



국민대학교  
전자정보통신대학  
컴퓨터공학부

# 캡스톤 디자인 I

## 종합설계 프로젝트


프로젝트 명	Mini BuT
팀 명	MBT
문서 제목	중간보고서

Version	1.3
Date	2019-04-14

팀원	오상화(조장)
	김수은
	민태준
	박재효
	박종민
지도교수	이 상 환 교수

### CONFIDENTIALITY/SECURITY WARNING


이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 “Mini BuT”을 수행하는 팀 “MBT”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “MBT”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## 문서 정보 / 수정 내역


Filename	중간보고서-Mini BuT.doc
원안작성자	오상화, 김수은, 민태준, 박재효, 박종민
수정작업자	

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-11	박종민	1.0	내용 추가	간단한 내용 약술
2019-04-12	박종민	1.1	내용 추가	간단한 내용 약술
2019-04-13	전원	1.2	내용 추가 및 수정	수행 내용 및 향후 추진계획 작성
2019-04-14	오상화, 김수은, 박재효, 박종민	1.3	내용 추가	수정 사항 작성

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## 목 차

1. 프로젝트 목표	4
2. 수행 내용 및 중간결과	5
2-1. 계획서 상의 연구내용	5
2-2. 수행내용	8
3. 수정된 연구내용 및 추진 방향	16
3-1. 수정사항	16
4. 향후 추진계획	18
4-1. 향후 계획의 세부 내용	18
5. 고충 및 건의사항	21

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

# 1 프로젝트 목표


## 1.1 목표

본 프로젝트는 1인 가구 및 소형가구를 타겟으로 하는, 실내 미세먼지 농도 관리를 실시간으로 도와주는 홈 IoT 플랫폼 개발을 목표로 한다.

- 1) 실내 환경 데이터와 실외 환경 데이터를 수집하여 적절한 실내 환경을 조성할 수 있도록 주변 환경 기기(IoT 창문, 공기청정기 등)를 통제하는 기능
- 2) 실내 미세먼지 상태를 LCD와 LED를 사용하여 사용자에게 알림 기능
- 3) 웹을 통해 실외에서도 실내 환경을 확인할 수 있고 연동 기기를 직접 제어하는 기능

이를 시연하기 위해 본체 제품과 함께 웹에서 연동 기기(창문, 공기청정기)를 조작하는 것을 웹 화면을 통하여 시뮬레이션하여 보여준다.

위 기능을 통하여 더 나은 실내 환경 개선을 하고자 한다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

서버는 Mini BuT본체에서 송신한 실내 미세먼지 측정 데이터와, KMU Crowd Sensor Cloud, 공공 API를 통해 지역성과 대표성이 확보되는 실시간 실외 미세먼지 데이터를 종합하여 실내·외 미세먼지 데이터를 얻고 이를 구축된 DB에 저장한다.

이러한 데이터들을 참고하여 서버는 Mini BuT 사용자의 실내 미세먼지 농도를 적절히 관리하기 위한 데이터 처리 알고리즘을 통해 제어 명령을 내리고, Mini BuT은 이를 수신하여 연동 기기를 제어한다.

또한 웹을 통하여 사용자는 현재 실내 환경과 연동 기기의 가동 여부를 실시간으로 알릴 수 있고, 사용자가 연동 기기의 전원을 직접 제어할 수도 있어 더욱 효과적으로 실내 환경 개선에 도움을 줄 수 있다.

자세한 개발 내용은 다음과 같다.

#### 2.1.1 라즈베리파이 및 각종 센서를 이용하여 제품의 하드웨어 구현

라즈베리파이에 미세먼지 센서를 탑재하고 LED와 LCD로 사용자가 쉽게 실내 상태를 확인할 수 있도록 한다. 미세먼지 센서는 SDS-011 모델로, PM10과 PM2.5를 모두 측정한다. LCD는 I2C를 이용해 제어하고 LED는 라즈베리파이의 GPIO를 통해 작동시킨다.


미세먼지 센서가 30초간 미세먼지를 실내를 측정하면, 남은 30초간 LCD로 미세먼지 상태를 표시한다. 동시에 LED는 색을 통해 실내 미세먼지 상태를 알린다. 미세먼지 농도가 좋음이면 파랑, 보통이면 초록, 나쁨이면 노랑, 위험이면 빨강을 나타낸다.

#### 2.1.2 실내 환경 분석을 위한 데이터 수집 및 제어 판단

##### a. 라즈베리파이를 통해 실내의 환경 파악, 조정

라즈베리파이의 미세먼지 센서가 실내 미세먼지 데이터를 측정한다. 라즈베리파이에 서버를 설치하여 서버와 MQTT 통신을 한다. 라즈베리파이는 정보를 분석하고 가공하는 일이 아닌, 실내 환경 데이터를 측정하거나 얻은 데이터를 서버에 송신한다.

연동 기기의 실행 여부를 MQTT 통신을 사용하여 서버에서 받아와 연동 기기를 실행시키거나 종료시켜 실내 미세먼지 농도를 관리한다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## b. 실내·외 환경 정보 저장

mongoDB를 사용하여 실내·외의 미세먼지 값을 저장할 DB를 서버에 구축한다. 라즈베리파이의 미세먼지 센서로부터 얻은 실내의 미세먼지 값과 대기환경공단에서 제공하는 공공데이터 Open API와 KMU Crowd Sensor Cloud를 통해 받아온 현재의 실외 미세먼지 값을 DB로 전달한다.

