

## 사물 인터넷을 활용한 재배 시설의 공기 질 관리 시스템 구현

이 평 화<sup>1</sup> · 민 정 익<sup>2</sup> · 송 주 환<sup>2\*</sup>

<sup>1</sup>전주대학교 스마트미디어학과 학사과정

<sup>2</sup>전주대학교 스마트미디어학과 교수

# Implementation of Air Quality Management System for Cultivation Facility using IoT(Internet of Things)

Pyeong-Hwa Lee<sup>1</sup> · Jeong-Ik Min<sup>2</sup> · Ju-Whan Song<sup>2\*</sup>

<sup>1</sup>Undergraduate course, Department of Smart Media, Jeonju University, Jeonju 55069, Korea

<sup>2</sup>Professor, Department of Smart Media, Jeonju University, Jeonju 55069, Korea

### [요 약]

최근 미세먼지에 관한 관심이 높아지고 있다. 세계보건기구 WHO(World Health Organization)를 비롯한 다수의 기관은 대기오염이 사람과 동식물에 매우 위험함을 알리고 있다. 경제활동인구의 감소와 특히 농업 종사자 감소는 인간 생활의 3요소 중에서 식(食)을 위협하고 있다. 농업 분야에 사물 인터넷(IoT; Internet of Things)을 적용하면 농업인에게 편리함과 효율을 증대할 수 있을 것으로 여겨진다. 각종 센서를 통하여 농업 환경 정보를 수집하고 각종 시설을 자동 제어할 수 있다. 본 논문에서는 농업 종사자 감소와 미세먼지 등 환경적 요인으로 인한 문제에 대처할 수 있는 재배 시스템을 제안한다. 이 시스템은 라즈베리 파이에 부착된 센서를 통하여 공기 질을 측정하고 측정값을 활용하여 자동으로 환기 시스템을 작동하고 스마트폰 애플리케이션을 통해 원격 제어와 모니터링을 할 수 있다.

### [Abstract]

Recently, interest in fine dust is increasing. Many organizations, including the WHO(World Health Organization), have announced that air pollution is very dangerous for people, animals and plants. The decline in the economically active population, and in particular the decline in agricultural workers, threatens eating among the three elements of human life. The application of the IoT(Internet of Things) in agriculture is expected to increase convenience and efficiency for farmers. Through various sensors, agricultural environment information can be collected and various facilities can be automatically controlled. In this paper, we propose a cultivation system that can cope with problems caused by reduction of agricultural workers and environmental factors such as fine dust. The system measures air quality through sensors attached to the Raspberry Pi, and uses the measured values to automatically activate the ventilation system and remotely control and monitor via smartphone applications.

**색인어** : 사물 인터넷, 미세먼지, 라즈베리 파이, 센서, FCM

**Key word** : IoT(Internet of Things), Fine Dust, Raspberry Pi, Sensor, FCM

<http://dx.doi.org/10.9728/dcs.2020.21.3.627>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 30 January 2020; **Revised** 15 March 2020

**Accepted** 25 March 2020

**\*Corresponding Author; Ju-Whan Song**

**Tel:** +82-63-220-2912

**E-mail:** [jwsong@jj.ac.kr](mailto:jwsong@jj.ac.kr)

## I. 서론

무역경제신문의 이광희 기자는 ‘농업 IoT, 우리 농업 살릴 수 있다.’ 기사를 통해 주변 나라 중 초고령 사회를 넘어선 지 오래된 일본에서는 전체적인 경제활동인구가 줄어들고 더불어 농업 종사자 또한 감소하고 있다고 말한다. OPTiM 사에서는 기존 농업과 차별화되는 IoT 적용 모델을 제시하였고 시장 깊은 곳까지 영향력을 끼치고 있음을 말하였다[1]. 대한민국의 국정 모니터링 지표인 e-나라지표에 나타난 표를 보면 2010년 약 117만 농가에 300만 농업인구가 있었지만, 2018년에는 102만 농가에 231만 농업인구로 농가 수는 약 13%, 농가 인구는 약 24%의 감소하고 있다[2]. 우리나라도 앞으로의 경제활동인구 감소와 농업 종사자 감소에 대한 대응책을 마련하여야 한다.

현재 국내의 농업에서 스마트팜이 도입되어 유리온실 또는 비닐하우스 등이 자동화되어 적절한 기준으로 관리되고 있다. 그러나 아직은 일부 농가에서만 스마트팜을 도입해서 사용하고 있고, 대부분의 재배 시설은 농민의 경험에 의한 관리를 하고 있다[3]. 따라서 소규모의 비닐하우스를 관리하기 위한 시스템이 필요하다.

사물 인터넷(Internet of Things: IoT)은 생활 속 사물들을 유무선 네트워크로 연결해 정보를 공유하는 기술환경을 말한다[4]. 본 논문에서는 사물 인터넷 기술을 활용하여 소규모 농업 시설의 공기 질을 제어하기 위한 시스템을 구현하였다.

본 논문은 라즈베리 파이를 기반으로 공기 질을 측정하고 일정 수치 이상의 값이 측정될 때는 자동으로 환기팬을 작동시키고, 경고등과 경고음을 작동시켜 작업자가 인지할 수 있도록 하였다. 또한, 관리자가 원격으로 농업 시설의 현재 공기 질을 확인할 수도 있고, 원격으로 환기 팬 등을 제어할 수 있도록 하였다. 이를 위하여 시스템을 농업에 적용하기 위해서는 IoT의 기본 조건인 인터넷, 무선 통신과 외부 접속을 위한 포트 포워딩 등이 갖춰진 환경으로 세팅이 되어있어야 한다.

