

<https://doi.org/10.7236/IIBC.2017.17.5.231>

IIBC 2017-5-32

## 3D 프린터와 NodeMCU를 사용한 스마트 화분의 구현

# An Implementation of Smart Flowerpot made with 3D Printer and NodeMCU

나채빈\*, 최연웅\*, 김세광\*, 서장규\*, 황기태\*\*

Chaebin Na\*, YeonWoong Choi\*, SeKwang Kim\*, JangGui Seo\*, Kitae Hwang\*\*

**요약** 본 논문은 온도와 빛, 수분의 량을 탐지하여 자동으로 습도와 조도를 채워주는 스마트 화분을 만든 사례를 소개한다. 화분을 담는 용기는 3D 프린터로 제작하고, 용기의 내부에는 Wi-Fi가 내장된 NodeMCU 마이크로컨트롤러를 장착하였다. 온도와 습도, 조도를 감지하는 센서와, 펌프를 NodeMCU에 연결하고 제어 프로그램을 작성하여 NodeMCU 스스로 습도와 조도를 조절하도록 하였다. 또한 subscribe-publish 모델로 작동하는 MQTT 서버를 구축하고, NodeMCU와 본 연구에서 개발한 안드로이드 앱이 정보를 교환하도록 하였다. 안드로이드 앱의 사용자는 식물에게 적절한 온도, 습도, 조도를 스마트 화분에게 전달하고, 스마트 화분으로부터 현재 온도, 습도, 조도 정보를 모니터링 할 수 있게 하였다.

**Abstract** This paper presents an implementation of a smart flowerpot which can adjust humidity and illumination automatically after monitoring the temperature, humidity, and illumination. We made a container of the flowerpot with a 3D printer and embedded a NodeMCU micro controller in it. We attached a temperature sensor, a humidity sensor, an illumination sensor, and a water pump to the NodeMCU. We developed a control program that adjusts humidity and illumination and ran it on the NodeMCU. Also we developed an Android application and set up an MQTT server. Using the MQTT server, the NodeMCU and the Android application can exchange messages which keep sensor values and commands. Using the Android application, the user can send the proper temperature, humidity, and illumination to the smart flowerpot and monitor the sensor values.

**Key Words** : Smart, Flowerpot, IoT, MQTT

## I. 서 론

인터넷의 발전은 개인이 혼자 사용하던 컴퓨터 사용의 패러다임에서 전 세계의 사용자들을 연결하고 이들 사이에 정보를 소통하고 공유하는 컴퓨터 사용의 패러다임으로 바꾸었다. 최근에 와서는 사물에 다양한 센서와

통신 가능한 마이크로컨트롤러(micro controller)를 장착하고, 사물 주변으로 데이터를 수집하고 인터넷에 연결하여 다른 컴퓨터와 정보를 주고받는 사물인터넷(IoT : Internet of Things) 기술로 발전하고 있다.[1,2]

사물인터넷 기술은 스마트 홈, 스마트 팜, 의료, 교육, 웨어러블 장치, 에너지 분야, 엔터테인먼트 등 매우 광범

\*준회원, 한성대학교 컴퓨터공학부

\*\*중신회원, 한성대학교 컴퓨터공학부(교신저자)

접수일자: 2017년 7월 26일, 수정완료: 2017년 9월 26일

게재확정일자: 2017년 10월 13일

Received: 26 July, 2017 / Revised: 26 September, 2017 /

Accepted: 13 October, 2017

\*\*Corresponding Author: calafk@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

위한 분야에 응용을 시도하고 있다. 사물인터넷 기술에는 데이터를 수집하는 센서와 이를 처리하고 네트워크로 연결하는 마이크로컨트롤러가 핵심적인 역할을 한다.[1,4]

