

<https://doi.org/10.7236/IIBC.2018.18.1.151>

IIBC 2018-1-21

라즈베리파이와 MQTT를 이용한 스마트 가드닝 구현

An Implementation of Smart Gardening using Raspberry pi and MQTT

황기태*, 박혜진**, 김지수***, 이태윤****, 정인환****

Kitae Hwang*, Heyjin Park**, Jisu Kim***, Taeyun Lee****, Inhwan Jung****

요약 본 논문은 라즈베리파이를 이용하여 온도, 토양습도, 조도에 따라 자동으로 물과 빛을 제공하며, 원격 카메라를 이용하여 실시간으로 식물 상태를 전송하는 스마트 화분의 구현 사례를 소개한다. 화분의 용기는 5개의 층으로 분리하고, 각 용기는 3D프린터로 직접 제작하였다. 용기는 5개 층을 연결하여 사용하며 추후 확장할 수 있도록 설계하였다. 그리고 용기 내부에 라즈베리파이와 센서, 펌프, 그리고 카메라를 장착하였다. 본 연구에서는 사용자가 원격에서 카메라나 센서 정보를 받아 스마트 화분을 관찰하고 제어할 수 있도록 안드로이드 앱을 개발하였으며, 앱과 라즈베리파이 사이의 데이터 통신 및 제어는 MQTT 프로토콜을 이용하였다.

Abstract This paper presents an implementation of a smart plant pot which can supply light and water automatically according to the result of detection on current temperature, humidity and illumination, and deliver the images of the plant realtime by using a camera installed in the pot. We designed a container of the plant pot divided into five layers, printed each of them with a 3D printer, and then assembled them. Inside of the container, we installed sensors, a pump, and a camera. We developed an Android application so that the user can control the plant pot and monitor its state. In communication between the Android application and the Raspberry Pi, MQTT protocol was utilized.

Key Words : Smart Gardening, MQTT, Raspberry Pi, 3D Printer

1. 서론

마이크로컴퓨터를 이용하여 소형 전자 제품을 제어하는 임베디드 기술이 인터넷에 연결된 사물들이 데이터를 주고받는 사물인터넷(IoT : Internet of Things) 기술로 발전하고 있다^[1,2,3].

이 논문은 이전 논문에서 NodeMCU와 3D 프린터로 제작한 스마트 화분^[4]을 개선하여 새로 구현한 내용을 소개한다. 스마트 화분은 스스로 식물의 성장 상태를 관찰

하여 수분, 조도, 온도를 조절하는 똑똑한 화분으로 인터넷을 통해 사용자와 연결하거나 정보를 소통 공유하는 인터넷 화분이다^[4,5]. 인도의 그린토피아^[5]가 처음으로 스마트 가드닝 시스템을 판매하기 시작했으며, 국내에서는 2017년도에 교원웰스가 웰스팜 시스템을 개발하여 판매하고 있다^[6].

본 논문에서 새로 구현한 스마트 가드닝 시스템은 이전의 논문에서 사용한 NodeMCU 보드 대신 Broadcom BCM2837 64비트 ARMv8 프로세서와 1GB의 RAM을

*중심회원, 한성대학교 컴퓨터공학부(교신저자)

**준회원, 한성대학교 산업경영공학과

***준회원, 한성대학교 컴퓨터공학부

****준회원, 한성대학교 전자정보공학과

****정회원, 한성대학교 컴퓨터공학부

접수일자: 2017년 11월 21일, 수정완료: 2018년 1월 5일

게재확정일자: 2018년 2월 9일

Received: 21 November, 2017 / Revised: 5 January, 2018

Accepted: 9 February, 2018

*Corresponding Author: calafk@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

장착한 라즈베리파이(Raspberry Pi3)를 사용하고^[7], 카메라 모듈을 장착하여 식물을 촬영하여 사용자에게 전송하는 기능을 추가하였다. 3D 프린터^[8,9]를 이용하여 화분을 담은 용기를 5개의 층으로 나누어 제작하고 조립하여 사용하도록 개선하였다. 내부에는 센서들, 라즈베리파이 보드, 카메라를 심어 놓았으며, 조도가 부족할 때 사용하기 위해 LED 광원을 추가하였다.