인터넷 기술과 보급이 발달한 대한민국에서 기초적인 환경 설정만 갖추진다면 초고속 인터넷 속도와 이미 많은 곳에 보급이 된 인터넷을 사용하여 농업 분야에 적용할 수 있을 것으로 판단된다. IoT 중 센서 모듈과 데이터베이스, 통신 등을 작물 재배 환경에 맞추어 자료를 수집 제공하고 시설을 제어하는 시스템을 구현하였다.

본 논문의 구성은 다음과 같다. II장에서는 본 논문에서 구현한 IoT를 활용 시스템에 관한 관련 연구를, III장에서는 시스템의 구성과 개발환경과 라즈베리 파이에 구현된 서버 프로그램에 대해 설명하고, IV장에서는 안드로이드로 구축된 클라이언트 프로그램에 관해 설명하고, V장에서는 결론 및 향후 연구 방향을 서술한다.

## II. 관련 연구

### 2-1 농업 IoT에 대한 정부의 노력

우리나라의 농업 시설현대화 사업 추진은 1968년부터 비닐 하우스 설치를 지원하였고, 2000년대 이후에는 생산시설이 노후화되었고, 2010년 u-IT 사업으로 농업 경쟁력 향상을 목표로 한 생산 정밀화 모델이 개발이 본격화되고 생산성 향상 등의 성과가 나타났다. 2013년 농식품 ICT 융복합 확산대책을 마련해 생산, 유통, 소비 등 부문별 ICT 융복합 현황을 진단하고, 스마트팜 보급, R&D, 산업 생태계 조성 등 정책 방향이 설정된 후 스마트팜을 집중적으로 보급하고 있다.

대한민국은 2014년부터 보급사업을 시작하여 2017년 기준, 시설 온실 4,000ha(현대화된 온실의 40%), 스마트축사 790개소(전업농의 3.4%)에 유형별로 스마트팜 보급 및 확산이 이루어졌다. 22년까지 7,000ha(시설현대화 면적의 70%), 축사 농가 5,750호(전업농의 25%)에 스마트팜 보급을 목표로 하고 있다[5].

농촌경제연구소의 농업 전망 2019에서는 스마트팜의 보급 실적 및 확산 목표를 보여주고 있고, 주요 국가 농업 관련하여 4차 산업혁명 대응 동향 등의 정부의 스마트팜 보급 목표와 다른 나라와 우리나라의 대응 동향에 관해서 설명하고 있다[6]. 표 1에는 우리나라의 스마트팜의 보급 실적과 확산 목표를 나타냈다. 보급 실적은 매년 10% 이상의 증가율을 보임을 알 수 있다.

표 1. 스마트팜 보급 실적 및 확산 목표

Table 1. Smart Farm Distribution Performance and Target

Plants	'17	'18	'19	'20	'21
Tomato	785	824	865	909	954
Paprika	575	587	601	614	627
Strawberry	600	720	864	1,037	1,244
Cucumber	201	231	266	306	352
Watermelon	600	660	726	799	878
Melon	400	460	529	608	700
Flower	839	965	1,110	1,276	1,467
Total	4,000	4,447	4,961	5,549	6,222
The rate of increase		11.2%	11.6%	11.9%	12.1%

### 2-2 라즈베리 파이(Raspberry Pi)와 아두이노(Arduino)

본 논문을 통해 구현한 시스템에서는 센서 제어에 많이 사용하는 아두이노 대신 라즈베리 파이 3B+를 사용하였다. 아두이노는 자전거, 라즈베리 파이는 자동차와 같다고도 표현할 만큼 라즈베리 파이의 성능은 아두이노보다 뛰어나다. 아두이노의 경우 ATmega8, ATmega168, ATmega328과 같은 megaAVR 시리즈의 칩을 사용하고 있고 라즈베리 파이의 경우 브로드컴에서 제작된 고성능 칩을 사용한다.

아두이노의 경우 단독으로 제어하기 어렵고 다른 기기를 이용해 접근해야 하는 번거로움이 있지만, 라즈베리 파이는 자체적으로 리눅스 기반의 라즈비안(Raspbian) 운영체제가 구동되기 때문에 다른 기기에 의존하지 않고 단독으로 운영할 수 있어 본 논문에서 사용하게 되었다. 하드웨어 구성으로 볼 때 여러 개의 USB 포트와 인터넷에 연결할 수 있는 이더넷 포트, 스피커 잭, HDMI 포트 등이 있어 본 논문에서 제안하는 시스템 구현에 적합하다. PC나 노트북에 비해 가볍고 작아 이동하기 좋고 가격도 저렴하게 구매할 수 있어 라즈베리 파이를 사용했다. 라즈베리 파이는 MicroSD 카드에 리눅스 운영체제를 기반으로 한 운영체제 라즈비안을 설치하여 사용하였다.

### III. 시스템 설계 및 구현

#### 3-1 시스템 구성 및 개발환경

본 논문을 통해 제안하고자 하는 시스템의 구조는 그림 1과 같다.

