

# IoT 기반 대기 오염 관련 스마트 생활 정보 서비스 구현

## Implementation of Smart Information Services on Air Pollution based on IoT

고가영, 창병모  
숙명여자대학교 컴퓨터과학과

Ga-Young Koh(dianakoh@sookmyung.ac.kr), Byeong-Mo Chang(chang@sookmyung.ac.kr)

### 요약

사물인터넷(Internet of Things:IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술로 스마트 홈, 스마트 공장 등 다양한 분야에서 활용되고 있다. 최근 미세먼지 등 대기 오염 정보에 대한 관심이 높아짐에 따라 IoT 기술을 기반으로 날씨 정보, 대기 오염 정보를 제공하는 앱이 개발되고 있다. 본 연구에서는 IoT 기술을 활용하여 시각적인 대기 오염 및 날씨 정보를 제공하고 이를 활용하여 가전 자동화 등의 유용한 서비스를 개발함을 목표로 한다. 이를 위해 본 연구에서는 IoT 기술을 기반으로 하여 대기 오염 정보를 이용한 가전 자동화, 메시지 및 알람, 시각적인 대기 오염 정보 및 날씨 정보를 제공하는 스마트 서비스를 설계, 구현하였다.

■ 중심어 : | 사물인터넷 | 대기 오염 | 날씨 | SmartThings |

### Abstract

The Internet of Things (IoT) is a technology that connects to the Internet with built-in sensors and communication functions on various objects and has been used in various fields such as smart home and smart factory. Recently, as interest in air pollution information such as fine dust is increasing, the apps for providing weather information and air pollution information based on IoT technology are being developed. In this study, we aims to provide visual air pollution and weather information and develop useful services such as appliance automation using this information. for this study, we designed and implemented Smart Services that provide home automation, messaging and alarm, visual air pollution and weather information using air pollution and weather information based on IoT.

■ keyword : | IoT | Air Pollution | Weather | SmartThings |

## I. 서론

사물인터넷(Internet of Things:IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술로 최근에 다양한 분야에서 활용되고 있다.

특히 스마트 홈을 위한 IoT 플랫폼으로 개발된 SmartThings는 스마트 홈을 비롯한 다양한 분야에서 응용되고 있으며, 최근에는 SmartThings를 기반으로 하여 날씨 정보와 같은 생활 정보 서비스를 제공하는 앱이 개발되고 있다[1-3]. 스마트 홈 서비스 관련 연구

\* 본 연구는 기후변화특성화대학원 지원사업("한반도 기후변화 모니터링과 영향평가를 통한 적응대책 연구") 지원에 의해 수행되었음.

접수일자 : 2019년 09월 10일  
수정일자 : 2019년 10월 21일

심사완료일 : 2019년 10월 21일  
교신저자 : 창병모, e-mail : chang@sookmyung.ac.kr

에서는 사용자 설정에 따른 시각적인 맞춤형 대기오염 및 날씨 정보 제공과 이러한 정보를 사용한 알림서비스 및 가전 자동화가 필요함을 설명하였다[4].

시각적인 대기오염 정보를 제공하는 것과 관련된 기존 연구에서는 아두이노를 기반으로 공기질 센서 데이터를 수집하여 그래프 형태로 제공하였다[5]. 또한, IoT 플랫폼을 활용하여 아이콘 및 그래프를 사용한 시각적인 날씨 및 대기 오염 정보를 제공하고자 하였다[6].

본 연구에서는 미세먼지 등 대기 오염 정보에 대한 관심이 높아져가고 있는 상황에서 IoT 기술을 활용하여 대기 오염 및 날씨 정보를 활용한 스마트 서비스를 제공함을 목표로 한다. 이를 위해 SmartThings를 기반으로 하여 대기 오염 관련 스마트 생활 정보 서비스를 제공하고, 대기 오염 정보를 바탕으로 공기청정기 자동 제어 서비스를 설계 구현하였다.