사용자가 안드로이드 폰으로 스마트 화분과 통신할 수 있도록 PC 서버에 MQTT broker를 구축하였다. MQTT broker는 IoT 장치들이 텍스트 기반의 메시지를 전달하기 쉽도록 설계된 메시지 중계 서버이다. 스마트 화분의 라즈베리파이는 인터넷으로 MQTT broker를 통해 사용자와 연결하도록 구현하였다^[10,11]. 이들은 MQTT broker를 이용하여 센서 데이터나 명령들을 주고받는다. 사용자는 안드로이드 단말기를 통해 화분에게 명령을 내리고 카메라를 작동시켜 사진을 찍을 수 있도록 앱을 수정하였다. 라즈베리파이가 촬영한 사진 이미지를 안드로이드 단말기에 전송하기 위해 라즈베리파이에 몽구스 웹 서버(Mongoose)^[12]를 설치하고 안드로이드 단말기에서는 웹을 이용하여 사진 이미지를 전달받는다.

2장에서 본 논문의 연구 배경을 설명하고, 3장에서 스마트 가드닝의 전체 시스템의 설계에 대해 설명한다. 4장에서는 구현한 내용을 소개하며, 5장에서 결론을 맺는다.

II. 연구 배경

1. 라즈베리파이

라즈베리 파이는 영국의 라즈베리파이 재단이 학교에서 기초 컴퓨터 과학교육을 증진시키기 위한 단일 칩을 사용한 저가형 싱글보드 컴퓨터이다^[7]. 라즈베리파이는 데비안, 아치 리눅스 및 QtonPi 등의 리눅스 배포판의 운영체제가 탑재 가능하기 때문에 리눅스에서 실행되는 모든 응용프로그램을 실행시킬 수 있어 응용성이 매우 높다. 센서나 카메라 등을 연결하기 쉬운 하드웨어 구조를 가지고 있으며, 센서들로부터 값을 읽고 제어할 수 있는 다양한 라이브러리가 지원되기 때문에 개발자는 임베디드 제품이나 IoT 장치를 쉽게 개발할 수 있다.

2. MQTT 서버

MQTT(Message Telemetry Transport or Message Queue

Telemetry Transport)는 ISO 표준으로 TCP/IP 위에서 실행되는 subscribe-publish 방식으로 메시지를 주고받는 통신 프로토콜이다^[10].

MQTT 프로토콜은 그림 1과 같이 메시지를 중계해주는 MQTT broker(혹은 MQTT 서버)를 사이에 두고 양단에는 MQTT 클라이언트가 있다. 그림에서 번호는 프로토콜의 동작 과정을 나타낸다. 메시지를 받고자 하는 수신 클라이언트는 특정 토픽과 함께 자신을 MQTT broker에 가입(subscribe)하는 절차를 거친다. 이 토픽에 해당하는 메시지를 기다린다는 의미이다. 메시지를 보내고자 하는 송신 클라이언트는 MQTT broker에게 토픽과 함께 메시지를 보낸다(publish). MQTT broker는 토픽을 기다리는 가입들에게 송신자가 보낸 메시지를 대신 전송해준다. MQTT는 최근들어 IoT 장치들의 메시지 전송을 위해 많이 사용되고 있다.

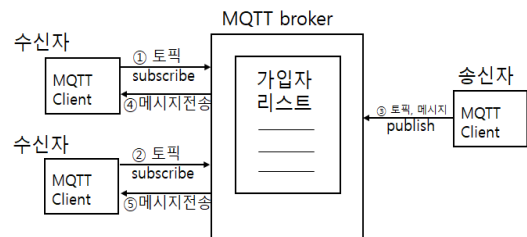


그림 1. MQTT 프로토콜
Fig. 1. MQTT Protocol

3. 몽구스 웹 서버

몽구스 웹 서버는 윈도우, 리눅스, 맥 운영체제 환경에서도 동작하도록 만든 오픈 소스 웹 서버이다^[12]. 몽구스는 표준 C 코드로 작성되었으며, 윈도우, 리눅스, 맥 운영체제 등 거의 모든 환경에서 컴파일된다. 소스 코드의 크기가 매우 작고 컴파일된 실행 코드 역시 작아 적은 량의 메모리를 가진 임베디드 환경에 매우 적합하다.