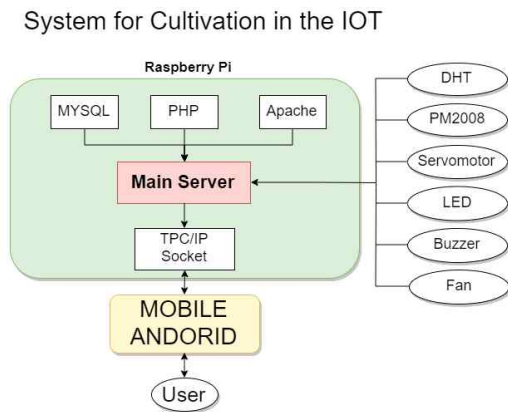


그림 1. 시스템 구조도  
Fig. 1. System Framework

시스템 구현을 위해서 첫 번째 라즈베리 파이에 라즈비안 운영체제를 설치하고, 네트워크에 자동으로 연결될 수 있도록 설정한다. 두 번째로는 프로그램이 구동되기 위한 소프트웨어를 설치하였다. 설치한 소프트웨어는 웹서버로 아파치(Apache) 웹서버를, 데이터베이스로는 Mysql DB, 서버 프로그래밍 언어인 PHP를 설치하였고, 개발용 프로그래밍 언어는 라즈비안에 기본으로 설치되어있는 파이썬 3(Python 3)을 사용하였다. 추가 센서 구동에 필요한 라이브러리는 구현하면서 설치하여 사용하였다. 라즈베리 파이의 전원이 들어오면 파이썬으로 개발한 서버(Server) 프로그램이 구동된다. 서버 프로그램은 라즈베리 파이에 연결된 각 센서의 정보를 초기화하고, 그들이 측정한 값들을 받아 데이터베이스에 저장한다.

연결된 센서들은 정해진 시간 단위로 온도, 습도, 미세먼지 등을 실시간으로 측정하여 라즈베리 파이에 전자 신호로 전달

되고 라즈베리 파이는 이 값들을 데이터베이스에 저장한다. 또한, 클라우드 환경인 ThingSpeak 사이트(<http://thingspeak.com>)에 측정된 데이터를 저장하여 측정값들을 그래프로 볼 수 있도록 하였다.

서버 프로그램은 측정된 값이 임계치를 넘게 되면 관리자에게 통보하고, 자동으로 팬을 작동시켜 환기하거나, 서보모터를 이용해서 창문을 여닫도록 하였다. 관리자는 외부에서 웹(Web)을 통하여 접속하거나 전용 안드로이드 앱으로 접속하여 측정값들을 볼 수 있고, 각 센서를 제어할 수 있도록 하였다.

#### 3-2 라즈베리 파이와 센서 모듈 연결

본 논문에서는 측정용 센서로 온습도 센서인 DHT, 미세먼지 센서인 PMS7003, 알람을 주기 위해 LED와 Buzzer 모듈을, 환기를 위해서 Fan과 Servomotor를 사용하였다. 표 2에 본 연구에 사용한 각종 센서에 대한 정보를 나타냈다.

표 2. 센서 정보  
Table 2. Sensor Specifications

Sensor	Characteristics
DHT	humidity measuring range : 20-90% RH humidity accuracy : ±5% RH temperature measuring range : 0-50 °C temperature accuracy : ±2% °C power supply : 3 - 5.5V DC low power consumption
PMS7003	minimum distinguishable particle diameter : 0.3µm power supply : 5V high anti-interference performance because of the patent structure of six sides shielding interface : I2C, UART physical size : 48(W) x 37(H) x 12(D) [mm]
LED	power supply : +3.3-5V pins : Signal(Yellow), Vcc(Red), GND(black)
Buzzer	power supply : +5V interface type : analog pins : S-Signal, V-VCC, G-GND physical size : 19 x 27mm
Fan	physical size : 120mm x 120mm x 25mm power supply : 12V
Servomotor	physical size : 21.8mm x 11.4mm x 22.6mm rotation range : 0 - 180° power supply : 4.8 V

각 모듈은 라즈베리 파이와 연결하기 위해 다음과 같이 작업하였다.

1) DHT11 : DHT 모듈은 11과 22의 2가지 종류가 있는데, 본 논문에서는 그중 11을 사용하였다. DHT 모듈을 사용하기 위해서는 복잡한 프로그래밍이 필요한데, 이 부분을 라이브러리로 만들어져 있어 내려받아 사용하면 된다. 우리는 Adafruit\_Python\_DHT 라이브러리를 내려받아 사용하였다.

DHT11은 VCC, GND, Signal의 3개의 핀(pin)이 있는데, VCC와 GND는 각각 라즈베리 파이의 전원과 그라운드(ground) 핀에 연결하고 Signal은 GPIO 핀에 연결하여 사용하였다.

2) PMS7003 : 미세먼지 센서 PMS7003을 사용하여 센서값을 얻기 위해서는 PMS7003 외에 인터페이스 보드와 USB to Usart 변환케이블을 이용하여 PMS7003과 라즈베리 파이를 연결하였고 PMS7003 라이브러리를 내려받아 사용하였다.

3) LED 모듈 : 미세먼지의 측정값이 임계치를 넘어서면 주위에 알리기 위해서 LED 모듈을 사용한다. LED 모듈은 VCC, GND, Signal의 3개의 핀이 있는데, VCC와 GND는 각각 라즈베리 파이의 전원과 그라운드 핀에 연결하고 Signal은 GPIO 핀에 연결하여 사용하였다. LED 모듈은 신호를 보내거나 멈추는 것으로 제어하기만 하면 되기 때문에 GPIO를 통해 신호를 보내는 코드를 작성하여 사용하였다.

4) Buzzer 모듈 : LED 모듈과 함께 미세먼지 알람을 위해 Buzzer 모듈을 사용한다. Buzzer 모듈은 VCC, GND, Signal의 3개의 핀이 있는데, VCC와 GND는 각각 라즈베리 파이의 전원과 그라운드 핀에 연결하고 Signal은 GPIO 핀에 연결하여 사용하였다. Buzzer 모듈은 신호를 보내거나 멈추는 것으로 제어하기만 하면 되기 때문에 GPIO를 통해 신호를 보내는 코드를 작성하여 사용하였다.

