

오픈소스기반스마트팜플랫폼표준기술검증에 관한 연구

윤성현, 최문환, 박주영

한국전자통신연구원 표준연구본부

{shy72, mhchoi, jypark}@etri.re.kr

A Study on the Verification of Open Source Based Smart Farm Platform Standard Technology

Sunghyun Yoon, Mun-Hwan Choi, Jooyoung Park

Protocol Engineering Center, Electronics and Telecommunications Research Institute

요약

본 논문에서는 농업의 미래로 각광받고 있는 스마트팜에 적용할 수 표준기술을 검증하기 위해 오픈소스 기반 스마트팜 플랫폼 테스트베드를 구현하고 시험하였다. 작물생육을 위한 환경 및 제어 관련 일차적 데이터를 생성하고 소비하는 스마트팜 장치 개체의 구현에는 MQTT를 적용하였고, 농가 및 작목반 수준의 데이터 관리를 위한 연결성에는 IoTivity를 적용하였다.

I. 서론

스마트팜은 기존의 농업에 IoT 등 ICT 기술을 융합하여 생산성과 효율성을 제고함으로써 고부가가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT기기가 활용된다[1]. 스마트팜의 일반적인 구성은 생육 및 환경 정보를 모니터링 하기 위한 센서 (온도, 습도, CO₂, 토양, 양분 등), 스마트팜 시설을 제어하는 구동기(창개폐기, 환기, 난방, 관수, 관비 등), 각 시설에 부착된 장치들을 제어하는 제어기, 그리고 이들을 통합적으로 관리하는 운영관리시스템 등으로 구성된다. 운영관리시스템은 인터넷을 통해 클라우드와 연결되어 원격에서의 관리를 포함하여 인공지능과 같은 클라우드 기반의 다양한 스마트팜 서비스가 가능하도록 진화하고 있다[2].

그러나 아직 스마트팜에 사용되는 기자재나 서비스들간 상호호환이나 상호연동을 제공하기 위한 표준은 그 기술의 발전 속도에 비해 다소 늦어지고 있다. 이런 이유로 최근 기술 발전속도와 보조를 맞추기 위한 노력이 국내외에서 활발하게 진행되고 있다[3][4].

본 논문에서는 오픈소스[5][6] 기반의 스마트팜 플랫폼 테스트베드 구축을 통해 스마트팜에 적용가능한 표준기술 적용성에 대한 개념검증을 수행함으로써 향후 다양한 분야로 확장될 스마트팜의 가능성을 확인한다.

II. 본론

본 논문에서 구축한 오픈소스 기반 스마트팜 플랫폼에 활용된 오픈소스 소프트웨어는 크게 MQTT와 IoTivity이다. MQTT(Message Queue Telemetry Transport)는 낮은 대역폭, 낮은 전력을 갖는 자원이 제한된 환경에서 비동기로 메시지를 교환하는 경량 메시징 프로토콜로서, 특성상 특히 사물인터넷(IoT)에서 사물간의 통신을 위해 많이 활용되고 있다. MQTT는 Publish/Subscribe(Pub/Sub) 메시징 프로토콜로서, 메모리 및 전력 이용을 효율화하기 위한 가벼운 패킷 구조를 채택함으로써, 저전력, 낮은오버헤드 및 대역, 신뢰할 수 없는 네트워크, No TCP/IP 기반에서 운용할 수 있다는 장점이 있기 때문에, 소형기기의 제어와 센서정보 수집 등에 유리한 특징을 가지고 있다.

IoTivity는 OCF(Open Connectivity Foundation)가 후원하는 오픈소스 프로젝트이다. OCF는 사물인터넷 오픈 플랫폼을 개발하는 글로벌 표준화

기구로서, IoTivity 오픈 소스와 연계한 표준화를 추진한다. 따라서 IoTivity는 OCF의 모든 의무 요구사항을 구현한다.

그림 1은 본 논문에서 구축한 스마트팜 플랫폼 테스트베드 구성이다.

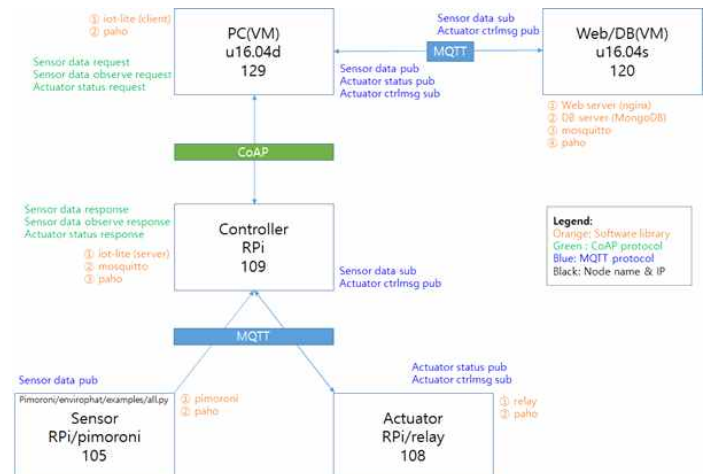


그림 1 - 스마트팜 플랫폼 테스트베드 구성

테스트베드는 크게 5개의 노드로 구성되며 각 노드의 구성은 다음과 같다.