본 연구에서 구현한 대기 오염 관련 스마트 생활 정보 서비스는 시각적 대기 오염 정보 서비스, 대기 오염 정보 자동 알림 서비스, 공기청정기 자동 제어 서비스 등이다. 시각적 대기 오염 정보 서비스는 주요 대기 오염 물질의 현재 수치를 시각적으로 보여주고 수치 변화를 그래프 형태로 표시한다. 대기 오염 정보 자동 알림 서비스는 사용자에게 현재 미세먼지 수치와 유의 사항 등을 전달하고, 미세먼지가 나쁨일 경우 알람을 통해 즉각적으로 알 수 있게 하는 기능이다. 또한 미세먼지 수치를 이용하여 공기청정기를 자동으로 제어하는 서비스를 제공함으로써 항상 좋은 실내 공기질을 유지할 수 있도록 한다. 기타 날씨 관련 서비스는 현재 날씨를 보여줄 뿐 아니라 이 정보에 따라 그날의 옷차림을 추천하고 이를 시각적으로 보여준다.

본 논문의 구성은 다음과 같다. 2장 배경에서는 SmartThings에 대해 소개한다. 3장에서는 관련 연구들을 설명하고 비교한다. 4장에서는 본 논문에서 구현한 대기 오염 정보 관련 및 날씨 정보 관련 주요 기능을 기술한다. 5장에서는 이들의 구현에 대해서 기술한다. 6장에서는 실험 결과에 대해 기술하고 7장에서 결론을 맺는다.

## II. 배경

SmartThings는 스마트 홈을 위한 사물인터넷(IoT) 플랫폼으로서 [그림 1]과 같이 SmartThings 클라우드와 연결된 허브를 통하여 IoT 디바이스들을 연결하고 관리한다[7]. 또한 SmartThings 모바일 앱은 SmartThings 클라우드와 연결되어 IoT 디바이스와 스마트 앱을 관리하는 모바일 앱이다[8].

SmartThings 클라우드 상에서 실행되는 앱을 스마트 앱이라고 하며 디바이스 핸들러는 SmartThings 클라우드에 위치하며 물리적 디바이스에 대한 가상 표현이다. 각각에 대한 설명은 다음과 같다.

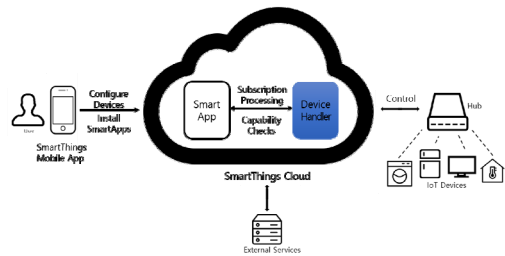


그림 1. SmartThings 플랫폼 구조

### (1) 스마트 앱(Smart App)

스마트 앱은 IoT 디바이스의 기능(capability)을 사용하여 스마트 서비스를 제공하는 SmartThings 클라우드 상에서 실행되는 작은 앱으로 Groovy 언어로 작성한다. 사용자는 SmartThings 모바일 앱을 통해서 스마트 앱을 클라우드에 설치할 수 있으며 클라우드와 연결된 허브를 이용하여 각 디바이스에 접근할 수 있다.

스마트 앱의 구조는 [그림 2]와 같이 definition, preferences, predefined callbacks, event handler로 구성된다[8]. 또한, 이 스마트 앱은 사용자의 움직임에 따라 불을 켜고 끄는 앱으로서 사용자의 움직임이 있을 경우 불을 켜고 움직임이 사라지면 불을 끈다.

- definition: 스마트 앱의 이름 및 앱의 세부 정보들을 정의한다.
- preferences: 스마트 앱의 설치 및 업데이트 시의 화면 페이지를 정의하는 부분으로서 스마트

앱에서 사용할 디바이스를 선택할 수 있다.

- predefined callbacks: installed(), updated(), initialize()를 포함하며 스마트 앱의 설치 및 업데이트 시 스마트 앱에서 사용되는 이벤트 핸들러 등록(event subscription)을 설정한다.
- eventHandler: 스마트 앱에서 사용되는 이벤트 핸들러 및 다른 함수들을 구현하는 부분이다.