5) Fan 모듈 : 미세먼지와 온도 측정 중 임계치 초과할 때 작동하게 되는 Fan 모듈을 사용하였고, VCC, GND와 회전 방향을 지정하는 Signal 2개 등 총 4개의 핀으로 구성되어 있다. 회전 방향 핀에 전원이 가해지면 해당 방향으로 팬이 회전하여 내부 공기를 외부로 배출할 수도 있고, 외부 공기를 내부로 들어들 수도 있다.

6) Servomotor : 미세먼지와 온도 측정에 있어서 임계치에 따라서 180도의 회전각을 조절하여 창문을 여닫는 것을 구현하였다. Servomotor 모듈은 VCC, GND, Signal의 3개의 핀이 있는데, VCC와 GND는 각각 라즈베리 파이의 전원과 그라운드 핀에 연결하고 Signal은 GPIO 핀에 연결하여 사용하였다.

### 3-3 데이터베이스에 저장

서버 프로그램은 정해진 시간마다 센서에서 측정된 값을 라즈베리 파이 데이터베이스에 저장하게 되는데 파이썬으로 개발된 서버 프로그램에서 pymysql 라이브러리를 사용하여 MySQL과 연결하고 측정값들을 저장하였다. 처음 MySQL에 측정값을 저장할 때 테이블이 없으면 테이블 생성(CREATE)하게 되고 테이블이 있으면 데이터 입력(INSERT)하게 된다. 표 3에는 측정된 값들을 저장한 테이블의 데이터 일부를 나타냈다.

표 3. MySQL에 저장된 데이터

Table 3. Stored Data in MySql

No	Date	pm_1_0_cfl	pm_2_5_cfl	pm_1_0_atm	pm_2_5_atm	air_0_3	air_0_5	air_1_0
1	2019-11-17 11:39	9	10	9	10	1473	429	43
2	2019-11-17 11:42	8	9	8	9	1314	380	29
3	2019-11-17 11:45	9	12	9	12	1446	439	65
4	2019-11-17 11:48	9	12	9	12	1575	477	59
5	2019-11-17 11:51	8	10	8	10	1401	411	54
6	2019-11-17 11:54	8	10	8	10	1353	403	33
7	2019-11-17 11:57	7	9	7	9	1287	387	40
8	2019-11-17 12:00	8	10	8	10	1395	423	54
9	2019-11-17 12:03	8	10	8	10	1392	423	49
10	2019-11-17 12:06	8	10	8	10	1404	415	37
11	2019-11-17 12:09	8	9	8	9	1458	421	29
12	2019-11-17 12:12	7	8	7	8	1254	386	34
13	2019-11-17 12:15	9	11	9	11	1386	424	55
14	2019-11-17 12:18	8	9	8	9	1365	409	41
15	2019-11-17 12:21	9	11	9	11	1557	461	41
16	2019-11-17 12:24	10	13	10	13	1635	508	59
17	2019-11-17 12:27	8	10	8	10	1386	419	36
18	2019-11-17 12:30	8	10	8	10	1380	421	78

클라우드 사이트인 ThingSpeak에 값을 올린다. Thing Speak은 인터넷, 근거리 통신망을 통해 HTTP 및 MQTT 프로토콜을 사용하여 사물에서 데이터를 저장하고 검색하는 오픈 소스 사물 인터넷 응용 프로그램 및 API이다. 회원가입을 한 후, 미리 미세먼지와 온도 key 이름을 지정하여 그래프 설정을 한다. 라즈베리 파이 서버에서 ThingSpeak 주소로 Write API Key와 데이터값들을 보내게 되면 사전에 설정한 그래프의 Write API Key 값의 그래프를 찾아 데이터값이 추가되고 그래프가 그려진다. SCI를 통해서 표 3의 값들이 ThingSpeak에 업로드되고 이를 차트화한 결과가 그림 2와 그림 3이다. 그림 2는 PM1.0에 대한 측정값을 차트로 보여주고, 그림 3은 측정 온도를 차트로 보여준다.



그림 2. Thingspeak PM1.0 차트  
Fig. 2. Thingspeak PM1.0 Chart

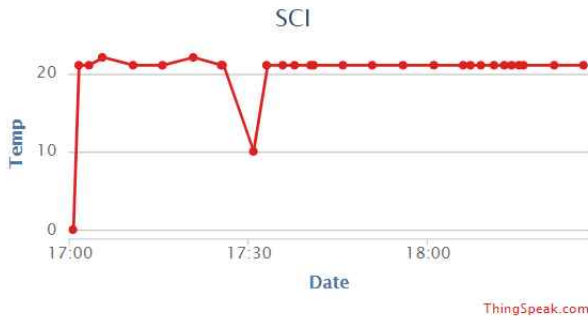


그림 3. Thingspeak 온도 차트  
Fig. 3. Thingspeak Temp Chart

### 3-4 센서의 자동 제어

미세먼지 센서(PMS 7003)와 온도 센서(DHT11)를 이용해 자동화를 구현하기 위해서 표 4의 조건을 기준으로 환기시설을 작동하도록 했다.