MQTT 프로토콜을 사용하여 라즈베리파이에서 측정한 실내 미세먼지 데이터를 웹 서버에게 전달한다. 메시지 전달 성능에서 MQTT 프로토콜은 HTTPS 통신에 비하여 3G 상에서는 90배, Wifi 상에서는 70배 이상의 속도차이를 보이고 있다. 그러므로 Mini BuT은 실시간으로 데이터값을 빠르게 보내기 위하여 MQTT 프로토콜을 사용한다.

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
Message/ Hour	1,708	160,278	3,628	236,314
Messages Received	240/ 1,024	1,024/ 1,024	524/ 1,024	1,024/ 1,024

<표 1. HTTPS와 MQTT의 성능 비교>

## c. 데이터를 실시간으로 처리하고 판단할 서버 생성


서버는 다양한 애플리케이션을 구현하는데 유용한 라이브러리가 내장된 Python을 사용하여 개발한다. 서버는 현재 실내·외의 환경 데이터를 MQTT 통신과 네트워크 통신으로 얻고 이를 DB에 실시간으로 저장한다. 그 후 DB에 저장된 실내·외 데이터를 비교 분석하여 Python으로 작성된 데이터 처리 알고리즘을 통해 연동 기기를 어떻게 제어해야 할지 지속적으로 판단한다.

데이터 처리 알고리즘은 다음과 같다.



<그림7. 데이터 처리 알고리즘>

서버는 DB에 저장된 현재 실내 데이터 값과 실외 데이터 값, 그리고 사용자가 설정한 실내 미세먼지 설정 값을 읽어온다. 여기서 실내 미세먼지 설정 값이란 Mini BuT의 사용자가 원하는 대로 설정할 수 있는 값으로, 사용자가 원하는 실내의 미세먼지 농도 값을 의미한다. 세가지 값을 읽어온 서버는 실내의 환경 관리를 세 단계로 나누어 실내 기기의 가동 여부를 결정한다. 결정된 값은 DB에 값이 저장된다. 이렇게 저장된 값은 서버가 일정 시간에 한번씩 Mini BuT에게 MQTT

 <b>국민대학교 컴퓨터공학부 캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

통신을 통하여 전송한다.

Mini BuT은 서버에게 받은 실내 기기 가동 여부와 실제 실내 기기의 가동 여부가 다를 경우, 받은 값으로 실내 기기 가동여부를 반영하여 연동 기기를 실행하거나 종료한다.

### 2.1.3 웹페이지 제작

웹 페이지는 Flask와 uWsgi, Nginx를 기반으로 하여 Html/Css/Js를 활용하여 구축한다.

#### a. 사용자 웹페이지 제작

사용자가 실내 환경을 확인하고 관리하기 위한 웹을 제작한다.

웹은 서버로부터 실시간 실내·외 미세먼지 데이터를 받아와 사용자에게 이미지와 그래프로 시각적인 자료를 제공해 사용자가 직관적으로 알 수 있게 한다. 사용자에게 현재와 과거의 미세먼지 데이터를 제공해줄 뿐만 아니라, 앞으로의 미세먼지 농도 예상 수치를 그래프를 통하여 사용자에게 제공한다. 또한 웹을 이용하여 사용자가 어디서나 실내 연동 기기의 가동 여부를 확인하거나 설정할 수 있게 한다.

사용자는 연동 기기의 가동 여부를 선택하여 직접 명령을 내릴 수도 있으며, 실내 먼지 설정값을 정하여 간접적으로 연동 기기의 가동 여부에 관여할 수 있다. 웹을 통해 변경된 연동 기기 가동 여부 값은 DB에 저장되며, 저장된 명령을 서버가 일정 시간에 한 번씩 Mini BuT으로 명령을 보내게 된다.


사용자는 이러한 웹페이지의 기능으로 인하여, 실내 환경을 더욱 효과적으로 관리할 수 있게 된다.

#### b. 관리자 웹페이지 제작

관리자가 전체 사용자들의 Mini BuT 사용에 대한 문제 파악과 이를 해결을 하는 것을 도와주는 웹을 제작한다.

관리자 웹페이지는 외부 미세먼지 데이터 수집에 사용되고 있는 기기들의 수치, 위치 등의 정보를 보여준다. 이는 지도를 통하여 보여주며, 그래프와 표 등으로 관리자가 더욱 효율적으로 데이터 값을 값을 파악할 수 있게 도와준다.

데이터 값이 비정상적이거나 Mini BuT의 전원이나 연결 문제 등으로 인해 데이터 제공이 멈추는 문제가 파악이 될 경우, 크게는 같은 실외 데이터를 공유하는 지역구 단위로 작게는 Mini BuT을 사용하는 한 가구 단위로 문제 해결에 필요한 제어명령을 보낼 수 있도록 한다.