- | | | |
|-----------|-----------------|--------------|
| 1. 센서노드 | RPi w/Pimoroni | Raspbian |
| 2. 구동기노드 | RPi w/GPIO | Raspbian |
| 3. 제어기노드 | Raspberry Pi B+ | Raspbian |
| 4. 운영노드 | Virtual Machine | Ubuntu 16.04 |
| 5. 웹&DB노드 | Virtual Machine | Ubuntu 16.04 |

센서노드 및 구동기노드가 생성하는 데이터는 MQTT를 통해 제어기노드로 전달되며, 구동기노드를 제어하는 제어정보는 MQTT를 통해 구동기노드로 전달된다. 제어기노드와 운영노드간 데이터 교환은 IoTivity를 통해 이루어진다. 제어기노드는 센서노드 및 구동기노드를 대표하는 서버로 동작하고, 운영노드는 클라이언트로서 동작한다. 운영노드와 웹&DB노드 간 데이터 교환은 MQTT를 사용한다.

다음은 각 노드 구성에 대한 세부사항을 기술한다.

① 센서노드

센서노드는 스마트팜의 환경정보를 측정하는 노드로서, 라즈베리파이에 온습도/조도/모션 등의 센서모듈을 가진 Pimoroni 보드를 탑재하여 구현된다. 센서모듈 소프트웨어는 제조사가 제공하는 파이썬 코드를 사용하며, 제어기와 MQTT 통신을 위한 소프트웨어는 오픈소스로 제공되는 Paho 파이썬 라이브러리를 활용한다. 즉, 센서노드는 어떤 센서기기를 사용하는가에 따라 사용하는 센서기기의 제조사가 제공하는 실행코드에 MQTT 라이브러리만 추가된다.

동작이 개시되면 센서노드는 정해진 주기에 따라 센서모듈이 측정한 데이터를 MQTT브로커로 Publish한다.

② 구동기노드

구동기노드는 스마트팜을 제어하기 위한 동작을 수행하는 노드로서, 라즈베리파이 GPIO 호환 릴레이 모듈을 연결하여 구현된다. 센서노드와 마찬가지로 구동기노드 소프트웨어 또한 제조사가 제공하는 파이썬 코드에 MQTT를 위한 Paho 파이썬 라이브러리를 활용하여 구현된다. 상대적으로 저사양 컴퓨팅 환경으로 구성되는 센서노드 및 구동기노드에서 필요한 MQTT 기능은 Pub/Sub만 수행하면 되기 때문에 브로커는 필요치 않고 단지 MQTT 클라이언트 기능만 요구된다.

동작이 개시되면 구동기노드는 최초 구동기노드의 상태를 MQTT브로커로 Publish한다. 또한 제어신호를 받기 위해 MQTT브로커의 해당 구동기노드의 제어명령 관련 Topic을 Subscribe하고, 제어신호를 구독할 때마다 해당되는 동작을 수행하고 그 결과를 Publish한다.

③ 제어기노드

제어기노드는 센서노드와 구동기노드로부터의 데이터를 운영노드로 전달하고, 운영노드로부터의 제어신호를 구동기노드로 전달하는 데이터 전달 기능을 수행하는 노드로서, 라즈베리파이 관련 소프트웨어를 탑재하여 구현된다. 센서노드 및 구동기노드와 연결되어 상호 데이터 교환기능을 제공하는 제어기노드에는 MQTT 서버로서의 기능을 수행하기 위한 Pub/Sub 브로커가 탑재된다. 본 논문의 테스트베드에는 오픈소스 메시지 브로커로 널리 활용되고 있는 mosquitto를 활용한다.

아울러 제어기노드 또한 센서노드 및 구동기노드에서와 같이 MQTT클라이언트가 필요하며, 본 테스트베드에는 paho-mqtt 라이브러리를 활용한다. 다만 제어기노드에서는 파이썬이 아닌 C 라이브러리를 활용하는데, 이는 IoTivity와의 연계를 위함이다.