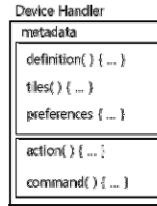


그림 3. 디바이스 핸들러 구조

```

1 // definition
2 definition() {
3   name: "Control the light according to movement",
4   author: "your name",
5   description: "Turn on/off the light according to movement",
6   category: "Convenience"
7 }
8 // preferences
9 preferences {
10  section("Select devices") {
11    input "motion1", "capability.motionSensor"
12    input "light1", "capability.switch"
13  }
14 }
15 // predefined callbacks
16 def installed() {
17   initialize()
18 }
19 def updated() {
20   unsubscribe()
21   initialize()
22 }
23 def initialize() {
24   subscribe(motion1, "motion", motionHandler)
25 }
26 // event handler
27 def motionHandler(evt) {
28   if (evt.value == "active") {
29     light1.on()
30   } else if (evt.value == "inactive") {
31     light1.off()
32   }
33 }
34 }
35 }
  
```

그림 2. 스마트 앱 코드

## (2) 디바이스 핸들러(Device Handler)

디바이스 핸들러는 SmartThings 클라우드에 위치하며 물리적 디바이스에 대한 가상 표현으로서 논리적 디바이스를 만들어준다. SmartThings 모바일 앱에서는 사용자가 정의한 디바이스 핸들러 타입으로 디바이스를 생성 및 설정할 수 있으며, 스마트 앱에서는 디바이스 핸들러에서 정의된 기능(capability)을 이용하여 입력 디바이스를 선택하여 사용할 수 있다.

디바이스 핸들러의 일반적인 구조는 [그림 3]과 같으며 [그림 4]는 [그림 5]와 같은 간단한 콘센트를 위한 디바이스 핸들러의 예이다[8].

- definition: [그림 4]의 definition 부분처럼 디바이스의 이름은 "TestDevice"로, capability는 "Switch"로, attribute는 "power"로 정의하고 이 디바이스에서 사용되는 command는 "Handler"로 선언한다.
- tiles: [그림 5]와 같이 SmartThings 모바일 앱에서 디바이스를 시각적으로 보여주는 기능을 한다. 디바이스의 capability, attribute를 타일 형태로 나타내어 디바이스 정보들을 쉽게 확인할 수 있다.
- preferences: [그림 4]의 preferences 코드처럼 디바이스의 특정 property를 설정할 수 있다.
- action 및 command: [그림 4]처럼 디바이스의 action인 on(), off()를 정의하고 디바이스에서 사용되는 command인 Handler()를 정의한다. 여기서는 외부 API 정보를 가져오기 위해서 http request 라이브러리를 이용한다.

```

1 //metadata - definition, tiles, preferences
2 metadata {
3   definition (name: "TestDevice", namespace: "yournamespace", author: "your name") {
4     capability "switch"
5     attribute "power", "string"
6     command "Handler"
7   }
8   tiles {
9     standardTile("switchTile", "device.switch", width: 2, height: 2) {
10      state "off", label: "${name}", action: "switch.on",
11      icon: "st.switches.switch.off", backgroundColor: "#ffffff",
12      state "on", label: "${name}", action: "switch.off",
13      icon: "st.switches.switch.on", backgroundColor: "#00a0dc"
14    }
15    valueTile("powerTile", "device.power", decoration: "flat") {
16      state "power", label: "${currentValue} W"
17    }
18    main "switchTile"
19    details("switchTile", "powerTile")
20  }
21  preferences {
22    input "accessKey", "text", type: "password", title: "The API key"
23  }
24 }
25 def installed() {
26   runEvery15Minutes(Handler)
27 }
28 def uninstalled() {
29   uncancel()
30 }
31 def update() {
32   unsubscribe()
33   runEvery15Minutes(Handler)
34 }
35 // actions and command
36 def on() {
37   sendEvent(name: "switch", value: "on")
38 }
39 def off() {
40   sendEvent(name: "switch", value: "off")
41 }
42 def Handler() {
43   def url = "https://api.thingspeak.com/channels/${accessKey}/feeds.json?result=1"
44   try {
45     httpGet(url) { resp ->
46       sendEvent(name: "power", value: "${resp.data.feeds.field1[0]}")
47     }
48   } catch(e) { log.error "error: ${e}"
49 }
50 }
  
```

그림 4. 디바이스 핸들러 코드

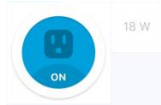


그림 5. TestDevice 화면

### III. 관련 연구

본 연구와 관련된 날씨 정보 서비스에 대한 연구는 다음과 같다.