표 4. 센서 작동 기준

Table 4. Sensor Operating Standard

Fine Dust	Temperature	Ventilation System
higher	lower	Off
higher	higher	Off
higher	normal	Off
lower	lower	Off
lower	higher	On
lower	normal	On

실내 온도가 적정 온도를 벗어나게 되면 광합성 작용과 호흡량 증가로 탈수 상태에 빠질 수 있고 냉해나 동해를 입을 수 있다. 그리고 미세먼지는 광합성에 장애가 될 수 있기에 미세먼지가 임계치 이상, 미세먼지가 적정이고 실내 온도가 미만일 때에는 환기시설을 폐쇄, 이외에는 환기시설을 개방하게 되는 논리를 적용하였다. 미세먼지의 경우 세계보건기구(WHO) 권고 기준( $\mu\text{g}/\text{m}^3$ )을 참고하면 권고기준인 PM-10이 50 이상, PM-2.5는 25 이상이 되면 적용하도록 초기화했다. 표 5는 2005년에 WHO에서 발표한 일평균 미세먼지에 대한 권고기준과 잠정목표를 나타냈다[7].

표 5. 미세먼지 권고기준(24시간 평균)

Table 5. Guideline levels for each pollutant(24-hour concentrations)

	PM-10	PM-2.5	Basis for the selected level
Interim target-1 (IT-1)	150	75	Based on published risk coefficients from multi-centre studies and meta-analyses (about 5% increase of shortterm mortality over the AQG value).
Interim target-2 (IT-2)	100	50	Based on published risk coefficients from multi-centre studies and meta-analyses (about 2.5% increase of shortterm mortality over the AQG value).
Interim target-3 (IT-3)	75	37.5	Based on published risk coefficients from multi-centre studies and meta-analyses (about 2.5% increase of shortterm mortality over the AQG value).
Air quality guideline (AQG)	50	25	Based on relationship between 24-hour and annual PM levels.

### 3-5 데이터 전송

애플리케이션에 데이터를 불러오기 위해서 라즈베리 파이 웹 서버를 구동하여 데이터를 JSON 형태로 보여주는 PHP 파일을 제공하여 주게 되고 클라이언트에서 이 PHP 파일 주소로 접속하여 MySQL에서 데이터를 불러오게 된다.

웹 서버를 구현하기 위해서 아파치(Apache), PHP, MySQL을 사용하였다. 라즈베리 파이 서버 프로그램은 센서의 측정값들을 MySQL에 저장하고 클라이언트가 웹서버로 측정 데이터를 요구하면 웹서버에 최신의 데이터값을 JSON 포맷으로 변환하여 전송하게 된다. 그림 4에는 클라이언트가 서버에 접속해서 MySQL에 저장된 측정값을 가져오는 과정을 나타냈다.

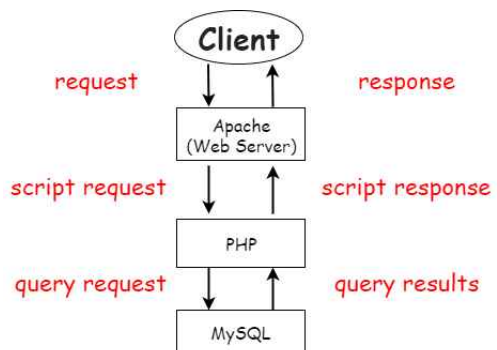


그림 4. APM 구동 원리  
Fig. 4. APM Driving principle

### 3-6 소켓 통신(Socket)

기존 HTTP(hypertext Transfer Protocol)의 경우 클라이언트

가 요청을 보낼 때만 서버가 응답하는 형태를 띠기 때문에 서버로부터 Client가 응답을 받은 후 연결이 끊기게 된다. 필요한 경우에만 접근하여 사용할 때는 쉽지만 본 논문에서 구현하고자 하는 센서 제어를 위한 안드로이드 클라이언트와 라즈베리 파이 서버와의 통신은 실시간 통신이 필요하므로 단 방향적 통신인 HTTP만을 사용하기에 무리가 있다. 본 논문에서 구현한 서버와 클라이언트의 통신은 TCP/IP(Transmission Control Protocol/Internet Protocol) 기반 서버를 구축했다. 그림 5에는 본 논문에서 구축한 클라이언트 서버의 소켓 통신 구조도를 나타냈다.

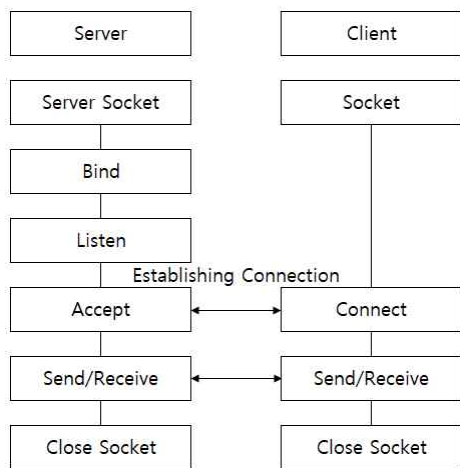


그림 5. 소켓 통신 구조도  
Fig. 5. Socket Framework

서버는 서버소켓(Socket)을 만들고 소켓에 주소를 할당하고 클라이언트로부터 연결요청 기다린다. 클라이언트로부터 연결요청이 오고 허용을 하게 되면 이때부터 실시간 데이터 송수신이 이루어지게 되고 송수신이 완료되면 소켓이 종료되게 된다.

#### IV. 안드로이드 클라이언트 앱

안드로이드 클라이언트 앱은 5개로 구성된다. 첫 화면은 최근 측정값들을 보여주고, 데이터 목록을 보여주는 화면으로 이동하는 버튼, 센서 제어를 위해 서버와 연결하는 화면으로 이동하는 버튼, 그래프 차트를 보여주는 화면으로 이동하는 버튼, PHP 주소 이름을 바꾸는 다이얼로그를 실행시키는 버튼이 포함되어 있다.