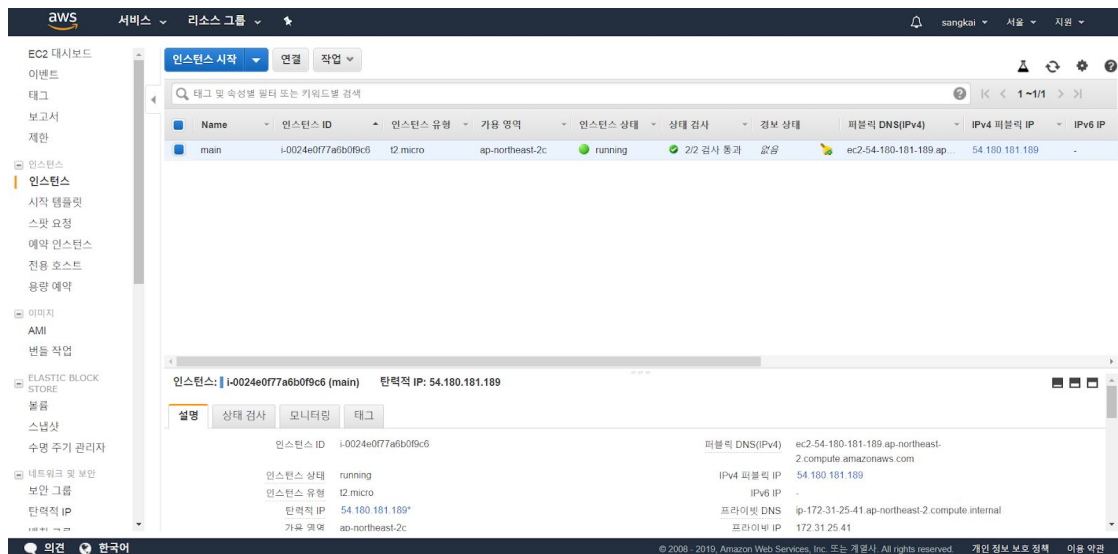
 <div> 국민대학교 컴퓨터공학부 캡스톤디자인 I </div>	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## 2.2 수행내용

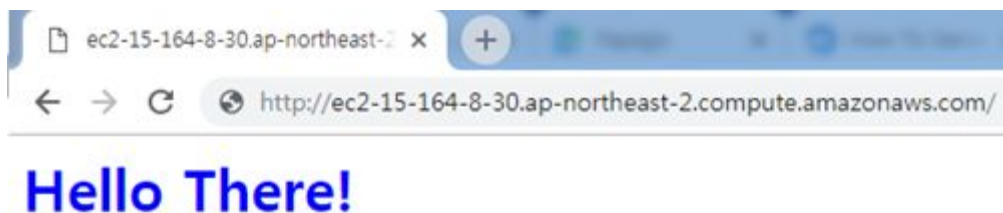
### 2.2.1 서버 구현

#### a. 서버 기본 환경 구성 완료

AWS EC2 Instance를 통하여 프로젝트를 진행할 서버를 구축하였다. AWS EC2 서버에서 Flask와 uWsgi, Nginx를 설치하여 웹에서 웹페이지 구현을 할 수 있도록 웹 서버 환경을 구현하였다. 사용자 페이지와 관리자페이지, Mini BuT 제어명령 시각화 페이지를 구현 할 수 있도록 하는 웹 서버이다.




<그림1. AWS EC2 서버 생성 >



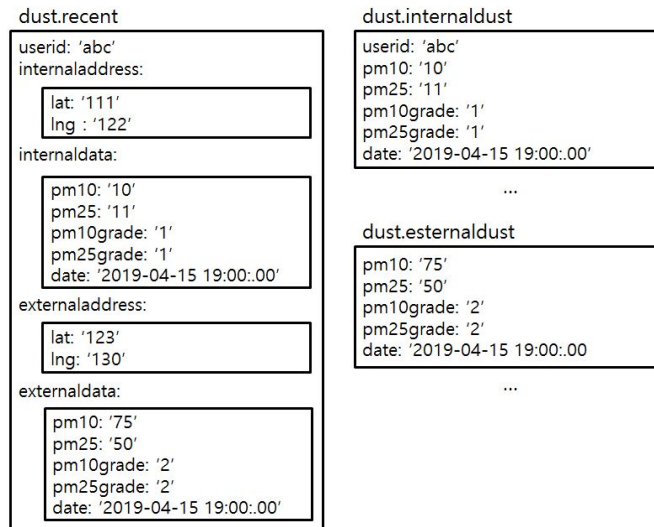
< 그림2. 웹 페이지 샘플코드 실행화면 >



 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## b. DB 설계 및 DB 서버 생성 완료

Databasename = dust



< 그림3. DB 설계 >

MongoDB로 DB 서버를 생성 후 인터페이스 언어로 python과 연동하기 위하여 pymongo를 설치하여 DB 서버와 연결하는 기능을 구현하였다.

DB 설계를 하여 미세먼지 데이터를 DB에 저장하는 방식을 정하였다.

recent collection은 사용자 한 명의 최근 실내외 미세먼지값을 저장한다. internaldust collection은 실내 미세먼지 값 누적으로 저장한다. 마찬가지로 externaldust collection은 실외 미세먼지 값을 누적으로 저장한다.

## c. 실외 데이터 수집 모듈 일부 구현 완료

공공데이터포털에서 제공하는 Open-API를 통하여 실외 데이터 현재 값을 저장하는 기능을 (Local에서) 구현하였다.

API를 GET방식으로 HTTP 요청하여 xml data를 받아온다. xml data 중 미세먼지 데이터를 DB 서버에 저장한다. API가 새로운 데이터를 제공하는 시간이 1시간 단위이므로 위와 같은 코드는 1시간 단위로 실행하도록 한다.