#### (1) SmartWeather Station For Korea

SmartWeather Station For Korea[1]는 SmartThings에서 제공하는 SmartWeather Station Tile[2] 디바이스 핸들러를 기반으로 만들어졌다. SmartWeather Station Tile은 zip코드를 이용하여 날씨 정보를 받아 tile형태로 보여준다. SmartWeather Station For Korea는 날씨 정보와 함께 Air Korea API 정보를 이용하여 국내 대기 오염 수치 및 지수를 tile 형태로 보여주도록 확장한 서비스이다.

#### (2) Severe Weather Alert

Severe Weather Alert[3]는 SmartThings Public으로 제공하는 스마트 앱으로서 zip 코드 또는 설치된 SmartThings 허브의 기본 위치 정보(위도 및 경도)를 사용하여 getTWCAlerts 라이브러리를 통해 alert 정보를 가져와 사용자에게 기상 주의보 및 경보를 SMS 메시지로 보내주는 기능을 한다.

#### (3) CO2 Vent

CO2 Vent[9]는 SmartThings Community 스마트 앱으로서 CO2 센서를 이용하여 환풍기(vent)를 자동화하는 기능을 한다. 사용자가 기준이 되는 CO2 값을 정하도록 하고 그 기준 값보다 CO2값이 높을 때 환풍기가 꺼져 있는 상태라면 환풍기의 스위치가 켜지도록 하였고 반대로 기준 값보다 CO2 센서의 값이 낮을 때 환풍기가 켜져 있는 상태라면 환풍기의 스위치가 꺼지도록 구현한 앱이다.

#### (4) 원기날씨

원기날씨[10]는 안드로이드 및 ios에서 일반적으로 사용되는 대기 오염 및 날씨 정보 앱으로서 대기 오염 정보로 6개의 주요 대기 오염 물질의 현재 상태와 통합 대기 상태를 알려주고, 날씨 정보로 기온, 날씨 아이콘, 강수량, 습도 등을 알려준다.

#### (5) WTHRD

WTHRD[11]는 ios에서 사용되는 날씨 정보 앱으로서 날씨 정보로 현재 기온과 습도를 알려주고 이에 따라 그림으로 나타낸 옷차림 정보를 보여준다. 대기 오염 정보로는 통합 대기 지수를 알려준다.

본 연구에서 구현한 서비스는 SmartThings 플랫폼의 스마트 앱 및 디바이스 핸들러를 통해 미세먼지 값을 이용하여 공기청정기를 자동화하고 메시지 및 알람 기능을 제공한다. 또한, 날씨 정보 및 대기 오염 정보를 시각적으로 보여준다.

기존 연구에서는 날씨 정보, 대기 오염 정보 중 일부 정보를 보여주는 경우가 많았고[10][11], 일부 연구에서는 날씨 정보와 대기 오염 정보를 통합하여 보여주려고 하였다[1][2]. 본 연구에서는 부분적인 정보를 통합하여 보여주는 것에서 더 확장하여 대기 오염 물질 변화 추이를 보여주는 그래프나 옷차림 정보, 아이콘 정보 등의 시각적인 정보를 제공하고 사용자의 편리성을 높이기 위해 이를 하나의 화면에서 볼 수 있도록 구성하였다.

또한, IoT 기술을 적용하여 데이터 값에 따라 공기청정기를 자동화하는 기능을 구현하였다. 디바이스의 자동 모드인 경우 부분적으로 자동화 기능을 제공하나 이는 사용자가 수동으로 전원을 켜 후 설정을 통해 동작되며 내부 센서 값에 따라 동작되는데 전원이 꺼져있을 경우 동작하지 않는다. 본 연구에서는 외부 미세먼지 값을 사용하여 사용자의 수동적인 조작 없이도 공기청정기가 자동으로 동작할 수 있도록 하였다.