##### 4-1 메인화면

사용자가 SCI 안드로이드 앱을 실행하게 되면 인터넷을 통하여 라즈베리 파이에 접속하게 되고, 라즈베리 파이의 데이터베이스 MySQL에서 최근 데이터에 저장된 기록 날짜, 인덱스, 온도 수치, 미세먼지 수치, 이상 여부에 대한 데이터를 내려받

아 화면에 보이게 된다. 이때 데이터에는 라즈베리 파이에 연결된 센서들이 측정한 가장 최신의 값의 정보만 받게 된다.

PHP에 접속하기 위해 java.net 클래스인 HttpURLConnection을 사용하였고, 이 클래스를 통하여 받은 데이터는 JSON 형식으로 되어있어 org.json에서 제공하는 JSONArray, JSONException, JSONObject 등의 매서드를 사용하여 문자열 형태로 변경 사용하였다.

그림 6에는 안드로이드 클라이언트 앱의 메인화면을 나타냈다.

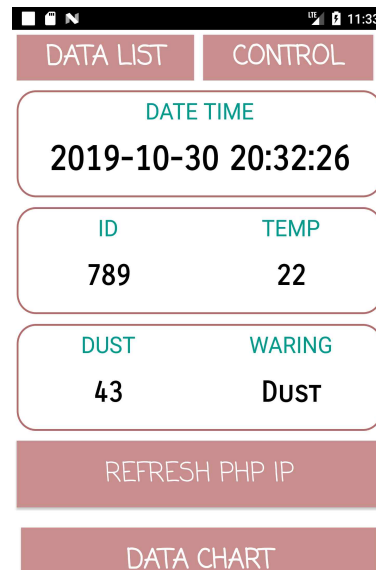


그림 6. 메인화면  
Fig. 6. Main Activity

메인화면에는 최신 기록을 보여줄 뿐만 아니라, 데이터 목록을 보여주는 화면으로 이동할 버튼, 그래프 차트를 보여주는 화면으로 이동할 버튼, 센서를 제어하기 위해 서버와 연결하기 위해 IP와 Port를 입력하는 화면으로 이동할 버튼 등으로 구성되어 있다. 메인화면이 실행되면 라즈베리 파이에 자료를 요청하게 되고, 라즈베리 파이는 MySQL에 저장된 측정값들을 JSON으로 변환하여 보내주고 앱은 비동기 통신을 통해 값을 받아 화면에 출력한다. 안드로이드와 라즈베리 파이에서 자료를 주고 받는 소스 코드를 표 6과 표 7에 나타냈다. 변수선언과 예외처리와 같은 부분은 모두 제거하고 실제 실행에 필요한 코드만 나타냈다.

표 6. 메인 액티비티 소스코드

Table 6. Main Activity Source Code

```

public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        new MyAsyncTask().execute("http://192.168.0.20/list.php");
    }

    private class MyAsyncTask extends AsyncTask<String, Integer, String> {

```

```
protected String doInBackground(String... params) {
    URL url = new URL(params[0]);
    HttpURLConnection conn
        = (HttpURLConnection) url.openConnection();
    BufferedReader in = new BufferedReader(
        new InputStreamReader(url.openStream(), "utf-8"));
    while ((buf = in.readLine()) != null) {
        sb.append(buf);
    }
}

protected void onPostExecute(String str) {
    JSONObject jsonObj = new JSONObject(str);
    JSONArray jsonArray = jsonObj.getJSONArray("runners");
    JSONObject json = null;

    for (int i = 0; i < jsonArray.length(); i++) {
        json = jsonArray.getJSONObject(i);
        if (json != null) {
            m_arList.add(new DTO(
                json.getInt("no")
                , json.getString("date")
                , json.getString("temp")
                , json.getString("soil")
                , json.getString("dust")
                , json.getString("warning")
            ));
        }
    }
    adapter.notifyDataSetChanged();
}
}
```

표 7. PHP 소스코드  
Table 6. PHP Source Code

```
<?php
header("Content-Type: text/html; charset=UTF-8");
$conn = mysqli_connect("localhost", "ID", "PW", "DB");

$sql = "SELECT * FROM tdust ORDER BY _id DESC LIMIT 1;";
$result = mysqli_query($conn, $sql);

$runners = array();
while( $row = mysqli_fetch_array($result) ){
    $arr = array(
        'no' => $row['no']
        , 'date' => $row['date']
        , 'temp' => $row['temp']
        , 'soil' => $row['soil']
        , 'dust' => $row['dust']
        , 'warning' => $row['warning']
    );
    array_push($runners, $arr);
}

echo json_encode(array('runners'=>$runners),
    JSON_PRETTY_PRINT|JSON_UNESCAPED_UNICODE);
mysqli_close($conn);
?>
```

4-2 데이터 목록 보기

메인화면에서 DATA LIST 버튼을 클릭하게 되면 그림 7과

같은 화면이 나타난다.

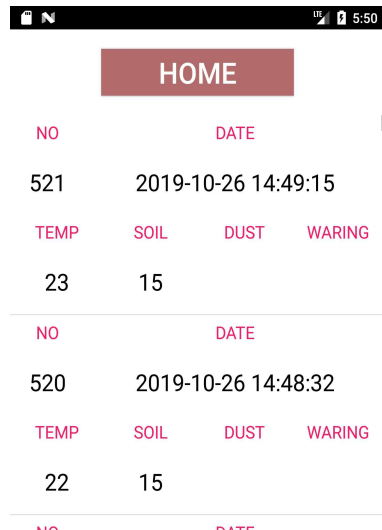


그림 7. 데이터 리스트 액티비티  
Fig. 7. Data List Activity

여기에는 측정된 날짜와 온도, 미세먼지 측정값 등을 보여준다. HOME 버튼을 클릭하면 메인화면으로 이동하고, 그 아래로는 3분 단위로 측정된 센서 측정값들을 리스트뷰를 통하여 보여준다. 데이터 리스트 액티비티는 메인 액티비티와 거의 비슷한 코드이다. 안드로이드에서는 TextView 대신에 ListView 위젯을 사용하여 20의 값을 리스트로 보여주고, PHP에서는 Sql 명령 “SELECT \* FROM tdust ORDER BY \_id DESC LIMIT 1;”에서 1 대신 20으로만 수정하면 된다.