국민대학교  
컴퓨터공학부  
캡스톤디자인 I

## 중간보고서

프로젝트 명

Mini BUT

팀 명

MBT

Version 1.3

2019-04-14

```
knusw@knusw-ThinkPad-T440:~/바탕화면$ python3 apl.py
data time : 2019-04-17 14:00
pn10 Value : 59
pn25 Value : 19
pn10 Grade : 2
pn25 Grade : 2

이전 db 값
[{"_id": "Objectid('5c9d8776b02410ce0a6447e')", "date": "2019-03-28 22:00", "pn10value": "35", "pn25value": "20", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c9da8887d4339717f936bda')", "date": "2019-03-29 13:00", "pn10value": "45", "pn25value": "18", "pn10grade": "2", "pn25grade": "1"},
{"_id": "Objectid('5c4630d7d43390ef4f18238')", "date": "2019-04-03 16:00", "pn10value": "32", "pn25value": "16", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c7ffa77d43391058228aa7')", "date": "2019-04-12 11:00", "pn10value": "69", "pn25value": "28", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c8006f47d4339115d746b22')", "date": "2019-04-12 12:00", "pn10value": "55", "pn25value": "29", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c819657d433913fa2a9627')", "date": "2019-04-12 13:00", "pn10value": "62", "pn25value": "33", "pn10grade": "2", "pn25grade": "3"},
{"_id": "Objectid('5c819707d433913fa2a9629')", "date": "2019-04-12 13:00", "pn10value": "62", "pn25value": "33", "pn10grade": "2", "pn25grade": "3"}]

이후 db 값
[{"_id": "Objectid('5c9d8776b02410ce0a6447e')", "date": "2019-03-28 22:00", "pn10value": "35", "pn25value": "20", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c9da8887d4339717f936bda')", "date": "2019-03-29 13:00", "pn10value": "45", "pn25value": "18", "pn10grade": "2", "pn25grade": "1"},
{"_id": "Objectid('5c4630d7d43390ef4f18238')", "date": "2019-04-03 16:00", "pn10value": "32", "pn25value": "16", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c7ffa77d43391058228aa7')", "date": "2019-04-12 11:00", "pn10value": "69", "pn25value": "28", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c8006f47d4339115d746b22')", "date": "2019-04-12 12:00", "pn10value": "55", "pn25value": "29", "pn10grade": "2", "pn25grade": "2"},
{"_id": "Objectid('5c819657d433913fa2a9627')", "date": "2019-04-12 13:00", "pn10value": "62", "pn25value": "33", "pn10grade": "2", "pn25grade": "3"},
{"_id": "Objectid('5c819707d433913fa2a9629')", "date": "2019-04-12 13:00", "pn10value": "62", "pn25value": "33", "pn10grade": "2", "pn25grade": "3"},
{"_id": "Objectid('5cb6c297d43395aef69b0')", "date": "2019-04-17 14:00", "pn10value": "59", "pn25value": "19", "pn10grade": "2", "pn25grade": "2"}]
```

< 그림4. 외부데이터 저장 >

### d. 서버와 라즈베리파이 MQTT 통신 환경 구현 완료


라즈베리파이와 AWS EC2 서버 둘 다 MQTT 통신 할 수 있는 MQTT 서버를 구현하고 라즈베리파이에서 AWS EC2 서버로 메시지를 전송하면 수신되는 MQTT 클라이언트를 구현하였다. 실내 미세먼지 정보를 포함한 메시지가 전달되면 수신되는 것을 확인하는 것을 구현하였다.

```
knusw@knusw-ThinkPad-T440:~/바탕화면$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 localhost:27017 0.0.0.0:* LISTEN
tcp 0 0 localhost:domain 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:1883 0.0.0.0:* LISTEN
tcp 0 0 knusw-ThinkPad-T4:34790 tl-in-f189.1e100.https ESTABLISHED
tcp 0 0 knusw-ThinkPad-T4:55914 kg12s13-in-f4.1ehttps TIME_WAIT
tcp 0 0 knusw-ThinkPad-T4:36878 ec2-52-43-171-187.https ESTABLISHED
tcp 0 0 knusw-ThinkPad-T4:58780 kg12s02-in-f14.1ehttps ESTABLISHED
tcp 0 0 knusw-ThinkPad-T4:56844 103.22.229.133:http TIME_WAIT
tcp 0 0 knusw-ThinkPad-T4:41888 kg07s38-in-f3.1ehttps ESTABLISHED
tcp6 0 0 :::localhost:1pp [::]:* LISTEN
tcp6 0 0 [::]:1883 [::]:* LISTEN

knusw@knusw-ThinkPad-T440:~/바탕화면$ netstat -nap | grep mosquitto
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
knusw@knusw-ThinkPad-T440:~/바탕화면$ sudo netstat -nap | grep mosquitto
tcp 0 0 0.0.0.0:1883 0.0.0.0:* LISTEN 321
15/mosquitto
tcp6 0 0 :::1883 :::* LISTEN 321

knusw@knusw-ThinkPad-T440:~/바탕화면$ mosquitto_pub -d -t hello/world -m "hi, i am sangka!"
Client mosqpub1179-knusw-Thin sending CONNECT
Client mosqpub1179-knusw-Thin received CONNACK (0)
Client mosqpub1179-knusw-Thin sending PUBLISH (d0, q0, r0, m0, 'hello/world', ...
(16 bytes))
Client mosqpub1179-knusw-Thin sending DISCONNECT
knusw@knusw-ThinkPad-T440:~/바탕화면$ mosquitto_sub -d -t hello/world -m "hi, i am sangka!"
Client mosqsub1179-knusw-Thin sending CONNECT
Client mosqsub1179-knusw-Thin received CONNACK (0)
Client mosqsub1179-knusw-Thin sending PUBLISH (d0, q0, r0, m0, 'hello/world', ...
(16 bytes))
Client mosqsub1179-knusw-Thin sending DISCONNECT
knusw@knusw-ThinkPad-T440:~/바탕화면$
```