사물인터넷의 도래와 함께 마이크로컨트롤러의 기술도 발전하였다. 전문 개발자들에 의해서만 이루어져 왔던 임베디드 장치(IoT 장치)와 임베디드 소프트웨어는 오픈 소스 아두이노(Arduino) 플랫폼의 등장으로 누구나 쉽게 개발할 수 있게 되어 순식간에 다양한 응용 분야로 확산되었다. 그 후 초기 아두이노에 사용된 AVR 마이크로컨트롤러 대신 메모리와 처리 속도 향상, Wi-Fi 기능 부착 등 기능과 성능이 강화된 다양한 마이크로컨트롤러가 개발되어 상용화되었다. 마이크로컨트롤러의 기술 발전과 함께 값싸고 다양한 센서들이 시장에 공급되어 사물인터넷은 다양한 분야에서 더욱 발전하고 있다.[1,3]

본 논문은 스마트 화분을 대상으로 사물인터넷의 기초 형태를 구현한 사례를 소개한다. 스마트 화분은 스스로 화분 속의 식물의 성장 상태를 관찰하여 자동으로 수분, 조도, 온도를 조절하는 똑똑한 화분이다. 그리고 스마트 화분은 인터넷을 통해 사용자와 연결하거나 정보를 소통 공유하는 인터넷 화분이다. 인도의 Greenopia[5]가 처음으로 스마트 화분을 제작하여 판매하기 시작했고, 국내에서는 2013년도에 엔싱이 플랜티라는 스마트 화분을 개발하고 지금도 판매하고 있다.[6] 둘 다 다양한 센서를 기반으로 화분에 심겨진 식물의 상태를 파악하고 앱을 통해 사람과 공동으로 식물을 키우는 기능을 제공한다. 최근에는 여러 중소기업 혹은 개인들이 스마트 화분을 만들고 있다.

본 논문에서는 3D 프린터[7,8]를 이용하여 화분을 담을 용기를 설계 제작하고, 내부에는 센서들과 화분의 작동을 관측 제어하는 제어 장치를 제작하여 심어 놓았다. 본 연구에서는 이 제어 장치로 W-Fi 기능을 내장한 ESP8266 마이크로컨트롤러[4]로 갖추고 사물인터넷 장치를 개발할 수 플랫폼인 NodeMCU 보드[9]를 이용하였다. ESP8266에 내장된 W-Fi 모듈을 인터넷으로 서버에 접속하고 서버를 통해 사용자로 연결하도록 구현하였다. 사용자의 안드로이드 단말기는 서버와 접속하여 화분을 모니터링하고 화분에게 명령을 내릴 수 있는 앱을 제작하여 작동시킨다. 서버는 IoT 장치와 사용자 컴퓨터 사이에 간단한 메시지 형태로 데이터를 전달할 수 있도록 MQTT 서버[10,11]를 구축하여 사용하였다.

본 논문은 2장에서 본 논문의 연구 배경을 설명하고,

3장에서 스마트 화분의 전체 시스템의 설계에 대해 설명한다. 4장에서는 구현한 내용을 소개하며, 5장에서 결론을 맺는다.

## II. 연구 배경

### 1. 오픈 소스 IoT 플랫폼

IoT 단말기를 개발하는 오픈 소스 플랫폼의 시초가 된 것은 아두이노(Arduino)이다.[12] 아두이노는 2005년, 이탈리아 IDII(Interaction Design Institutelvrea)에서 하드웨어에 익숙지 않은 학생들이 자신들의 디자인 작품을 손쉽게 제어할 수 있도록 하기 위해 고안되었다.

아두이노는 PC에서 C 언어로 작성하여 크로스 컴파일(cross compile)한 바이너리 코드를 USB를 통해 쉽게 업로드 할 수 있어 어셈블리어로 작성하는 부담이 없기 때문에, 누구든지 쉽게 개발할 수 있는 장점이 있다. 또한 아두이노 보드의 가격은 비교적 저렴하고, 윈도우를 비롯해 맥 OS X, 리눅스와 같은 여러 OS에서 개발 환경이 지원된다. 아두이노 보드는 장난감, 로봇, 의료, 교육, 자동차 등 다양한 응용에 개발되고 있다.

