를 사용하나 최근에는 아두이노(arduino) 보드용 IDE(Integrated Development Environment)를 그대로 사용하고 아두이노 보드용의 다양한 실드(shield)를 바로 활용할 수 있는 방법도 제공되고 있다. 따라서 이러한 NodeMCU의 기본구조에 IoT 개발에 필요한 IDE용 USB 인터페이스 및 프로그램용 플래시메모리를 추가한 다양한 IoT용 개발 보드들이 Adafruit, WeMos 등의 많은 업체들에서 다양한 규모의 솔루션들로 제공되고 있다. 그림 4는 본 논문에서 사용하는 WeMos사의 D1 Mini 제품의 핀 구성도를 보여주고 있다. 이 제품에는 NodeMCU 기본 구조에 32Mbit 크기의 프로그램용 플래시메모리와 개발용 USB 인터페이스, 전원부가 추가로 내장되어 있다.

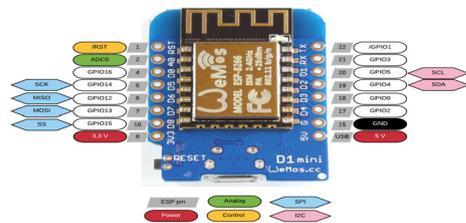


그림 4. WeMos D1 Mini 제품의 핀 배치도[17]
 Fig. 4. Pin diagram of WeMos D1 Mini

III. 3장 시스템의 설계

1. 전체 시스템의 구성

그림 5는 전체 시스템의 구성도이다.

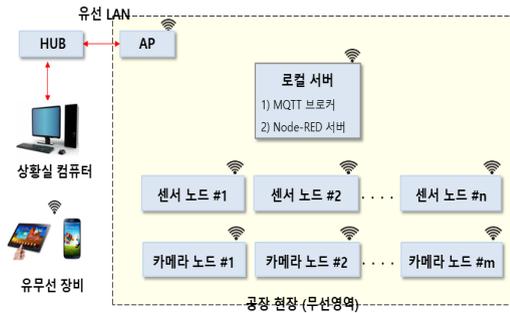


그림 5. 전체 시스템 구성도
Fig. 5. System configuration diagram

그림 5에서 보이듯이 전체 시스템은 각 설비에 장착되는 개별 센서 노드들과 공장 현장에 설치되어 본 시스템의 핵심기능을 하는 로컬 서버, 로컬 서버의 모든 데이터를 확인하고 설정할 수 있는 상황실 컴퓨터를 비롯한 다양한 통제장비로 구성이 되며, 마지막으로 특정 센서 노드의 이상발생시 통제장비의 지시에 따라 실시간으로 현장의 영상을 수집해 전송하는 카메라 노드들로 구성된다.

2. 센서 노드 (Sensor Node)

센서 노드의 역할은 공장 내의 개별 설비에 장착이 되어 지정된 측정주기마다 다양한 센서들의 값을 읽고 처리하여 별도로 지정된 전송주기마다 Wi-Fi 기반의 MQTT를 이용하여 로컬 서버를 통해 상황실로 전송하는 역할을 한다. 또한 상황실에서 필요에 따라 변경되어 제시되는 각종 노드별 센서들의 개별 임계치를 기억하여 이를 센서 측정치와 비교하여 정상범위를 초과할 경우 비주기적인 경고 메시지를 MQTT를 통해 서버와 상황실로 보내주는 역할을 한다. 상황실 컴퓨터에서는 개별 센서 노드별로 초기 또는 필요시 마다 센서 읽기 주기, 센서 데이터 전송주기, 개별 센서들의 임계치 등을 MQTT를 통해 실시간으로 변경을 하여 무선 네트워크의 부하 및 센서 노드의 전원소모를 제어한다. 따라서 센서 노드에는 publisher, subscriber 기능이 설치된다.