< 그림5. 로컬서버 내 MQTT 통신 >

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

```

message qos= 0
message retain flag= 0
message recived 3,{'pm10': 11.7, 'time': '29.03.2019 07:51:59', 'pm2.5': 8.5}
message topic= vds1/data
message qos= 0
message retain flag= 0
message recived 4,{'pm10': 13.3, 'time': '29.03.2019 07:53:02', 'pm2.5': 9.3}
message topic= vds1/data
message qos= 0
message retain flag= 0
^CTraceback (most recent call last):
  File "mqtt.py", line 18, in <module>
    client1.loop_forever()
  File "/usr/local/lib/python3.6/dist-packages/paho/mqtt/client.py", line 1578,
in loop_forever
    rc = self.loop(timeout, max_packets)
  File "/usr/local/lib/python3.6/dist-packages/paho/mqtt/client.py", line 1057,
in loop
    socklist = select.select(rlist, wlist, [], timeout)
KeyboardInterrupt
knusw@knusw-ThinkPad-T440:~/바탕화면$ python3 mqtt.py
message recived 1,{'pm2.5': 8.2, 'time': '29.03.2019 08:05:04', 'pm10': 12.5}
message topic= vds1/data
message qos= 0
message retain flag= 0

```


<그림6. 라즈베리파이와 서버 간 MQTT 통신>

## 2.2.2 Mini BuT 본체 구현

### a. 실내 미세먼지 데이터 수집 모듈 구현

미세먼지 데이터를 수집하고 이를 json 파일로 저장하는 실내 미세먼지 데이터 수집 모듈을 구현하였다.

라즈베리 파이에 연결된 미세먼지 측정 센서를 통해서 미세먼지를 측정한다. 미세먼지 센서는 실내 미세먼지 값을 30초 동안 2초 간격으로 총 15번 실내 미세먼지를 측정하여 평균 값을 계산한다. 센서는 미세먼지 측정 후, 30초의 딜레이 시간을 갖게 된다. 이렇게 1분 간격으로 측정된 값인 pm10, pm2.5 값은 측정 시간과 함께 json 파일로 저장된다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

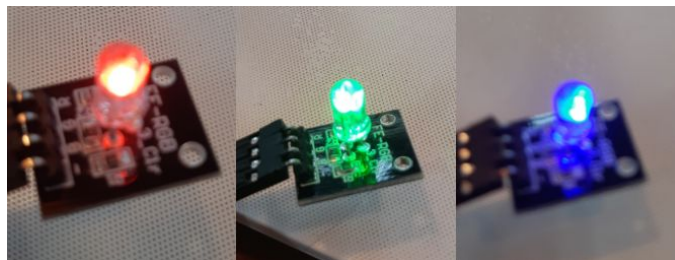
## b. 1602 LCD를 I2C로 연결하여 동작 확인 및 모듈 구현



<그림7. LCD 동작 확인>


I2C 주소를 설정하여 버스를 활성화시키고 이 주소에 명령어를 보내서 LCD에 메시지를 출력한다. 이를 이용한 LCD 모듈에는 PM10과 PM2.5 두 종류의 미세먼지 수치들을 위한 출력 포맷이 설정되어 있어서 두 수의 자릿수가 각각 다르더라도 동일한 포맷에 출력하도록 하였다. 연동 기기에 관해서도 현재 상정 중인 공기청정기와 창문의 상태를 출력할 포맷을 구현하였다.

## c. LED 동작 확인 및 모듈 구현



<그림8. LED 동작 확인>

red, green, blue, yellow, magenta, cyan, white의 총 7가지 색에 각각 on, off blink 기능을 모듈화해 놓았다. 이는 GPIO 출력을 통해 이뤄진다. 간단한 클래스 생성 및 함수 호출로 해당 기능을 사용할 수 있도록 구현하였다.

 <b>국민대학교 컴퓨터공학부 캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

#### d. 측정된 실내 미세먼지 데이터 전송 모듈 구현

json 파일로 저장된 실내 미세먼지 데이터를 MQTT 통신을 사용하여 서버에 전송한다. Mini BuT은 전송해야 할 서버와 연결을 위해 MQTT Broker인 Mosquitto를 사용하였고, Mqtt 프로토콜을 사용하여 서버와 연결을 하였다. Mini BuT과 서버와 연결이 되면 저장된 json 파일을 열어 가장 최근에 저장된 실내 미세먼지 데이터 정보를 1분에 한 번 씩, 서버에 전송을 한다.