아두이노 외에 ARM CPU를 사용하는 Freescale 단말 플랫폼, 누리텔레콤의 AiMiR, TI 사에서 만든 LaunchPad 등 여러 종류의 IoT 단말 개발 플랫폼이 개발되어 있고, 이들보다 고성능의 프로세서와 고용량 메모리를 탑재하고 더 많은 데이터를 수집하고 처리할 수 있는 게이트웨이급 수준의 단말기를 개발할 수 있는 Raspberry Pi, BeagleBoard, Edison 등의 플랫폼도 상용화 되어 있다.[1]

### 2. 3D 프린터

3D 프린터는 계층적으로 물질을 쌓아가면서 3차원 물체를 만들어내는 장치이다. 1981년 일본 나고야 시 공업연구소의 고다마 히데오(小玉秀男)가 처음 이론화했고 1986년 미국의 척 헐(Chuck Hull)이 특허를 얻어 설립한 3D 시스템스(3D Systems)사에서 처음으로 제품화하였다.

가공방식은 적층가공과 절삭가공으로 크게 두 가지로 나뉘는데 3차원 인쇄는 적층가공 방식에 속한다. 적층가공은 가루나 액체 형태의 재료를 균허가며 한 층씩 쌓는 방식이다. 비교적 복잡한 모양을 만들 수 있고, 제작과 채색을 동시에 진행할 수 있다는 장점이 있다. 다만 완성품

의 표면이 매끄럽지 못하여 품질이 상대적으로 떨어진다.

3D프린터의 제작 과정은 모델링, 프린팅, 마무리를 거친다. 모델링은 일반적으로 CAD 또는 3차원 모델링 소프트웨어를 이용하여 3차원 데이터를 완성하며, 모델링 과정에서 만들어진 도면을 이용해 물체를 만들어낸다. 인쇄 과정은 사용 방법과 모델의 크기와 복잡성에 따라 몇 시간에서 며칠 정도의 시간이 소요될 수 있다. 마무리로 인쇄된 결과물에 대해서 필요할 경우 사포로 연마하거나, 색칠하거나, 인쇄된 파트들을 조립하는 공정이 추가될 수 있다.

3D 프린터의 보급은 제 3의 산업혁명으로 불리며 미래 산업 전반에 걸쳐 제조 기술의 큰 변화를 가져올 것으로 예상되고 있다. 기존의 생산 방식을 탈피하여 일괄된 방식으로 어떤 형태의 제품도 만들어낼 수 있기 때문이다. 치과 등의 의료 분야는 물론, 각종 가정용품을 비롯해 자동차나 비행기 등에 쓰이는 기계장치도 3D 프린터에 의한 생산이 가능하다. 이미 자동차 업계에서는 엔진 등 핵심 부품을 3D 프린터로 만들어내는 공정을 연구하고 있다.[8]

### III. 스마트 화분의 설계

#### 1. 전체 시스템 설계

본 연구에서 구현하고자 하는 스마트 화분의 전체 시스템은 그림 1과 같다.

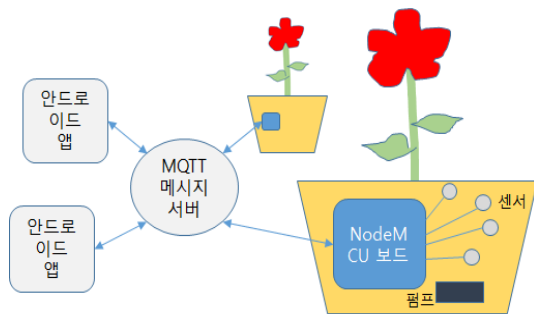


그림 1. 전체 시스템 구성  
 Fig. 1. System View

스마트 화분 시스템은 화분을 담는 컨테이너와 스마트 화분의 제어 장치, 사용자 안드로이드 스마트 폰에서 실행되는 앱, 그리고 스마트 화분의 제어 장치와 안드로이드

이드 앱의 메시지 통신을 위해 구축된 메시지 서버로 구성된다. 화분을 제어하는 제어 장치는 NodeMCU 보드와 센서들, 그리고 물 펌프로 구성된다.