표 1. 관련 연구 비교

	디바이스 자동화	날씨 정보	대기 정보	옷차림 정보	그래프 정보	알림 기능
(1)		○	○			
(2)						○
(3)	○					
(4)		○	○			○
(5)		△	△	○		
디바이스 자동모드	△					
본 연구	○	○	○	○	○	○

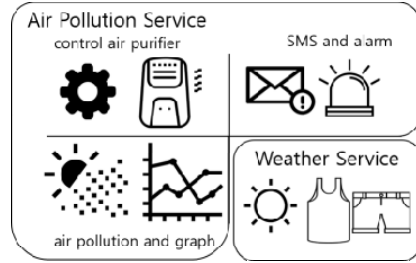


그림 6. 주요 서비스

## IV. 주요 서비스 및 구조

### 1. 주요 서비스

대기 오염 정보를 활용한 생활 정보 서비스로 [그림 6]과 같이 시각적 대기 오염 정보 서비스, 대기 오염 정보 자동 알림 서비스, 공기청정기 자동 제어 등을 제공한다[12].

- 시각적 대기 오염 정보 서비스: 미세먼지 수치에 따라 마스크 사용 여부를 아이콘으로 표시하며, 6개의 대기 오염 물질 수치 변화를 그래프로 보여준다.
- 대기 오염 정보 알림 서비스: 사용자가 설정한 시간에 따라 현재 미세먼지 수치 및 상태, 외출 시 주의 사항 등을 SMS 메시지로 알려준다. 또한, 미세먼지 수치가 나쁨 이상일 때, 주기적으로 알람을 올린다.
- 공기청정기 자동 제어 서비스: 대기 오염 정보 중 미세먼지 상태에 따라 공기청정기와 같은 가전을 자동으로 on/off 한다. 또한, 미세먼지 수치와 공기청정기의 on/off 상태를 웹 그래프로 제공한다.

본 논문에서는 기타 서비스로 날씨 정보를 활용한 생활 정보 서비스로 현재 날씨 상태 정보와 기온에 따른 옷차림 추천 정보를 제공한다.

- 날씨 상태 정보 및 옷차림 추천 정보: 날씨 상태 아이콘, 체감 온도, 강수 확률, 습도, 자외선 지수 등을 보여주고 기온에 따른 옷차림 추천을 해 준다.

### 2. 전체 시스템 구조

본 연구에서는 SmartThings를 기반으로 날씨 및 대기 오염 정보를 활용한 스마트 서비스를 스마트 앱과 디바이스 핸들러 형태로 구현하였다.

스마트 앱은 미세먼지 정보에 따라 공기청정기를 자동으로 제어하며 SMS 메시지 및 알람 기능을 제공한다.

또한 디바이스 핸들러를 이용하여 물리적 디바이스의 기능들을 통합한 가상 디바이스를 개발하여 대기 오염 및 날씨 정보를 시각적으로 제공한다. 이 핸들러는 The Weather Company API와 Air Korea API를 통해서 날씨 및 대기 정보를 받아오며 추가 기능으로 통합 대기 수치 변화 그래프, 기온에 따른 옷차림 정보 등을 제공한다.

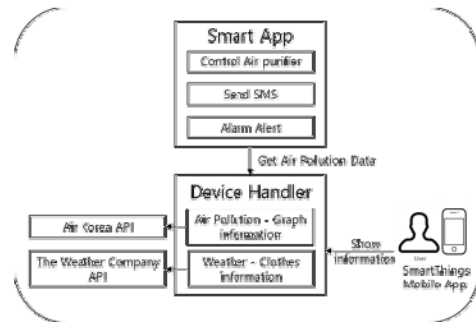


그림 7. 시스템 구조

## V. 구현

### 1. 시각적 대기 오염 정보 서비스

대기 오염 정보는 Air Korea API를 통해서 가져온

다. httpGet 라이브러리 함수를 이용하여 API 사이트의 url, parameter 등의 request 정보를 보내고 대기 오염 정보를 response로 받는다. 이 정보들을 [3]의 디바이스 핸들러를 기반으로 확장 구현하여 tiles를 통해 스마트앱에서 시각적으로 볼 수 있도록 구현하였다.