본 논문에서는 라즈베리파이에서 카메라로 촬영한 이미지를 안드로이드 단말기로 보내기 위해, 몽구스 웹 서버의 C 소스를 라즈베리파이에서 컴파일하여 사용하였다.

III. 시스템 설계

1. 시스템 구성

본 연구에서 구현한 전체 시스템은 그림 2와 같다.

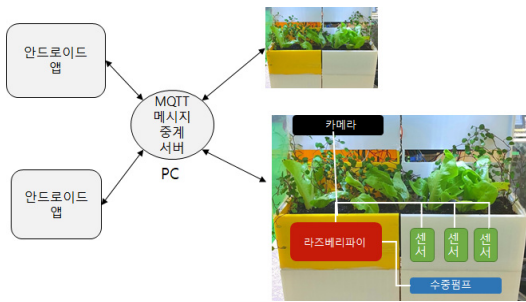


그림 2. 전체 시스템 구성
 Fig. 2. System View

스마트 가드닝 시스템은 MQTT broker가 실행되는 서버 컴퓨터(PC)와 스마트 화분, 그리고 사용자의 안드로이드 단말기에서 실행되는 앱으로 구성된다.

화분 컨테이너 내 제어장치의 하드웨어와 소프트웨어 모듈은 그림 3과 같다. 안드로이드 앱에서는 화분의 ID와 작물을 등록한 후, 센서값을 받아 작물에 대한 현재 상태를 확인하고 실시간으로 카메라를 찍어 확인할 수 있는 기능을 제공한다. 안드로이드 앱과 스마트 화분이 메시지를 주고받기 위해서는 MQTT 프로토콜을 사용한다.

안드로이드 앱은 라즈베리파이의 카메라가 찍은 사진을 라즈베리파이에서 실행되는 몽구스 웹 서버에 접속하여 웹 페이지와 함께 이미지를 내려 받아 출력한다.

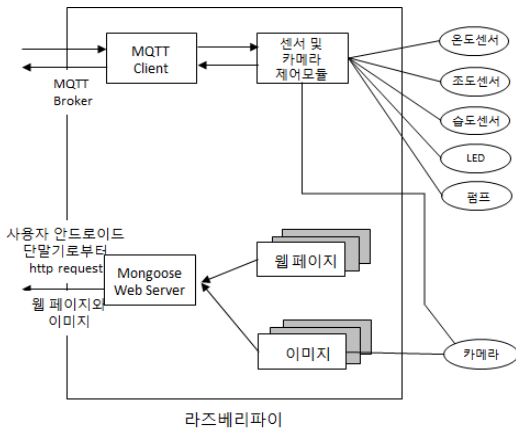


그림 3. 스마트 화분 내 제어 모듈 구성
 Fig. 3. Control Modules in Smart Plant Pot

2. 제어 모듈 설계

가. 하드웨어

스마트 화분을 제어하는 제어 장치로 기존의 NodeMCU 대신 라즈베리파이3 모델B를 이용하였고, 라

즈비안 리눅스를 설치하였다. 그리고 라즈베리파이 전용 카메라, 자외선 LED, ADC 컨버터, 수증펌프, 토양수분 센서, 온도센서, 조도 센서를 연결하였다. 수증모터펌프는 12v를 인가하기 위해 MOSFET 소자를 이용하여 외부전원 장치로 가동하였다. 자외선 LED는 채소를 키우는데 필요한 빛을 제공해주기 위한 광원으로 사용하였으며, 잎채소, 열매채소에 따라 적색과 청색의 비율이 3:1 또는 5:1인 LED를 사용할 수도 있다. ADC 컨버터는 라즈베리파이에서 아날로그 입력 핀이 없기 때문에 아날로그 값을 받기 위해 필수적으로 필요하며, 아날로그 값을 디지털로 바꾸어주는 컨버터를 이용하여 센서 값을 입력받았다.