제어 장치로 사용되는 NodeMCU 보드는 Wi-Fi를 가진 ESP8266 마이크로컨트롤러를 갖추고 있고, 화분, 제어 장치, 센서, 펌프를 담는 컨테이너는 3D 프린터로 제작하였다. 사용자가 안드로이드 단말기로 화분의 습도, 온도, 조도 등을 볼 수 있도록 앱을 작성하고, 이 앱과 제어 장치 사이의 통신을 위해 MQTT 서버를 구축하였다. MQTT 서버는 subscribe-publish 프로토콜로 간단한 메시지를 주고받는 서버이다.[10]

MQTT 서버를 두는 이유는 안드로이드 앱이 꺼져 있을 때, 스마트 화분의 제어 장치가 보낸 메시지를 잃어버리지 않고 저장해두었다가, 앱이 실행되면 그 때 전달받도록 하기 위함이며, 여러 개의 스마트 화분과 여러 명의 사용자를 연결하기 위함이다.

#### 2. 제어 모듈

##### 가. 하드웨어 설계

스마트 화분을 제어하기 위한 제어 장치로는 NodeMCU 오픈 소스 IoT 플랫폼을 활용하였다. NodeMCU는 ESP8266 Wi-Fi Soc(System on Chip) 칩에서 실행되는 펌웨어를 포함하고 있고, 기존의 아두이노 개발 환경(scratch)을 그대로 사용할 수 있도록 동일 환경을 지원한다.

NodeMCU를 이용하면 Wi-Fi 기능을 요하는 IoT 디바이스를 쉽게 개발할 수 있다. 현재 본 논문에서 사용한 NodeMCU는 80MHz의 ESP8266 CPU를 갖추고 128KB의 메모리에 4MB의 저장장치를 갖추고 있다.

제어 모듈은 화분 주변의 정보를 입력받기 위해 토양 습도센서, 온도센서, 조도센서, 수위센서를 연결하며, 화분 속에 담긴 물을 끌어 올려 식물에게 물을 주기 위한 펌프와 펌프를 가동시키기 위해 외부 전원 제어장치를 사용하였다.

##### 나. 소프트웨어 설계

NodeMCU 보드에서 작동하는 제어 소프트웨어는 센서 모니터링 및 모터 제어 부분과 메시지 서버와 통신 부분으로 구성되며 알고리즘으로 나타내면 그림 2와 같다.

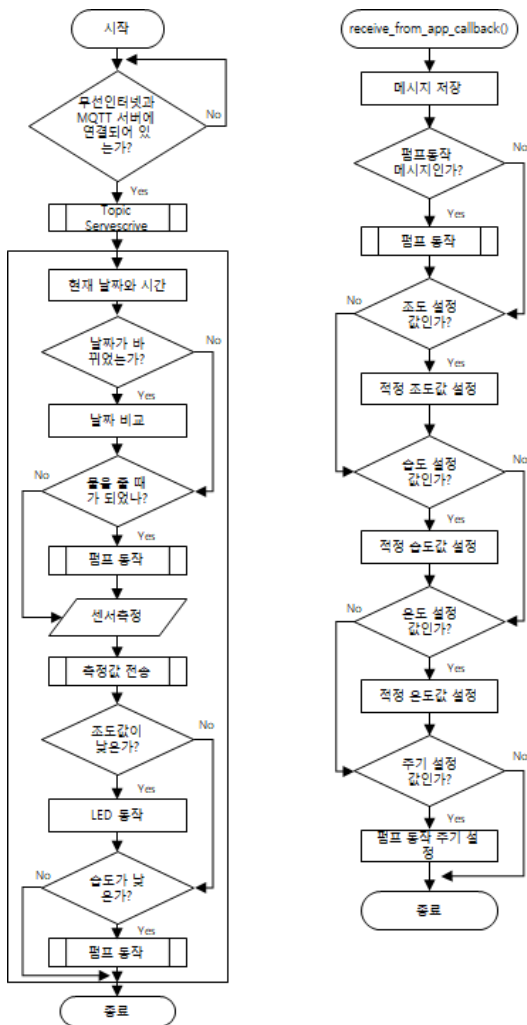


그림 2. 세션 모니터링 및 펌프 제어 알고리즘  
Fig. 2. Algorithm for sensor monitoring and pump control

### 3. 서버 및 통신 설계 : MQTT 프로토콜

스마트 화분의 제어 장치와 사용자 앱이 네트워크를 통해 직접 데이터를 전송하지 않고, 따로 메시지 서버를 두는 이유는 여러 스마트 화분과 사용자들을 연결하거나 앱의 사용자 단말기가 꺼져 있는 경우에도 메시지 전송이 이루어지도록 하기 위해서이다.