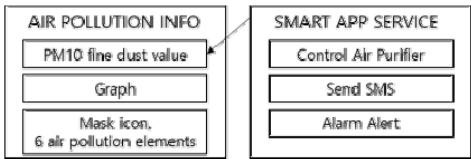


그림 8. 대기 오염 정보 앱의 프로그램 구조

기본적인 대기 오염 정보를 보여줄 뿐 아니라, [그림 9]와 같이 미세먼지 수치에 따른 마스크 사용 여부를 알려주고, 6개의 대기 오염 물질 수치 변화를 ThingSpeak 서버와 연동하여 그래프 형태로 구현하였다[1][12]. 이를 통해 현재 대기 상태뿐만 아니라 대기의 변화 추이를 확인할 수 있다.

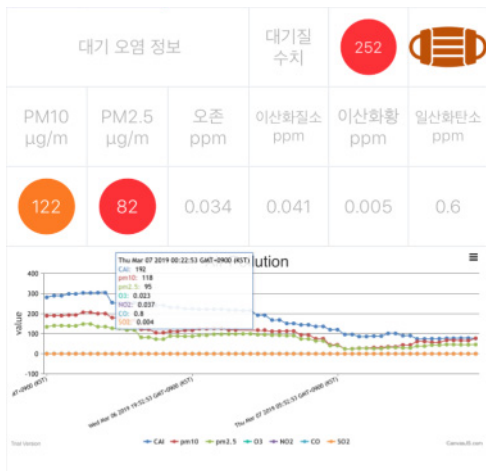


그림 9. 대기 오염 정보 타일

## 2. 대기 오염 정보 자동 알림 서비스

미세먼지 수치 및 상태 그리고 그에 따른 주의사항 정보를 [그림 10]과 같은 SMS 메시지 형태로 나타낸다. 이를 위해서 디바이스의 attribute 중 airQuality 값을 이용하였고 사용자가 설정한 시간에 메시지가 전달되도록 구현하였다. 또한 미세먼지 상태가 나쁨 이상일 경우에 알람이 주기적으로 울리도록 하였다.

(오늘) 오후 5:40

[국외발신]  
현재 외부 미세먼지 수치는 77로 보통 상태입니다.  
마스크를 준비해주세요.

alarm1 11:32 AM  
Alarm is Strobe  
Checking Weather And Air Pollution 11:32 AM  
Sent Strobe Command To Alarm1

그림 10. SMS 메시지와 알람

## 3. 공기청정기 자동 제어 서비스

SmartThings 허브에 연결된 공기청정기를 외부 미세먼지 수치 기준 혹은 사용자 설정 값에 따라 자동으로 제어하는 스마트 앱을 구현하였다. 이를 위해 디바이스 핸들러로 구현된 가상 디바이스를 통해 외부 미세먼지 수치를 받아 공기청정기를 제어하는 데 사용하였다. [그림 11]은 공기청정기 제어 알고리즘에 대한 순서도이다.