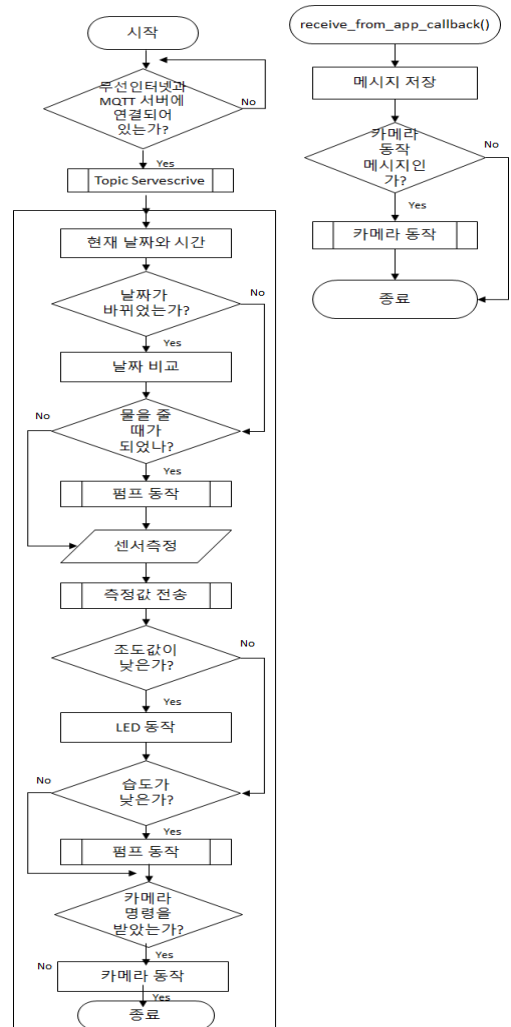


그림 4. 라즈베리파이의 제어 프로그램 알고리즘
 Fig. 4. Control Software Algorithm in Raspberry Pi

나. 제어 프로그램

라즈베리파이 보드에서 작동하는 스마트 가드닝의 기능은 크게 두 부분으로 나뉜다. 하나는 안드로이드 앱의 명령을 받아 카메라가 사진을 찍도록 하는 기능이며, 다른 하나는 온도 습도 조도 등을 측정하여 자동으로 온도, 조도, 습도를 조절하면서 식물을 재배하는 기능이다. 작물 종류(상추, 적근대, 로메인, 청경채)에 따라 적정토양 습도와 적정조도를 측정하여 물주기와 빛 주기 기능을 수행하고, 안드로이드 앱에게는 작물 현재 상태를 메시지 형태로 보낸다. 전체적인 알고리즘은 그림 4와 같다.

3. 서버 및 통신 설계: MQTT 프로토콜과 Mongoose 웹서버

사용자와 스마트 화분 사이에 전달되는 데이터의 종류는 센서 값들과 사진 이미지의 2 종류이다. 본 연구에서는 센서 데이터의 값들은 MQTT broker를 통해 전달하며, 사진 이미지는 웹 서버를 이용하여 전달한다. 각 센서 데이터 값은 4 바이트 정수이므로 텍스트 메시지를 보내는데 사용되는 MQTT를 이용해도 별 문제가 없지만, 이미지는 상대적으로 크기 때문에 별도로 웹을 이용하여 전송하는 것이 바람직하고 또한 구현하기 쉽다.

가. 센서 값 전송

본 연구에서는 그림 5와 같이 MQTT 프로토콜을 구현하지 않고, MQTT broker로 구현한 Mosquitto^[13,14] 오픈 소스의 바이너리를 윈도우 PC에 설치하여 사용하였다. 여기서 안드로이드 앱이 MQTT 수신 클라이언트가 되고, 라즈베리파이 보드가 센서 값을 읽어 전송하는 MQTT 송신 클라이언트가 된다. MQTT 송신 클라이언트와 MQTT 수신 클라이언트 소프트웨어는 모두 본 연구에서 개발하였다. 본 연구에 사용된 MQTT 토픽은 표 1과 같다.