센서 노드는 그림 6에서 보이듯이 IEEE 802.11n 모드를 지원하는 ESP8266 MCU를 기반으로 설계된 WeMos D1 Mini 보드를 기본으로 구현하였으며 아두이노 IDE 기반의 프로그래밍 환경을 구축하여 사용하였다. 그림 6의 좌측 그림에서 가장 왼쪽에 있는 부품이 WeMos D1

Mini 보드이며 우측 그림이 구현된 센서 노드의 프로토타입 사진이다.

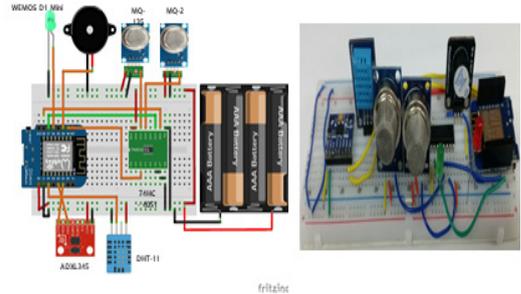


그림 6. 센서 노드의 구현
Fig. 6. Implementation of sensor node

센서 노드에서는 그림 6과 같이 온도, 습도, 진동, 유해가스, 가연성 가스 센서 등이 사용되고 있으며 그 이외에 배터리 전원을 사용할 수 있기 때문에 배터리 전압을 측정하는 기능도 구현되어 있다. ESP8266 MCU는 1 채널의 10비트 AD 변환기만을 내장하고 있기 때문에 복수의 아날로그 센서들 및 전원전압을 측정하기 위해서는 아날로그 멀티플렉서 IC를 추가로 사용하여 다채널의 아날로그 센서 입력을 측정할 수 있게 설계하였다.

센서 노드는 효율적인 전원관리를 위해 슬립(sleep) 모드로 대기하다가 상황실에서 미리 지정한 읽기주기마다 스스로 깨어나 각종 센서 값을 읽고 이 값들을 미리 설정된 임계치와 비교하여 설정된 임계치를 벗어난 비정상 범위일 경우 MQTT를 통해 로컬 서버로 센서 이상 정보를 실시간으로 전송한다. 현재 읽은 센서 값이 정상 범위일 경우이며 미리 지정된 전송주기에 시간이 도달하지 않은 경우에는 다시 슬립모드로 진입하여 전원 사용을 최소화 한다.

센서 노드들은 배터리 전원을 사용할 수 있게 설계되어 이동 및 재설치가 간편하다는 장점이 있으며, 별도의 5V 외부 전원을 공급할 경우에는 슬립모드를 사용하지 않고 최대의 성능을 발휘하도록 설계되어 있다.

3. 로컬 서버 (Local Server)

로컬 서버는 공장 내부의 현장에 설치가 되며 별도의 라즈베리파이3 제품을 이용하여 구성하였으며 MQTT 브로커와 센서 노드용 Node-RED 서버 소프트웨어가 설치된다. 또한 수집된 센서 정보들을 저장하는 데이터베이스 기능도 Node-RED와 연동되어 구축되어 있다.

로컬 서버의 주된 기능은 센서 노드와 무선으로 연결되어지는 MQTT 브로커의 역할을 하며, 센서 노드용 Node-RED 서버 역할을 하여 측정된 센서의 주기적 상태 및 비주기적 긴급 경고 정보를 자체 데이터베이스에 저장하고 이를 Node-RED 대시보드 기능을 이용해 쉽게 다양한 사용자 인터페이스방식으로 표현하여 상황실 컴퓨터나 각종 유무선 단말기에서 편리하게 접근할 수 있게 한다.