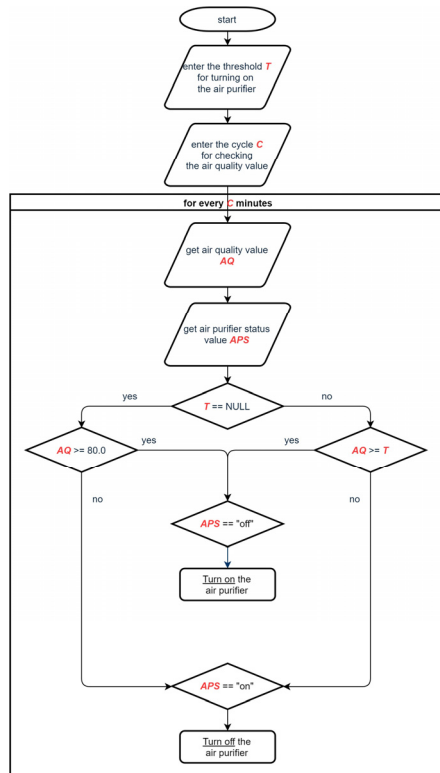


그림 11. 공기청정기 제어 알고리즘

공기청정기를 on/off 하는 기준 값(T)과 현재 외부 미세먼지 검사 주기 값(C: minutes)을 사용자에게 입력받고 C 주기마다 외부 미세먼지 값(AQ)와 현재 공기청정기의 상태 값(APS)를 확인하여 공기청정기를 자동으로 제어한다. T가 설정된 경우 T 값에 따라 T가 설정되지 않은 경우 대기의 나쁨 기준인 80.0을 기준으로 하여 공기청정기의 상태 값 APS가 "off"이면 공기청정기를 자동으로 켜다. [그림 12]는 사용자의 설정 값(T=40, C=10)에 따라 공기청정기가 실제로 동작하는 그림이다.



그림 12. 공기청정기 실제 동작

#### 4. 기타 날씨 정보 서비스

날씨 정보는 SmartThings 플랫폼과 연동된 The Weather Company API를 사용할 수 있으며 getTwcConditions 라이브러리 함수를 통해 날씨 정보를 가져온다[1]. 디바이스 핸들러에서 getTwcConditions 라이브러리를 사용하였고 디바이스 핸들러의 tiles를 통해 정보들을 SmartThings 모바일 앱에서 시각적으로 볼 수 있도록 구현하였다. 이 프로그램의 구조는 [그림 13]과 같다.

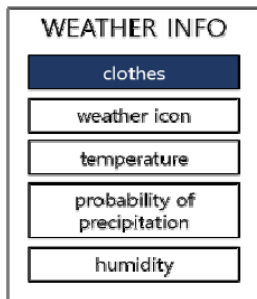


그림 13. 날씨 정보 앱의 프로그램 구조

특히, [그림 14]와 같이 체감 기온에 따른 옷차림 정보를 메인 타일로 구성하여 날씨 상태를 쉽게 파악할 수 있도록 하였다[1][12]. 세부 정보는 날씨 아이콘, 기온, 체감 기온, 강수 확률, 습도, 자외선 지수 등이다.

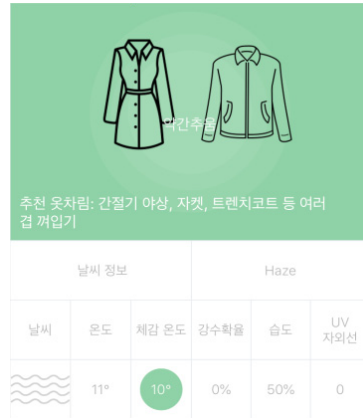


그림 14. 날씨 정보 타일

## VI. 실험 결과

본 연구에서는 사용자 설정 값에 따른 공기 청정기 제어 실험을 진행하였다. 사용자의 설정 값이 있을 때 외부 미세먼지 값에 따라 자동으로 공기청정기를 제어하는 실험을 진행하였다.

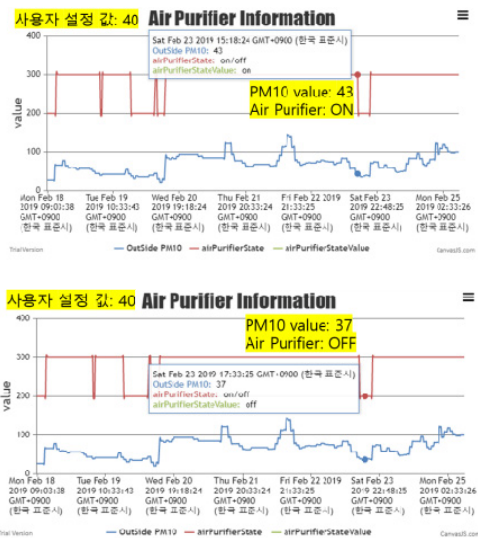


그림 15. 공기청정기 상태 그래프(위:ON /아래:OFF)



[그림 15]는 2019-02-18 ~ 2019-02-24의 외부 미세먼지 값과 공기청정기의 상태를 그래프로 나타낸 것이다. 그림과 같이 사용자의 설정 값이 40일 때 외부 미세먼지 값이 43이면 공기청정기가 ON 되어 있고 외부 미세먼지 값이 37이면 공기청정기가 OFF 되어 있는 것을 확인할 수 있다.