표 1. MQTT 메시지 토픽
Table 1. MQTT Message topics

토픽	관련 데이터
LIGHT	측정 조도
TEMPO	측정 온도
HUMI	측정 토양 습도
IMG	사진 찍기 명령

안드로이드 앱은 3개의 토픽 LIGHT, TEMPO, HUMI 과 함께 MQTT broker에게 가입(subscribe)하여, 센서 값을 기다린다. 라즈베리파이에 작성된 제어 프로그램은 무한루프를 돌면서 조도, 온도, 습도 센서 값을 각각 주기적으로 읽어 LIGHT, TEMPO, HUMI 토픽과 함께 각 센서 값(메시지)을 MQTT broker에게 보낸다(publish).

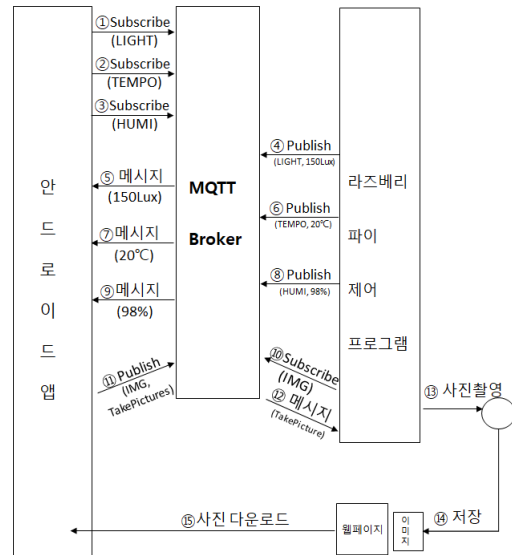


그림 5. 안드로이드 앱과 라즈베리파이 사이의 통신
Fig. 5. Data communications between Android App and Raspberry Pi

나. 사진 촬영 및 사진 이미지 전송

라즈베리파이에 부착한 카메라가 찍은 사진을 안드로이드 앱으로 보내기 위해 몽구스 웹 서버를 이용한다.

먼저 라즈베리파이의 제어 프로그램은 MQTT broker에게 IMG 토픽과 함께 이미지를 받기 위한 가입 절차(subscribe)를 진행한다. 사용자가 앱의 ‘사진 찍기’ 메뉴를 실행하면, 앱은 IMG 토픽과 함께 사진을 촬영할 것을 지시하는 메시지(“TakePicture”)를 MQTT broker에게 보낸다(publish). 메시지(“TakePicture”)를 전달 받게 되면 라즈베리파이의 제어 프로그램은 카메라를 제어하여 사진을 촬영하고, 찍힌 사진을 몽구스 웹 서버의 사진 폴더에 저장한다. 안드로이드 앱은 자동으로 몽구스 웹 서버에 접속하여 사용자에게 사진을 보여준다.

이전의 논문에서도 밝힌 바와 같이 스마트 화분의 제어 장치와 사용자 앱이 네트워크를 통해 직접 데이터를 전송하지 않고, 따로 메시지 서버와 웹 서버를 두는 이유는 여러 스마트 화분과 사용자들을 연결하거나 앱의 사

용자 단말기가 꺼져 있는 경우에도 메시지와 이미지 전송이 이루어지도록 하기 위함이다^[4].

4. 화분 컨테이너 설계

화분 컨테이너는 3D Max를 이용하여 3차원으로 모델링하고, 3D 프린터를 이용하여 직접 제작하였다. 그림 6은 컨테이너를 5개 층으로 분리하여 설계한 3D Max 컨테이너의 도면을 보여준다. 물이 담기는 물받이와 물통층, 물을 배수하는 층, 식물을 담는 화분 층, 화분의 겉통과 센서와 펌프가 놓여지는 부분과 라즈베리파이와 카메라를 놓는 층, 그리고 덮개 층으로 구성된다. 작은 3D 프린터로도 모듈을 프린팅하여 쉽게 조립할 수 있도록 하였다.

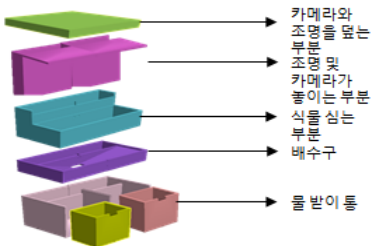


그림 6. 3D MAX로 그린 화분
 Fig. 6. Plant Pot modeled by 3D MAX

5. 안드로이드 앱 설계

안드로이드 앱의 목적은 스마트 화분으로부터 주기적으로 화분 상태를 보고 받아 사용자에게 그래픽으로 보여주는 것과 화분의 상태를 카메라로 찍어 실시간으로 전송받는 목적이다. 안드로이드 앱은 2 개의 액티비티(Activity)로 비교적 간단하게 구성하였으며, 각 액티비티는 그림 7과 같다.