4. MQTT 메시지의 정의

그림 7은 센서 노드와 로컬 서버 사이에 사용되고 있는 MQTT 메시지(topic)의 정의표이다. 메시지 그룹 1은 상황실에서 해당 센서 노드의 스케줄링과 관련한 시간 정보 파라미터를 설정하는 메시지 그룹이며, 메시지 그룹 2는 주기적인 센서 측정치의 전송을 위한 메시지들이며 메시지 그룹 3은 센서 노드에서 사용되어 지는 각종 센서들의 임계치를 설정하기 위한 메시지 형태이고 메시지 그룹 4는 측정치가 임계치를 벗어날 경우 센서 노드에서 상황실로 긴급 상황을 알리기 위해 사용되는 비주기적인 메시지 그룹이다.

```
// MQTT 메시지 그룹 1: 센서 노드에서의 센서 측정주기, 센서 값 전송주기 설정 (센서 노드에서 수신)
#define topic_READ_set "kpu5/node_0/SET/READ" // 센서 읽기 주기 설정(초단위)
#define topic_SEND_set "kpu5/node_0/SET/SEND" // 센서 측정값 전송주기 설정(초단위)

// MQTT 메시지 그룹 2: 센서 노드에서 센서 측정값 전송 메시지 (센서 노드에서 송신)
#define topic_TEMP_set "kpu5/node_0/TEMP" // 온도 측정치 전송
#define topic_HUMI_set "kpu5/node_0/HUMI" // 습도 측정치 전송
#define topic_LPG_set "kpu5/node_0/LPG" // LPG 가스 측정치 전송
#define topic_SMOKE_set "kpu5/node_0/SMOKE" // Smoke 측정치 전송
#define topic_CO2_set "kpu5/node_0/CO2" // CO2 측정치 전송
#define topic_ACC_X_set "kpu5/node_0/ACC_X" // X축 가속도 측정치 전송
#define topic_ACC_Y_set "kpu5/node_0/ACC_Y" // Y축 가속도 측정치 전송
#define topic_ACC_Z_set "kpu5/node_0/ACC_Z" // Z축 가속도 측정치 전송
#define topic_BAT_set "kpu5/node_0/SET/BAT" // 동작전압 측정치 전송

// MQTT 메시지 그룹 3: 새로운 센서 측정값 임계치의 설정 메시지 (센서 노드에서 수신)
#define topic_TEMP_set "kpu5/node_0/SET/TEMP" // 온도 임계치의 설정
#define topic_HUMI_set "kpu5/node_0/SET/HUMI" // 습도 임계치의 설정
#define topic_LPG_set "kpu5/node_0/SET/LPG" // LPG 가스 임계치의 설정
#define topic_SMOKE_set "kpu5/node_0/SET/SMOKE" // Smoke 임계치의 설정
#define topic_CO2_set "kpu5/node_0/SET/CO2" // CO2 가스 임계치의 설정
#define topic_ACC_X_set "kpu5/node_0/SET/ACC_X" // X축 가속도 임계치의 설정
#define topic_ACC_Y_set "kpu5/node_0/SET/ACC_Y" // Y축 가속도 임계치의 설정
#define topic_ACC_Z_set "kpu5/node_0/SET/ACC_Z" // Z축 가속도 임계치의 설정
#define topic_BAT_set "kpu5/node_0/SET/BAT" // 동작전압 임계치의 설정

// MQTT 메시지 그룹 4: 센서 측정값이 임계치를 초과할 경우의 긴급 메시지 (센서 노드에서 송신)
#define topic_TEMP_warn "kpu5/node_0/WARN/TEMP" // 온도 측정치 이상시의 긴급 메시지
#define topic_HUMI_warn "kpu5/node_0/WARN/HUMI" // 습도 측정치 이상시의 긴급 메시지
#define topic_LPG_warn "kpu5/node_0/WARN/LPG" // LPG 가스 측정치 이상시의 긴급 메시지
#define topic_SMOKE_warn "kpu5/node_0/WARN/SMOKE" // Smoke 측정치 이상시의 긴급 메시지
#define topic_CO2_warn "kpu5/node_0/WARN/CO2" // CO2 가스 측정치 이상시의 긴급 메시지
#define topic_ACC_X_warn "kpu5/node_0/WARN/ACC_X" // X축 가속도 측정치 이상시의 긴급 메시지
#define topic_ACC_Y_warn "kpu5/node_0/WARN/ACC_Y" // Y축 가속도 측정치 이상시의 긴급 메시지
#define topic_ACC_Z_warn "kpu5/node_0/WARN/ACC_Z" // Z축 가속도 측정치 이상시의 긴급 메시지
#define topic_BAT_warn "kpu5/node_0/WARN/BAT" // 저전압 측정치의 긴급 메시지
```