제어기노드를 IoT 서버로서 동작하기 위해 IoTivity가 활용된다. IoTivity는 경량의 IoT장치에 적합한 iotivity-lite 버전을 활용한다. 제어기노드는 센서노드 및 구동기노드의 데이터를 MQTT로 전달받아 이들에 대한 리소스를 대신하여 클라이언트에 표현하게 된다. 본 테스트베드에는 브릿징 방식이 아닌 멀티서버 방식으로 구현하여 제공된다. 또한 IoTivity 서버인 제어기노드와 IoTivity 클라이언트인 운영노드간 교환되는 데이터 포맷은 JSON이므로 MQTT로 수신된 데이터를 JSON으로 변경하는 cJSON C 라이브러리가 필요하며, 본 테스트베드에는 cJSON#parsing-json 라이브러리를 활용한다.

동작이 개시되면 제어기노드는 연결된 센서노드 및 구동기노드 관련 Topic을 모두 Subscribe한다. onboarding을 통해 연결된 IoTivity 클라이언트가 있으면, 구독을 통해 수신되는 데이터를 IoTivity 메시지로 클라이언트에게 전달한다.

④ 운영노드

운영노드는 제어기노드를 통해 센서노드 및 구동기노드의 상태 및 동작을 관리하고 제어하는 기능을 수행하는 노드로서, 다수의 제어기노드와 연결될 수 있기 때문에 상대적으로 고사양의 컴퓨팅 성능이 요구되므로 일반적으로 PC에 구현되는데, 본 테스트베드에서는 구현의 편의상 리눅스가 실행되는 VM(Virtual Machine)에 탑재된다. 운영노드에는 IoTivity 서버를 탐색하고, 온보딩을 통해 연결하고, 메시지를 주고 받을 수 있도록 IoTivity 클라이언트가 구현된다. 제어기노드와 동일한 iotivity-lite 버전이 활용되며, JSON포맷의 메시지를 처리하기 위해 마찬가지로 cJSON 라이브러리가 요구된다. 아울러 웹&DB노드와 MQTT를 통해 통신할 수 있도록 paho-mqtt 라이브러리가 활용된다. 동작이 개시되면 운영노드는 제어기노드로부터 IoTivity 메시지를 통해 전달된 데이터를 웹&DB노드로 MQTT를 통해 전달한다. 또한 역방향으로는 제어신호가 전달됨으로써, 웹페이지를 통해 센서노드 및 구동기노드를 제어하게 된다.

⑤ 웹&DB 노드

웹&DB노드는 제어명령을 위한 사용자UI와 데이터의 수집 및 저장의 기능을 수행하는 노드로서, 통상 클라우드에 구현되는데, 본 테스트베드에서는 구현의 편의상 리눅스가 실행되는 VM에 탑재된다. 웹서버는 nginx, express, DB서버는 mongodb, 그리고 MQTT브로커는 mosquitto를 활용한다. 센서노드 및 구동기노드가 생성하는 데이터는 제어기노드를 거쳐 운영노드로 전달되고 동시에 웹&DB노드에 저장된다. 제어명령을 위한 사용자UI는 웹 브라우저를 통해 구현된다.

III. 결론

본 논문에서는 오픈소스 기반의 스마트팜 플랫폼 테스트베드 구축을 통해 표준기술의 스마트팜 적용 가능성을 가늠하였다. 스마트팜은 1차 산업인 농업을 생산이 예측 가능한 제조업 수준으로 변모시킬 수 있는 파급효과를 가져올 것이며, 다양한 ICT기기의 활용을 통한 노동력 절감을 통해 농가 삶의 질을 크게 향상시킴으로서 미래 농업의 열쇠로 주목되고 있다. 그 배경에는 스마트팜에 적용될 수 있는 다양한 표준 기술이 개발되어 적용되기 때문이며, 이는 향후 스마트팜 보급의 확산에도 크게 기여하게 될 것이다. 또한 향후 스마트팜은 현재의 시설원예 중심에서 점차 노지분야, 축산 분야, 그리고 농업 뿐만 아니라 어업 및 임업 분야까지도 확장될 전망으로, 스마트팜 표준기술의 활용성은 더욱 중요해지리라 예상된다.

ACKNOWLEDGMENT

본 결과물은 농림축산식품부의 재원으로 농림수산식품기술기획평가원, (재)스마트팜연구개발사업단의 스마트팜다부처패키지혁신기술개발사업의 지원을 받아 연구되었음(421021-03)

참 고 문 헌

- [1] ICT Standard & Certification Special Theme: 스마트팜, TTA저널, vol.180, 2018.12.
- [2] 한국형 스마트팜 기술개발, 농촌진흥청, 2018.
- [3] 윤성현 외, “스마트팜”, ETRI Insight, pp. 29-30, 2019.02..
- [4] 박주영, ‘ITU-T SG13, SG20 스마트팜 표준화 동향’ pp.37, SEP Inside, 2020.06.
- [5] “MQTT: The Standard for IoT Messaging” (<http://mqtt.org>)
- [6] IoTivity (<https://iotivity.org/>)