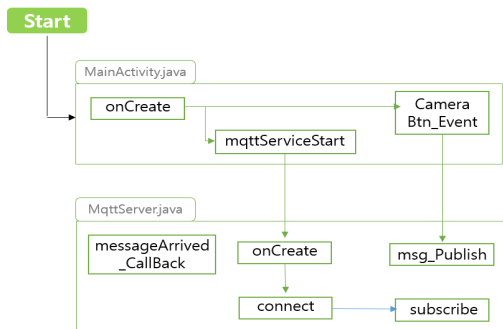


그림 7. 안드로이드 앱의 구조
 Fig. 7. Software modules in Android App

IV. 구현

그림 8은 화분 컨테이너를 3D 프린터로 출력하고 있는 모양을 보여준다. 3D 프린터는 큐비콘 사의 3dp-110f를 이용하고, PLA(Polylactic acid) 친환경 소재로 프린팅하였다.

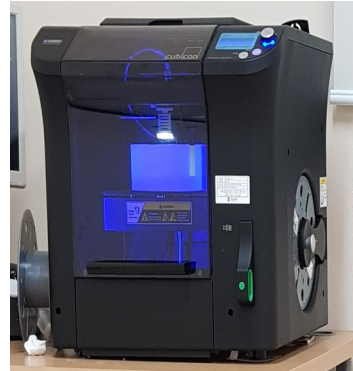


그림 8. 3D 프린터로 화분을 프린트하는 모습
 Fig. 8. Plant Pot being printed by 3D printer

그림 9는 식물, 제어 장치, 센서 등 모든 요소를 넣어 작동하는 스마트 화분의 모습을 보여준다. 조도가 약할 때 조도를 밝게 할 목적으로 화분의 천정에는 LED 조명을 달았다.



그림 9. 구현된 스마트 화분
 Fig. 9. Smart Plant Pot implemented

라즈베리파이를 기반으로 제작된 제어 장치는 그림 10에서 보여준다.

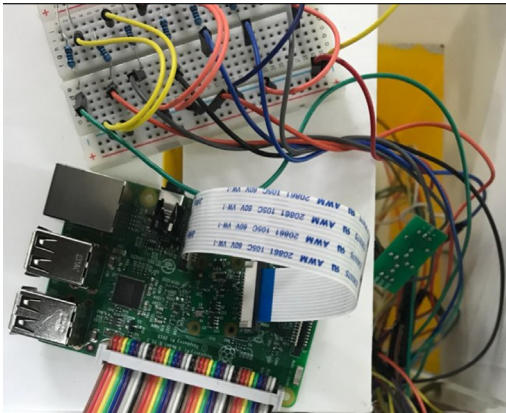


그림 10. 라즈베리파이로 제작된 스마트 화분 제어 장치
Fig. 10. Microcontroller made with RaspberryPi



(a) 스마트 화분으로 키울 식물 설정



(b) 화분의 온도, 조도, 습도 상태 보기

(c) 카메라가 찍은 식물 사진 보기

그림 11. 스마트 화분과 통신하는 안드로이드 앱
Fig. 11. Android Application communicating with Smart Plant Pot

그림 11은 구현된 안드로이드 앱을 실행한 스크린샷이다. 그림 11(a)에서 사용자가 키울 식물과 적정 조도, 습도, 온도 값을 설정하여 스마트 화분이 스스로 조도와 습도를 조절하도록 한다. 그림 11(b)는 스마트 화분의 조도, 습도, 온도의 3가지 상태를 주기적으로 받아 보여주

는 안드로이드 앱의 화면이다. 그림 11(c)는 스마트 화분의 식물상태를 실시간으로 전송받아 보는 화면이다. 사용자가 앱에서 '사진 찍기' 명령을 내리면 MQTT broker를 통해 라즈베리파이 카메라가 사진을 찍고, 앱은 사진 이미지는 몽구스 웹 서버에 접속하여 볼 수 있다.