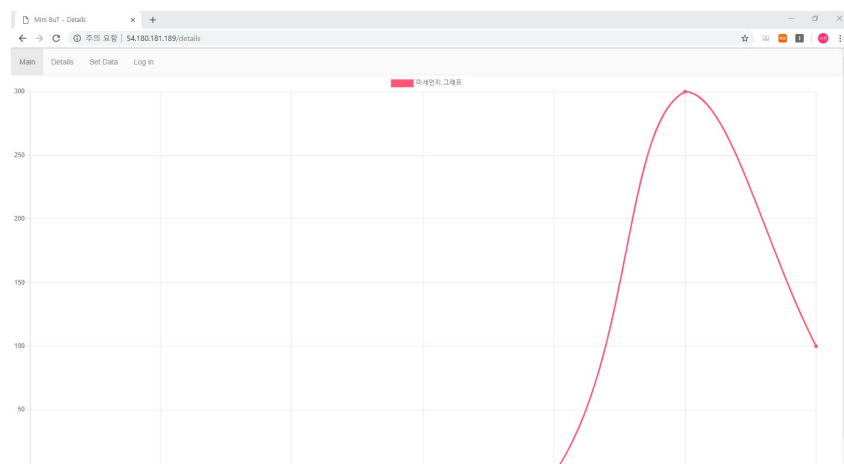
### 2.2.3 웹페이지 구현

웹 페이지는 Flask와 uWsgi, Nginx를 기반으로 하여 Html/Css/Js를 활용하여 구현한다.

#### a. DB 값을 웹 페이지에 표현


Flask 기반으로 구현한 웹을 Python 라이브러리인 Pymongo를 통해 mongoDB와 연동한 뒤, DB에 저장된 데이터를 웹으로 가져와 사용하였다. 가져온 데이터들을 자바스크립트 상에서 변수로 사용하여 DB 값을 웹 페이지에 구현하였다.

#### b. 그래프 차트 생성



<그림9. 그래프 차트 생성 페이지>

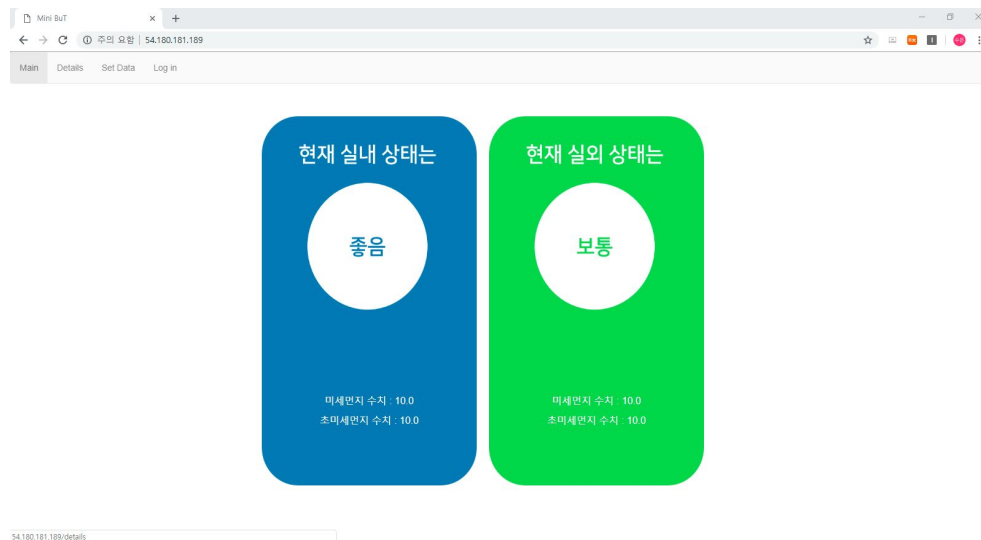
위의 구현 내용을 바탕으로 오픈소스 차트 라이브러리를 사용하여 자바스크립트 상에서

 <b>국민대학교 컴퓨터공학부 캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

DB에서 가져온 값을 사용해 웹페이지에 차트를 생성하는 기능을 구현하였다.

DB에 저장되는 값이 변함에 따라 그래프 차트의 변화를 볼 수 있으며 사용자는 이 차트를 보고 미세먼지 변화를 직관적으로 알아볼 수 있다.


### c. 현재 미세먼지 측정 값에 따른 시각화 표현



<그림10. 미세먼지 측정 값에 따른 시각화 표현>


DB에서 가져온 실내 미세먼지 데이터와 실외 미세먼지 데이터를 바탕으로 미세먼지 농도에 따른 다른 이미지를 출력하는 함수를 구현하였다. 이미지는 미세먼지 농도에 따라 매우 나쁨이면 빨강, 나쁨이면 노랑, 보통이면 초록, 좋음이면 파랑 등으로 한 눈에 알아 볼 수 있도록 구현하였다. 이와 함께 이미지 위에 현재 실내·외 미세먼지 데이터 값을 같이 출력해서 사용자가 간편하게 볼 수 있도록 구현하였다.



 <div> <b>국민대학교</b>  <b>컴퓨터공학부</b>  <b>캡스톤디자인 I</b> </div>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

항목	세부내용		1월	2월	3월	4월	5월	비고
SW / HW 설계	SW 설계	시스템 설계						
	HW 설계	Mini BuT 설계						
구현	개발 환경 구축	연동 기기(창문) 설계						
		라즈베리파이 환경 설정						
		물품 구매						
		라즈베리파이 서버 생성						
	통신 컴포넌트 구현	웹 서버 생성						
		DB 생성						
		Mini BuT→서버 수집 데이터 전송 모듈 구현						
		서버→Mini BuT 명령 전송 모듈 구현						
		Mini BuT→연동 기기 명령 전송 모듈 구현						
		웹→서버 명령 전송 모듈 구현						
		웹→서버 제어 설정값 전송 모듈 구현						
		서버→웹 데이터 전송 모듈 구현						
	데이터 처리 컴포넌트 구현	실내 데이터 수집 모듈 구현						
		실외 데이터 수집 모듈 구현						
		수집 데이터 분석 모듈 구현						
	웹 구현	데이터 시각화 모듈 구현						
		웹에서의 연동 기기 명령 모듈 구현						
		제어 설정값 변경 모듈 구현						
		웹 UI 디자인						
	연동 기기 구현	자동 구동 모듈 구현						
		수신된 명령 처리 모듈 구현						
	HW 제작	Mini BuT 본체 제작						
		Mini BuT 본체 케이스 제작						
		연동 기기(창문) 제작						
테스트	디바이스 성능 테스트							
	연동 기기 성능 테스트							
	소프트웨어 성능 테스트							