4-3 데이터 차트 보기

메인화면에서 DATA CHART 버튼을 클릭하면 그림 8과 같이 온도(Temp), 습도(Humidity), 미세먼지(PM1.0), 미세먼지(PM2.5) 데이터 대한 변화를 꺾은 선 그래프로 보여준다.



그림 8. 데이터 차트  
Fig. 8. Data Chart

이 부분은 ThingSpeak 클라우드 서비스를 이용하였고, 이곳에서는 저장된 값으로 차트를 생성해서 웹브라우저에 출력해준다. 본 논문에서는 에셋(assets)에 ThinkSpeak의 차트를 iframe으로 출력해주는 HTML코드를 넣어놓고, 안드로이드 액티비티에서는 저장된 HTML을 호출하여 웹뷰(Webview)에 출력하도록 해주면 그림 8과 같은 결과가 나타난다. 표 8에는 assets에 저장된 HTML 소스코드를 표 9에는 이를 출력하는 액티비티 소스 코드를 나타냈다.

표 8. Temp.HTML 소스코드

Table 8. Temp.HTML Source Code

```
<html>
<head></head>
<body>
<iframe width="450" height="260"
src="https://thingspeak.com/channels/894399/charts/.."></iframe>
</body>
</html>
```

표 9. DataChart 액티비티 소스코드

Table 9. DataChart Activity Source Code

```
public class DataChart extends AppCompatActivity {
    WebView wvTemp, wvHumi, wvpm10, wvpm25;

    protected void onResume() {
        super.onResume();
        wvTemp.loadUrl("file:///android_asset/Temp.html");
        wvHumi.loadUrl("file:///android_asset/Humidity.html");
        wvpm10.loadUrl("file:///android_asset/PM1_0_Chart.html");
        wvpm25.loadUrl("file:///android_asset/PM2_5_Chart.html");
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        wvTemp = (WebView) findViewById(R.id.tempChart);
        wvTemp.getSettings().setJavaScriptEnabled(true);
        wvTemp.loadUrl("file:///android_asset/Temp.html");

        wvHumi = (WebView) findViewById(R.id.humiChart);
        wvHumi.getSettings().setJavaScriptEnabled(true);
        wvHumi.loadUrl("file:///android_asset/Humidity.html");

        wvpm10 = (WebView) findViewById(R.id.pm1_0Chart);
        wvpm10.getSettings().setJavaScriptEnabled(true);
        wvpm10.loadUrl("file:///android_asset/PM1_0_Chart.html");

        wvpm25 = (WebView) findViewById(R.id.pm2_5Chart);
        wvpm25.getSettings().setJavaScriptEnabled(true);
        wvpm25.loadUrl("file:///android_asset/PM2_5_Chart.html");
    }
}
```

4-4 원격 제어

메인화면에서 CONTROL 버튼을 누르게 되면 그림 9와 같은 화면이 나타난다.



그림 9. IP와 Port 입력 액티비티  
Fig. 9. IP and Port Insert Activity

이 화면은 안드로이드 앱이 라즈베리 파이에 접속하여 원격으로 제어할 때 사용하는 것으로 라즈베리 파이의 IP 주소와 Port를 입력하여 접속한다. IP와 Port는 처음 한 번만 입력하면 다음부터는 SharedPreferences 클래스를 이용하여 저장된 값을 사용하게 된다.

그림 10은 라즈베리 파이 서버에 접속하여 부착된 장치 중 LED, Buzzer, Fan, Servomotor의 현재 상태를 보여주고, 그들 각각을 수동으로 켜고 끌 수 있는 액티비티를 보여준다.

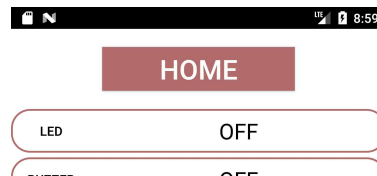


그림 10. 서버 통신 액티비티  
Fig. 10. Server Communication Activity

각각의 버튼을 누르면 해당 장치가 켜지고, 다시 누르면 꺼지는 방식으로 구현하였다.

4-5 FCM을 통한 경고 알람

라즈베리 파이 서버에서는 미세먼지와 온도 센서들이 측정 한 값이 임계치를 이탈하게 되면 해당 값들을 데이터베이스의 warning 필드에 저장되고 임계치 이탈 시 Fan과 Servomotor가 작동하고 경고상황을 사용자에게 알리게 된다. 사용자에게 알리기 위해 이상 FCM(Firebase Cloud Messaging)을 이용하였고, 안드로이드 앱은 FCM 메시지를 받아 경고 알람을 울리게 된다. 안드로이드 앱을 실행하고 있지 않은 상태에서도 미세먼지나 온도에 대한 값에 대한 이상이 발생할 때 알람을 받음으로 사용자가 작동환경에 이상이 발생했음을 알 수 있다. 그림 11은 알람을 받았을 때의 스마트폰 화면을 보여준다.