## VII. 결론 및 향후 연구

본 논문에서는 SmartThings를 기반으로 하여 대기 오염과 관련된 스마트 생활 정보 서비스를 스마트 앱과 디바이스 핸들러 형태로 구현하였다.

향후 미세먼지 수치를 이용하여 공기청정기를 켜고 끄는 기능에 더하여 작동 세기를 조절하고 실내, 외 비교를 통한 정확한 제어를 할 수 있도록 확장할 것이다. 뿐만 아니라 SmartThings 플랫폼과 연동되는 다른 가전 기기를 활용한 다양한 스마트 앱에 대해서 연구할 것이다.

### 참고 문헌

[1] WooBoong, Smart Weather Station for Korea, <http://github.com/WooBooung/BooungThings/blob/master/devicetypes/woobooung/smartweather-station-for-korea.src/smartweather-station-for-korea.groovy>, 2019.03

[2] Smart Weather Station Tile, <https://github.com/SmartThingsCommunity/SmartThingsPublic/blob/master/devicetypes/smartthings/smartweather-station-tile.src/smartweather-station-tile.groovy>, 2019.03

[3] SmartThingsCommunity, Severe Weather Alert, <https://github.com/SmartThingsCommunity/SmartThingsPublic/blob/master/smartapps/smartthings/severe-weather-alert.src/severe-weather-alert.groovy>, 2019.03

[4] 김민정, 신나리, 장수민, 최성규, 구유리, “사물인터넷 환경에서의 사용자 경험 향상을 위한 날씨정보 중심의 스마트홈 서비스 개발 연구,” 디자인융복합연구,

제16권, 제1호, pp.111-127, 2017.

[5] Giovanni B. Fioccola, Raffaele Sommese, Imma Tufano, Roberto Canonico, Giorgio Ventre, “Polluino: An Efficient Cloud-based Management of IoT Devices for Air Quality Monitoring,” 2016 IEEE 2nd Int. Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow(RTSI), 2016(9).

[6] 고가영, 창병모, “SmartThings 기반의 날씨 및 대기 오염 정보를 활용한 스마트 서비스 구현,” 한국정보과학회 학술발표논문집, pp.455-457, 2018(6).

[7] Y. Tian, N. Zhang, Y. H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, “Smartauth: User-centered authorization for the IoT,” 26th USENIX Security Symposium, 2017(8).

[8] SmartThings, SmartThings Classic Developer Documentation, 2019.

[9] SmartThingsCommunity, CO2 Vent, <https://github.com/SmartThingsCommunity/SmartThingsPublic/blob/master/smartapps/dianoga/co2-vent.src/co2-vent.groovy>, 2019.03.08.

[10] <https://apps.apple.com/kr/app/원기날씨-미세먼지-기상청-날씨/id118737186>, 2019.07.09

[11] <https://apps.apple.com/kr/app/wthrd-날씨와-옷/id1194408342>, 2019.07.09

[12] <https://thenounproject.com>, 2019.03.08.

### 저자 소개

고 가 영(Ga-Young Koh)

준희원



- 2018년 2월 : 숙명여자대학교 컴퓨터과학부(이학사)
- 2018년 3월 ~ 현재 : 숙명여자대학교 컴퓨터과학과(석사과정)

<관심분야> : 사물인터넷, 소프트웨어 분석



장 병 모(Byeong-Mo Chang)

정회원



- 1988년 2월 : 서울대학교 컴퓨터공학과(공학사)
- 1990년 2월 : KAIST 전산학과(공학석사)
- 1994년 2월 : KAIST 전산학과(공학박사)
- 1995년 3월 ~ 현재 : 숙명여자대학교

소프트웨어학부 교수

〈관심분야〉 : 사물인터넷, 프로그래밍 언어