그림 7. MQTT 메시지(topic) 정의
 Fig. 7. Definition of MQTT message(topic)

5. 카메라 노드 (Camera Node)

카메라 노드는 상황실에서 현장의 상태를 파악할 때

사용된다. 각각의 카메라 노드는 CPU 부하를 감안하여 별도의 라즈베리파이3 보드들로 구성되며 카메라 노드 전용의 Node-RED 서버 프로그램이 설치되어 로컬 서버의 부하를 덜어주는 구조로 설계되었다. 그림 8은 카메라 노드의 주요 구성을 보여주고 있으며 우측에는 카메라 노드 제어를 위한 Node-RED 제어 흐름도를 보여주고 있다.

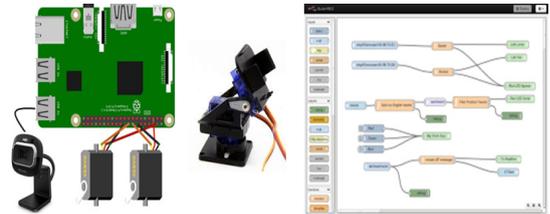


그림 8. 카메라 노드
 Fig. 8. Implementation of camera node

카메라 노드 역시 영상정보의 전송방식은 Wi-Fi를 이용한 무선통신을 사용하며 USB방식의 WebCAM 카메라로 영상을 처리하고, 카메라의 panning과 tilting 기능을 위해 2개의 서보모터가 사용되는데, 상황실의 Node-RED 화면에서 카메라 영상 송출 기능의 선택 및 4 방향에 대한 제어 명령을 내리면 이를 해당 카메라 노드에서 수신하여 실행하게 된다.

IV. 실험 및 결과

그림 9는 본 논문의 모든 기능이 구현된 시스템의 사진이다. 라즈베리파이3로 구현된 로컬 서버와 4개의 ESP8266 센서 노드로 구성되어 있으며 카메라 노드도 별도의 라즈베리파이3 보드에 구현이 되어 있다.

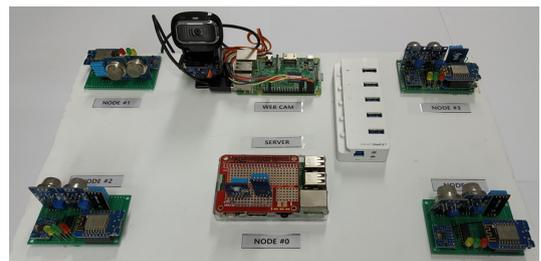


그림 9. 제안된 전체 시스템의 구현
 Fig. 9. Implementation of proposed system

그림 10은 센서 노드와 로컬 서버의 동작 검증과 관련한 시뮬레이션 절차의 일부를 보여주고 있다.

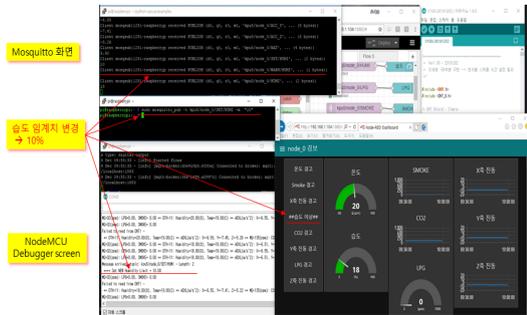


그림 10. 상황실에서의 임계치 임의의 변화에 따른 결과 검증 Fig. 10. Verification of abnormal condition processing