V. 결론

본 논문에서는 이전 연구에서 사용한 NodeMCU 대신 성능이 뛰어나고 리눅스를 운영체제로 하여 개발 환경이 우수한 라즈베리파이를 이용하여 스마트 가드닝 시스템을 다시 구현한 사례를 소개하였다. 3D 프린터로 화분과 물, 센서, 제어 장치, 카메라를 담은 용기를 만들고, 조도, 온도, 습도를 감지하여 스스로 습도와 조도를 조절하도록 제어 장치를 프로그래밍하고, 카메라 제어를 통해 실시간으로 식물 상태를 볼 수 있도록 하고, MQTT broker와 몽구스 웹 서버를 설치하여 안드로이드 앱과 센서 데이터 및 사진 이미지를 주고받도록 하였다. 조도를 조절하기 위해 특별히 LED를 화분의 천정에 추가하였으며, 작은 3D 프린터로도 만들기 쉽도록 화분 컨테이너를 4개의 층으로 분할 제작하여 조립하는 방식을 택하였으며, 옆으로 여러 개의 화분 컨테이너를 연결하여 많은 식물을 재배할 수 있도록 하였다.

추후 현재 부착된 카메라로부터 자라는 식물의 동영상을 스트리밍을 실시간 전송할 수 있는 기능과 스스로 빛을 따라 회전하거나 이동하는 등의 화분에 지능을 더할 계획이다.

References

- [1] S. T. Kim, J. S. Jeong, J. K. Song, H.Y. Kim, Trends of IoT Device Platforms and Building its Ecosystems, 2014 Electronics and Telecommunications Trends, Vol. 29, No. 4, Oct. 2014
- [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] SeoHyung Kim, IoT : Internet of Things Technology, IEIE Magazine, Vol. 43, No.3, pp.64-71, 2016

[4] C. Na, Y. Chio, S. Kim, J. Seo, K. Hwang, An Implementation of Smart Flowerpot made with 3D Printer and NodeMCU, The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 17, No. 5, pp.231-237, Oct. 2017

[5] <https://www.greenopia.com/>

[6] <http://m.kyowonwells.com/wellsfarm>

[7] https://en.wikipedia.org/wiki/Raspberry_Pi

[8] Global 3D printer industry and technology trend analysis, Technical Policy for Mechanical Engineering, ISSN 2287-8106, No. 71, Sep. 2013

[9] https://en.wikipedia.org/wiki/3D_printing

[10] <https://en.wikipedia.org/wiki/MQTT>

[11] Kim Dong Hwi, Oh Hyeung Seok, Jeon Hyun Sig, Lee Chang Seok, Park Hyun J, A Study on implementation of MQTT Logic for IoT Platform, Proceedings of 2017 winter conference, The Korean Institute of Communications and Information Sciences, pp. 1056-1057, 2017

[12] <https://code.google.com/archive/p/mongoose/>

[13] <https://mosquitto.org/>

[14] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," The Journal of Open Source Software, vol. 2, no. 13, May 2017

박 혜 진(준회원)



• 한성대학교 산업경영공학과
 <주관심분야 : 임베디드 시스템>

김 지 수(준회원)



• 한성대학교 컴퓨터공학부 재학
 <주관심분야 : 임베디드 시스템>

이 태 윤(준회원)



• 한성대학교 전자정보공학과 재학
 <주관심분야 : 임베디드 시스템>

정 인 환(정회원)



• KAIST 정보및통신공학과 박사
 • 삼성전자 시스템사업부 수석연구원
 • 한성대학교 컴퓨터공학과 교수
 <주관심분야 : 망관리, 멀티미디어통신, IoT>

저자 소개

황 기 태(종신회원)



• 서울대학교 컴퓨터공학과 학사
 서울대학교 컴퓨터공학과 석사
 서울대학교 컴퓨터공학과 박사
 • 경력
 University of California, Irvine 방문 교수
 University of Florida 방문 교수

<주관심분야 : 모바일 시스템>

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음