현재 프로젝트 진행사항을 봤을 때 프로젝트 계획서에서 작성한 진행 계획을 알맞게 설정하였다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

### 3 수정된 연구내용 및 추진 방향

#### 3.1 수정사항

##### 3.1.1 목표 변경

본 프로젝트의 기존 목표는 다음과 같았다.

- 1) 실내 환경 데이터와 실외 환경 데이터를 수집하여 적절한 실내 환경을 조성할 수 있도록 주변 환경 기기(IoT 창문, 공기청정기 등)를 통제하는 기능
- 2) 실내 미세먼지 상태를 LCD와 LED를 사용하여 사용자에게 알림 기능
- 3) 웹을 통해 실외에서도 실내 환경을 확인할 수 있고 연동 기기를 직접 제어하는 기능

이 중 1) 번 기능을 구현하기 위해 필요한 외부 미세먼지 데이터를 수집하는 과정에서 외부 미세먼지 데이터의 지역성 측면과 실시간 측면의 보강이 필요하다. 이를 위해 KMU Crowd Sensor Cloud를 활용하여 추가적인 외부 미세먼지 데이터 수집을하기로 하였다. 이에 따라 본 프로젝트의 목표 기능과 규모가 변경되었다.

자세한 변경 사항은 다음과 같다.

##### 3.1.2 상세 변경 사항


###### 1) 외부 미세먼지 데이터 수집 방식 추가

실외 미세먼지 데이터를 수집하는 과정에서 공공데이터 Open API와 Air Korea 등을 비롯한 미세먼지 데이터를 제공하는 사이트의 데이터를 분석해본 결과, 동일하게 서울시 기준 40여개의 측정소의 측정 데이터 값을 제공하였다. 이 부분에서 실제로 실외 미세먼지 데이터 값을 얻고자 하는 위치와 측정소의 거리가 멀다면, 실제 실외 미세먼지 데이터라고 하기에는 어려울 수 있다고 판단하였다. 이 점에서 데이터의 지역성 측면이 조금 더 보강이 되어야 한다고 판단했다.

뿐만 아니라 제공되는 외부 미세먼지 데이터들은 1시간을 주기로 갱신되는 점을 알 수 있었다. 제공되는 데이터는 이미 1시간 전의 값으로 실시간 데이터라고 하기에는 다소 적합하지 않다.

데이터의 실시간 측면이랑 지역성을 고려하여, 실외 데이터 수집을 기존에 고려했던 공공데이터 Open API를 사용 하되, KMU Crowd Sensor Cloud를 추가로 사용하게 되었다. KMU Crowd Sensor Cloud는 임의의 사용자가 개개인의 집에 설치한 미세먼지 측정기가 송신한 데이터들을 통합하여 관리하고 실시간 측정한 값을 제공하는 미세먼지 측정 플랫폼이다. KMU Crowd Sensor Cloud의 이러한 점은 Open API의 부족한 점인 데이터의 실시간 측면과 지역성을 일부 보완하는 역할이 가능하다. 또한, 실외 측정을 위하여 두 가지 방법을 모두 사용하기 때문에 예기치 못한 상황에 따른



 <b>국민대학교 컴퓨터공학부 캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

KMU Crowd Sensor의 다소 떨어지는 정확성은 공공데이터 Open API로 부터 보충할 수 있다.


KMU Crowd Sensor Cloud를 이용하여 추가적으로 외부 미세먼지 데이터를 수집하는 것에 따라 프로젝트의 목표의 방향이 기존의 실내 공기질을 전반적으로 관리하고 개선하는 것에서 실내의 미세먼지 농도를 관리하고 개선하는 것에 집중하는 것으로 변경되었다. 이에 따라 미세먼지에 관한 사항을 최우선사항으로 하여 기존에 함께 고려한 온·습도 및 이산화탄소 센서등은 추후에 구현하는 것으로 수정하였다.

## 2) 프로젝트 시연 방향성 변경

위와 같은 변경된 프로젝트의 방향성과 변경사항에 연관되어 추후에 프로젝트를 시연하는 부분에 있어서도 다음과 같이 적용하기로 한다.

기존에 시연 용으로 고려하였던 IR통신을 통한 창문의 제어는 프로젝트의 목표와 중요도에 비하여, 완성도 측면에서 비교적 많은 MD를 필요로 한다.

그러므로 시연을 위한 웹페이지를 따로 만드는 방향으로 수정을 하였다. 연동 기기의 동작을 시각화 하는 웹페이지를 만듦으로써 Mini BuT이 연동 기기를 얼마나 디테일하게 관리하는지를 기존에 고려하였던 창문을 통한 시연보다 더 직관적이고 가시적으로 보여줄 수 있다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### 4.1.1 서버 구현

##### a. 실외 데이터 수집 추가

KMU Crowd Sensor Cloud에서 제공하는 실외 데이터를 DB 서버에 추가로 저장할 수 있는 모듈을 구현할 예정이다.