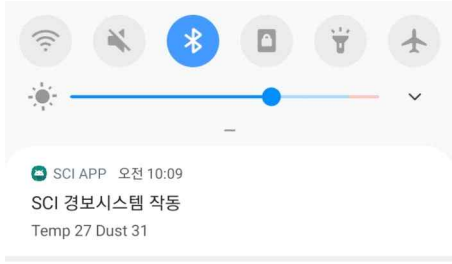


그림 11. FCM 알람  
Fig. 11. FCM Message

## V. 결론

본 논문을 통해 대기오염 및 온도 변화로 인해 농작물 성장 장애를 해소하기 위해 사람의 감각으로 점검하기 어려운 미세먼지, 온도 등의 중요 요소들을 센서 모듈을 통해 측정하여 라즈베리 파이의 MySQL과 ThingSpeak에 저장하고, 측정값에 이상이 발생하면 자동으로 환기 시스템을 동작하고 알람과 경고를 발생하여 농작물 관리를 할 수 있도록 하였다. 관리자는 외부에서 웹을 통하여 접속하여 측정값들을 확인할 수도 있고, 원격으로 환기 시스템을 작동할 수 있도록 있도록 하였다. 이를 통하여 공기 질을 관리하여 농작물 성장에 도움을 줄 수 있는 연구를 하였다. 인터넷 속도가 발달하고 스마트폰 보급률이 높은 대한민국에서 특히 본 논문에서 제안하는 시스템은 보급하기 수월할 것으로 판단되고 이를 통해 농작물 관리에 있어서 편리성과 효율을 높이는데 기대할 수 있다. 더 나아가 센서와 시설을 사용 규모에 맞게 재설계하여 적용한다면 환경에 민감한 의료, 건강, 유아, 노인 등의 분야에서도 사용할 수 있다.

## 참고문헌

[1] Trade Times. Agricultural IoT, we can save agriculture [Internet]. Available: <http://www.tradetimes.co.kr/news/articleView.html?idxno=624>

[2] e-National indicators, Farm and Farm Population [Internet]. Available: [http://www.index.go.kr/potal/main/EachDtlPageDetail.do?idx\\_cd=2745](http://www.index.go.kr/potal/main/EachDtlPageDetail.do?idx_cd=2745)

[3] I. S. Kim, K. R. Kim, H. C. Kim, M. T. Seo, K. S. Kim, and M. S. Ko, "A Study on an ICT-based System for Safe Management of Agricultural Facilities for Farmers' Safety Activities," *Journal of the Ergonomics Society of Korea*, Vol. 37, No. 4, pp. 489-502, 2018.

[4] T. H. Lee and M. K. Mun, "Design and Development of Mobile Air-Quality Measurement System based on IoT

Platform," *The Journal of the Korean Institute of Next Generation Computing*, Vol. 12, No. 3, pp. 75-84, 2016.

[5] Ministry of Agriculture, Food and Rural Affairs, A Study of Smart Farm Operation Analysis and Development Direction [Internet]. Available: [http://www.prism.go.kr/homepage/researchCommon/downloadResearchAttachFile.do?sessionId=D56668E6B9D4CB3C0EFE810A363F628A.node02?work\\_key=001&file\\_type=CPR&seq\\_no=001&pdf\\_conv\\_yn=N&research\\_id=1543000-201600035](http://www.prism.go.kr/homepage/researchCommon/downloadResearchAttachFile.do?sessionId=D56668E6B9D4CB3C0EFE810A363F628A.node02?work_key=001&file_type=CPR&seq_no=001&pdf_conv_yn=N&research_id=1543000-201600035)

[6] Korea Rural Economic Institute. Agricultural Vision 2019 [Internet]. pp. 272-279. Available: <http://library.krei.re.kr/pyxis-api/1/digital-files/d4b1612f-1649-44b8-b92e-ab0a1bb878e9>

[7] Air quality guidelines – global update 2005. World Health Organization [Internet]. Available: [https://www.who.int/phe/health\\_topics/outdoorair/outdoorair\\_aqg/en/](https://www.who.int/phe/health_topics/outdoorair/outdoorair_aqg/en/)



**이평화(Pyeong-hwa Lee)**

2015년~현 재: 전주대학교 스마트미디어학과 재학중

※ 관심분야 : 사물인터넷(IoT), 영상처리, 디지털콘텐츠, 멀티미디어, 모바일 프로그래밍 등



**민정익(Jeong\_Ik Min)**

1989년 : 한국과학기술원 (공학석사)

2001년 : 한국과학기술원 (공학박사)

1992년~2000년: LG CNS

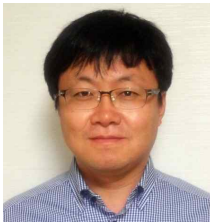
2000년~2010년: 한국 IBM, SAP

2011년~2014년: KT

2014년~현 재: 전주대학교 스마트미디어학과 교수

2017년~현 재: 전주대학교 대학원 스마트 Agro\_ICT 융합학과 교수

※ 관심분야 : 스마트팜, 인공지능, IOT, 블록체인, ERP, SCM, 정보전략



**송주환(Ju-Whan Song)**

1997년 : 전북대학교 대학원 (이학석사)

2003년 : 전북대학교 대학원 (이학박사-컴퓨터그래픽스)

2006년~현 재: 전주대학교 스마트미디어학과 교수

2019년~현 재: 전주대학교 대학원 스마트 Agro\_ICT 융합학과 교수

※ 관심분야 : 컴퓨터그래픽스, 영상처리, 사물인터넷(IoT), 디지털콘텐츠, 멀티미디어, 모바일 프로그래밍 등