상황실 컴퓨터에서 Node-RED 대시보드를 통해 특정 센서 노드의 습도 임계치를 10%로 변경할 경우, 이 내용이 로컬 서버를 통해 해당 센서 노드로 전송되고 이에 따라 센서 노드에서의 습도 측정치(20%)가 새롭게 설정된 임계치(10%)를 벗어나기 때문에 해당 센서 노드가 MQTT를 이용하여 긴급 습도관련 경고 메시지를 상황실 컴퓨터로 전송하는 절차를 보여주고 있다.

그림 11은 센서 노드와 로컬 서버, 카메라 노드와 로컬 서버 사이의 동작을 위한 Node-RED 제어 흐름도와 설계된 대시보드 화면의 일부를 보여주고 있다. 상단의 우측 그림은 카메라 노드에서 전송된 동영상을 Node-RED 화면에 표시한 사진이다.



그림 11. 다양한 Node-RED 대시보드 화면과 프로그래밍 화면 Fig. 11. Various Node-RED dashboard & programming screen

V. 결론

Wi-Fi, MQTT, Node-RED, NodeMCU, WebCAM, 서버모터 제어 등의 다양한 기술들을 복합적으로 사용하여 IIoT용 공장 설비 무선 모니터링 시스템을 설계하고 구현하였다. 모든 제안된 기능들이 원활히 작동하는 것을 확인하였으며, 특히 일반적인 공장 내부의 환경은 IEEE 802.11n 모드의 Wi-Fi와 MQTT 기술로도 충분히 무선 네트워크를 구축할 수 있다는 것을 확인할 수 있었으며 또한 Node-RED를 활용하여 손쉽게 제어 흐름 및 UI를 구성할 수 있었다.

향후에는 본 논문에서 부족한 부분인 수집된 데이터의 클라우드(clouding)화 기법 및 빅데이터(bigdata) 처리 기술을 적용하여 설비 및 각종 부품들의 이상 상태 및 예상고장 주기 등을 능동적으로 판단할 수 있는 연구와 IIoT의 새로운 표준으로 떠오르고 있는 OPC-UA와 관련된 다양한 센서 및 장비의 인터페이스 기능을 보장하여 단순 모니터링 기능위주의 시스템에서 진정한 스마트팩토리 제어 시스템으로의 발전을 위한 연구가 필요하다고 생각한다.

References

- [1] Thilo Sauter, "The Three Generations of Field-Level Networks—Evolution and Compatibility Issues", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 57, NO. 11, NOVEMBER 2010
DOI: 10.1109/TIE.2010.2062473
- [2] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi", The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON) Nov. 5-8, 2007, Taipei, Taiwan
DOI: 10.1109/IECON.2007.4460126
- [3] <http://mqtt.org/>
- [4] <https://nodered.org/>
- [5] Jose Luis Espinosa-Aranda, Noelia Vallez, Carlos Sanchez-Bueno, "Pulga, a tiny open-source