##### b. DB 설계 수정 및 적용

사용자 정보를 저장하기 위한 DB와 실시간 데이터 처리 제어 명령을 저장하기 위한 DB가 추가되어야 하기 때문에 DB 설계를 수정할 계획이다. 그리고 실외 데이터가 수집 할 수 있는 방식이 추가 될 경우 DB collection을 추가하여 저장할 수 있도록 설계를 수정한다.

##### c. 라즈베리파이(Mini BuT)에서 받은 측정 데이터 DB에 저장

MQTT 통신을 통하여 서버에서 받은 메시지를 통하여 DB 서버에있는 collection에 맞게 저장하는 기능을 구현할 것이다.


##### d. 수집 데이터 분석 모듈 설계 수정 및 구현

DB에 저장된 미세먼지 데이터 값을 통해 실내 환경 상태를 여러 단계로 나누고, 이를 기준으로 실내 기기 가동 여부를 결정하는 수집 데이터 분석 모듈을 구현할 것이다.

DB에 저장된 실내 미세먼지 데이터 값과 실외 미세먼지 데이터 값, 그리고 사용자가 설정한 실내 미세먼지 설정값을 읽어온다. 여기서 실내 미세먼지 설정값이란 Mini BuT의 사용자가 원하는 대로 설정할 수 있는 값으로, 사용자가 원하는 실내의 미세먼지 농도 값을 의미한다. 수집 데이터 분석 모듈은 이 세가지 값을 통해, 현재 실내 환경을 여러 단계로 나눈다. 단계마다 실내 기기 가동여부가 달라지며, 측정된 단계에 따라 실내 기기 가동여부를 결정한다. 결정된 값은 DB에 저장을 한다.

##### e. 서버에서 라즈베리파이로 제어 명령 전달 통신 환경 구현

서버에서 라즈베리파이로 제어 명령을 전달 할 수 있도록 라즈베리파이에 MQTT 서버를 설치하여 데이터를 받는 환경을 구축한다. 받은 제어 명령 데이터를 통하여 연동 기기를 제어 할 수 있는 기능을 구현할 것이다.

 국민대학교 컴퓨터공학부 캡스톤디자인 I	중간보고서		
	프로젝트 명	Mini BuT	
	팀 명	MBT	
		Version 1.3	2019-04-14

#### 4.1.2 Mini BuT 본체 구현

##### a. Mini BuT 전체 프로그램 완성

현재 모듈화되어 있는 미세먼지 센서, LCD, LED, MQTT를 이용한 서버 통신 모듈을 전부 통합하여 Mini BuT 프로그램을 완성할 것이다.

##### b. 서버에서 받은 명령어 처리 구현

현재는 서버에서 MQTT 통신을 통해 명령어를 받는 것까지 구현이 되어있으며, 이를 저장하고 처리하는 것이 구현이 되어있지 않다. 차후, 서버에서 보내는 명령을 저장하여 저장된 명령어에 맞춰서 연동 기기를 실행하는 기능을 구현할 것이다.

#### 4.1.3 웹 구현

##### a. 계정 페이지

사용자가 늘어남에 따라 개개인이 Mini BuT을 사용하게 된다면 개인 계정이 필요하게 될 것이고, 그에 따라 로그인 기능도 필요해질 것이다. 계정 정보는 DB에 저장되어 웹과 연동되어 사용될 수 있도록 계정 페이지를 구현할 것이다.

##### b. 설정 페이지

Mini BuT과 연동기기 간의 제어를 위한 설정 값을 개인이 원하는 대로 설정하기 위해 개인 설정 페이지를 구현할 것이다. 기본 제어 알고리즘은 서버에서 제공되지만, 설정 페이지에서 사용자가 미세먼지 농도, 작동 시간 등의 설정 값을 임의로 설정해 기기 제어에 개입할 수 있도록 한다.


##### c. 통계 페이지

수집한 데이터를 바탕으로 그래프와 이미지를 사용한 통계 웹페이지를 구현할 것이다. 그래프를 통해 미세먼지 변화를 볼 수 있다.


##### d. 관리자 페이지

관리자가 전체 사용자들의 Mini BuT 사용에 대한 문제 파악과 이를 해결을 하는 것을 도와주는 웹 페이지를 구현한다. 관리자 웹페이지는 외부 미세먼지 데이터 수집에 사용되고 있는 기기들의 수치, 위치 등의 정보를 보여준다. 이는 지도를 통하여 보여주며, 그래프와 표 등으로 관리자가 더욱 효율적으로 데이터 값을 파악할 수 있게 도와준다.

데이터 값이 비정상적이거나 Mini BuT의 전원이나 연결 문제 등으로 인해 데이터 제공이

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

멈추는 문제가 파악이 될 경우, 크기는 같은 실외 데이터를 공유하는 지역구 단위로 작게는 Mini BuT을 사용하는 한 가구 단위로 문제 해결에 필요한 제어명령을 보낼 수 있도록 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	Mini BuT	
	<b>팀 명</b>	MBT	
		Version 1.3	2019-04-14

## 5 고충 및 건의사항

캡스톤 프로젝트를 항시 진행하기 위한 충분한 공간이 부족하다.