메시지 서버와 스마트 화분, 그리고 서버와 안드로이드 앱 사이의 통신은 MQTT(Message Telemetry Transport) 프로토콜을 이용한다. MQTT는 ISO 표준으로 TCP/IP 위에서 실행되는 subscribe-publish 방식으로 메시지를 주고 받는 통신

프로토콜이다.[10] 메시지를 받고자 하는 쪽은 먼저 서버에 가입(subscribe) 한다. 메시지를 보내고자 하는 쪽에서 메시지 서버에 메시지를 보낸다(publish). 메시지 서버는 송신자를 대신하여 받은 메시지를 모든 가입자들에게 전송한다.

subscribe-publish 방식에서 수신자는 받고자하는 메시지의 토픽(topic, 메시지 종류나 특징)을 지정하고, 송신자는 메시지를 보낼 때 메시지 토픽을 함께 보낸다. 메시지 서버는 그림 3과 같이 메시지 중계자(broker)를 두고, 도착한 메시지를 토픽에 따라 분류하여 가입자에게 메시지를 보낸다.

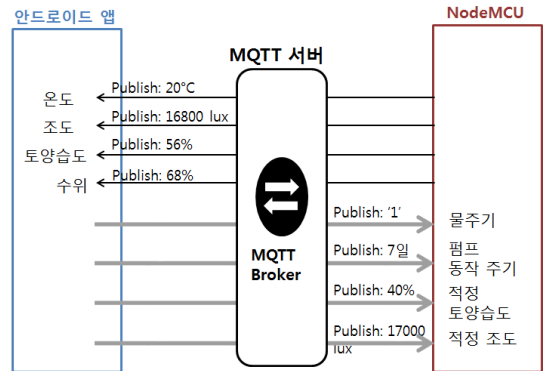


그림 3. MQTT 메시지 중계자  
Fig. 3. MQTT Message Broker

본 연구에서는 MQTT 프로토콜을 구현하지 않고, MQTT 프로토콜을 구현한 Mosquitto[13,14]로 불리는 오픈 소스의 바이너리를 그대로 윈도우 PC에 설치하여 서버를 구축하였다. 표1은 본 연구에서 사용한 메시지 토픽들이다.

표 1. 스마트 화분과 앱에서 사용한 MQTT 메시지 토픽  
Table 1. MQTT Message topics used in Smart Flowerpot and Android Application

토픽	관련 데이터	가입자
/Greeni/cds	측정 조도	NodeMCU, 앱
/Greeni/temp	측정 온도	앱
/Greeni/soilhumi	측정 토양 습도	NodeMCU, 앱
/Greeni/waterlevel	측정 물탱크수위	앱
/Greeni/watercycle	사용자 설정 펌프 동작 주기	NodeMCU
/Greeni/opsoilhumi	사용자 설정 토양 습도	앱
/Greeni/opcds	사용자 설정 적정 조도	NodeMCU

#### 4. 화분 컨테이너 설계

화분의 컨테이너는 3D Max를 이용하여 3차원으로 모델링하고, 3D 프린터를 이용하여 직접 제작하였다. 그림 4는 3D Max로 모델링한 컨테이너의 도면으로, 물이 담긴 물받이와 물통, 식물을 담은 내부 화분, 화분의 걸통, 그리고 센서와 펌프, NodeMCU 보드가 놓여지는 부분으로 구성된다.

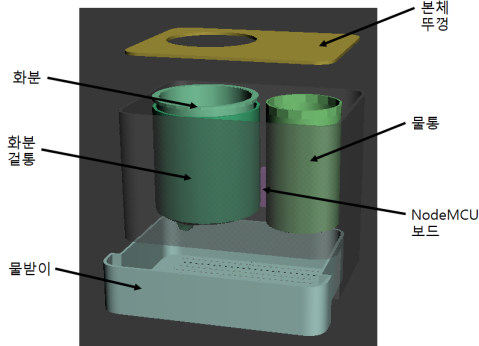


그림 4. 3D MAX로 그린 화분  
 Fig. 4. Flowerpot modeled by 3D MAX

#### 5. 안드로이드 앱 설계

안드로이드 앱의 목적은 2가지이다. 스마트 화분으로부터 주기적으로 화분 상태를 보고 받아 사용자에게 그래픽으로 보여주는 목적과 초기에 화분에 기르는 식물의 온도, 습도, 조도의 적절한 값을 스마트 화분으로 전송하는 목적이다.