- MQTT broker for flexible and secure IoT deployments”, 1st Workshop on Security and Privacy in the Cloud (SPC 2015)
DOI: 10.1109/CNS.2015.7346889
- [6] Tetsuya Yokotani, Yuya Sasaki, “Comparison with HTTP and MQTT on Required Network Resources for IoT”, The 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)
DOI: 10.1109/ICCEREC.2016.7814989
- [7] Hun Jung, “Study on the MQTT protocol design for the application of the real-time HVAC”, JInternational Journal of Internet, Broadcasting and Communication Vol.8 No.1 19–26 (2016)
DOI: 10.7236/IJIBC.2016.8.1.19
- [8] R A Atmoko, R Riantini, M K Hasin, “IoT real time data acquisition using MQTT protocol”, Journal of Physics: Conference Series, Volume 853, conference 1
DOI: 10.1088/1742-6596/853/1/012003
- [9] Dokyeong Lee, Deokjai Choi, “Implementation of Zigbee-based Publish/Subscribe System for M2M/IoT Services”, Journal of Korea Multimedia Society 17(12), 2014.12, 1461–1472
DOI: 10.9717/kmms.2014.17.12.1461
- [10] Dong-hyeon Kwon, Ji-yong Lim, Sung-uk Heo, Gwan-Hyung Kim, Am-suk Oh, “Implementation of Internet of Things Maker Kit Based on MQTT Protocol”, Journal of the Korea Institute of Information and Communication Engineering 21(11), 2017.11, 2145–2152
DOI: 10.6109/jkiice.2017.21.11.2145
- [11] Kitae Hwang, Heyjin Park, Jisu Kim, Taeyun Lee, Inhwon Jung, “An Implementation of Smart Gardening using Raspberry pi and MQTT”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 18, No. 1, pp.151–157, Feb. 28, 2018
DOI: 10.7236/JIIBC.2018.18.1.151
- [12] Young-Hwan Jang, Seung-Su Yang, Seok-Cheon Park, “Design and Implementation of Standby Power Cutoff System for Smart Home Environment Based on Zigbee-MQTT”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 17, No. 4, pp.129–135, Aug. 31, 2017
DOI: 10.7236/JIIBC.2017.17.4.129
- [13] Sung-Hoon Mah, Byung-Seo Kim, “Development of Automatic Sensor Detecting-based Home Automation Control Board for Modular Housing”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 17, No. 6, pp.33–39, Dec. 31, 2017
DOI: 10.7236/JIIBC.2017.17.6.33
- [14] Chaebin Na, YeonWoong Choi, SeKwang Kim, JangGui Seo, Kitae Hwang, “An Implementation of Smart Flowerpot made with 3D Printer and NodeMCU”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 17, No. 5, pp.231–237, Oct. 31, 2017
DOI: 10.7236/JIIBC.2017.17.5.231
- [15] Milica Lekić, Gordana Gardašević, “IoT sensor integration to Node-RED platform”, 17th International Symposium INFOTEH, 21–23 March 2018
DOI: 10.1109/INFOTEH.2018.8345544
- [16] Nam Ky Giang, Michael Blackstock, Rodger Lea, Victor C.M. Leung, “Developing IoT Applications in the Fog: a Distributed Dataflow Approach”, 2015 5th International Conference on the Internet of Things (IoT)
DOI: 10.1109/IOT.2015.7356560
- [17] <https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>

저자 소개

오 세 춘(정회원)



- 1984.2 고려대학교 전자공학과(공학사)
- 2017.2 한국산업기술대학교 컴퓨터공학과(공학석사)
- 2017.3 ~ 한국산업기술대학교 컴퓨터공학과(박사과정)
- 1999.9 ~ 2010.1 삼성전자 LCD 총괄 상무이사

<관심분야 : 임베디드시스템, IoT, 스마트팩토리>

김 태 형(준회원)



- 2012.2 : 한국산업기술대학교 컴퓨터공학과(공학사)
- 2017.3 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 석사과정
- <관심분야 : 소프트웨어공학, IoT, 스마트팩토리>

김 영 곤(정회원)



- 1983.2 경북대학교 전자공학과(공학사)
- 1985.2 연세대학교 분대학원 전자공학과(공학석사)
- 2000.2 한국과학기술원 전산학과(공학박사)
- 1985 ~ 2007 KT 수석연구원
- 2007 ~ 한국산업기술대학교 컴퓨터공학과 교수

<관심분야 : 소프트웨어공학, 정보통신시스템, 객체지향 분석 및 설계>

※ 이 논문은 산업통상자원부의 재원으로 한국산업기술진흥원(KIAT)의 지원을 받아 수행된 연구임. (2018년 스마트공장 운영설계 전문인력 양성사업, 과제번호 : N0002429)