이를 위해 안드로이드 앱은 실행을 시작할 때, MQTT 서버에 메시지 토픽과 함께 가입자로 등록한다. 안드로이드 앱은 2 개의 액티비티(Activity)로 비교적 간단하게 구성하였으며, 각 액티비티는 그림 5와 같다.

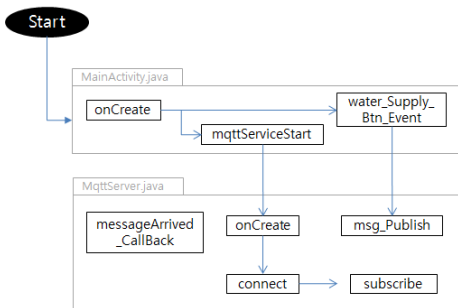


그림 5. 안드로이드 앱의 구조  
 Fig. 5. Software modules in Android App

#### IV. 구현

이 절에서는 스마트 화분의 구현에 대해 설명한다. 그림 6은 컨테이너를 3D 프린터로 출력하고 있는 모양을 보여준다. 3D 프린터는 큐비콘 사의 3dp-110f를 이용하였다.

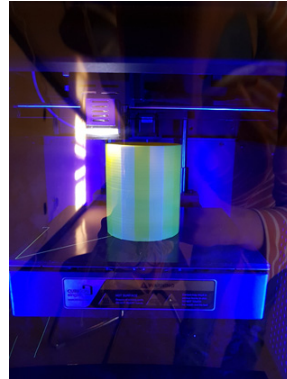


그림 6. 3D 프린터를 화분을 출력하는 모습  
 Fig. 6. Flowerpot being printed by 3D printer

그림 7은 완성된 컨테이너 식물과 제어 장치, 센서 등 모든 요소를 넣어 작동하는 스마트 화분의 모습을 보여 준다.



그림 7. 구현된 스마트 화분 전체  
 Fig. 7. Smart Flowerpot implemented

NodeMCU를 기반으로 제작된 스마트 화분의 제어 장치는 그림 8에서 보여준다.

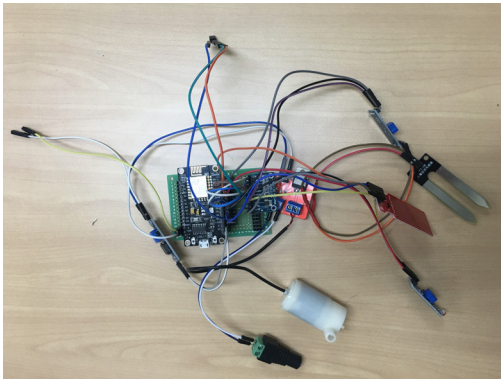
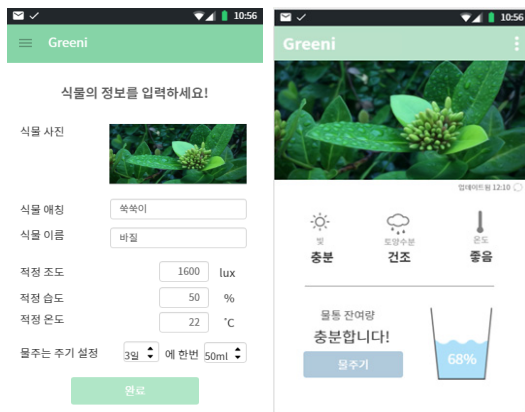


그림 8. NodeMCU로 제작된 스마트 화분 제어 장치  
Fig. 8. Microcontroller made with NodeMCU

그림 9는 구현된 안드로이드 앱을 보여준다. 그림 9(a)는 사용자가 스마트 화분에게 식물을 기르기에 필요한 정보를 보내는 화면이다. 적정 조도, 습도, 온도 값을 전달하여 스마트 화분이 스스로 조도와 습도를 조절하도록 한다.

그림 9(b)는 스마트 화분의 조도, 습도, 온도의 3가지 상태와 물통에 있는 물의 량을 보여주는 화면이다. 스마트 화분이 MQTT 서버에 주기적으로 보낸 데이터를 MQTT 서버가 앱으로 보낸 데이터이다. 조도, 습도, 온도의 구체적인 정보를 앱의 다른 액티비티를 통해 볼 수 있다. 물통에 있는 물의 량이 충분치 않은 경우, 화면의 색깔이 변하고 사용자에게 물통에 물을 채워야 함을 경고한다.



(a) 스마트 화분으로 필요한 정보 제공

(b) 스마트 화분의 상태 보여주기

그림 9. 스마트 화분과 통신하는 안드로이드 앱  
Fig. 9. Android Application communicating with Smart Flowerpot

스마트 화분의 동작 테스트는 시간이 많이 걸리는 어려움으로 인해 다양한 환경에서 많은 실험으로 검증하지는 못하고, 간단한 실험으로 화분의 작동을 검증하였다. 실험은 조도, 온도, 습도 센서를 물건으로 가려서 다양한 상황에서 스마트 화분의 제어 장치가 잘 작동하고 안드로이드 앱으로 메시지가 잘 전달되는지 확인하였으며, 대학 내 경진 대회를 통해 성능을 검증하였다. 실험을 지속하다가 물을 너무 많이 주어 식물이 죽는 일이 생기기도 하였다.

## V. 결론

사물인터넷 기술이 산업의 전분야로 특히 생활을 전분야로 파고들고 있다. 본 연구에서는 NodeMCU IoT 플랫폼을 이용하여 스마트 화분을 구현한 사례를 소개하였다. 3D 프린터로 화분과 물, 센서, 제어 장치를 담은 용기를 만들고, 조도, 온도, 습도를 감지하여 스스로 습도와 조도를 조절하도록 제어 장치를 프로그래밍하고, MQTT 서버를 설치하여 안드로이드 앱과 정보를 주고 받도록 하였다. 앱의 사용자는 원격에서 식물에게 필요한 최적의 온도, 조도, 습도를 알려주고, 화분의 상태 정보를 알 수 있도록 하였다. 추후 보다 면밀한 작동을 실험하고, 식물을 기르기에 필요한 다른 센서를 발굴하고 식물 데이터베이스를 만들 계획이다. 또한 화분에 회전 모터와 바퀴를 달아 스스로 빛을 따라 회전하거나 이동하는 기능을 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] <https://en.wikipedia.org/wiki/ESP8266>
- [5] <https://www.greenopia.com/>
- [6] <https://nthing.net/ko/planty>
- [7] Global 3D printer industry and technology trend analysis, Technical Policy for Mechanical Engineering, ISSN 2287-8106, No. 71, Sep. 2013
- [8] [https://en.wikipedia.org/wiki/3D\\_printing](https://en.wikipedia.org/wiki/3D_printing)
- [9] [http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html)
- [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://www.arduino.cc/>
- [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

**저자 소개**

**나 채 빈(학생회원)**



• 한성대학교 멀티미디어공학과 재학  
 <주관심분야 : 빅데이터>

**최 연 응(학생회원)**



• 한성대학교 컴퓨터공학부 재학  
 <주관심분야 : IoT>

**김 세 광(학생회원)**



• 한성대학교 컴퓨터공학부 재학  
 <주관심분야 : 컴퓨터보안>

**서 장 규(학생회원)**



• 한성대학교 컴퓨터공학부 재학  
 <주관심분야 : 모바일 컴퓨팅>

**황 기 태(정회원)**



• 서울대학교 컴퓨터공학과 학사  
 서울대학교 컴퓨터공학과 석사  
 서울대학교 컴퓨터공학과 박사  
 •경력  
 University of Callifornia,  
 Irvine 방문교수  
 University of Florida 방문 교수  
 <주관심분야 : 모바일 시스템>

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음