

# 소형 엣지컴퓨팅을 이용한 미세먼지 모니터링 시스템 개발

<sup>1</sup>황기환

## Development of Fine Dust Monitoring System Using Small Edge Computing

<sup>1</sup>KiHwan Hwang

### 요약

최근 초미세먼지 및 미세먼지에 대한 심각성은 국가적 차원의 재난으로 대두되고 있으나 지방 중소도시는 면적에 비해 미세먼지 측정소가 부족하여 미세먼지 관리가 어려운 측면이 있다. 미세먼지 데이터의 수집과 처리를 위한 컴퓨팅 자원은 규모가 크지 않지만 데이터를 공유를 위하여 클라우드와 민간 및 공공 데이터를 활용하는 것이 필요하다.

본 논문에서는 미세먼지 및 초미세먼지 그리고 온·습도를 측정하여 이를 처리하여 미세먼지 실시간 관제와 대국민서비스할 수 있는 소형 엣지컴퓨팅 시스템을 제안하였다. 미세먼지 데이터의 수집과 공공 및 민간 데이터를 활용하여 미세먼지 등급을 서비스하는 것은 데이터양이 크지 않고 처리부하가 크지 않기 때문에 라즈베리파이를 이용한 엣지컴퓨팅으로 처리하는 것이 효율적이다. 실험을 위하여 3가지 센서와 라즈베리파이 그리고 Thinkspeak를 이용하여 실험시스템을 구성하였으며 경북북부권지역에 대한 미세먼지 측정을 실험하였다. 실험결과 민간의 GIS데이터 기반에 시간에 따른 측정된 미세먼지 측정결과가 정확하게 확인되었다.

### Abstract

Recently, the seriousness of ultrafine dust and fine dust has emerged as a national disaster, but small and medium-sized cities in provincial areas lack fine dust monitoring stations compared to their area, making it difficult to manage fine dust. Although the computing resources for collecting and processing fine dust data are not large, it is necessary to utilize cloud and private and public data to share data.

In this paper, we proposed a small edge computing system that can measure fine dust, ultrafine dust and temperature and humidity and process it to provide real-time control of fine dust and service to the public. Collecting fine dust data and using public and private data to service fine dust ratings is efficient to handle with edge computing using raspberry pie because the amount of data is not large and the processing load is not large. For the experiment, the experiment system was constructed using three sensors, raspberry pie and Thinkspeak, and the fine dust measurement was conducted in northern part of kyongbuk region. The results of the experiment confirmed the measured fine dust measurement results over time based on the GIS data of the private sector.

keyword: fine dust, monitoring system, edge computing, data service, IoT

<sup>1</sup> 경북전문대학교 소방안전관리과 교수 (hkh@kbc.ac.kr)

## I. 서론

현재 미세먼지와 초미세먼지는 인체와 생태계 그리고 산업과 경제 전반에 걸쳐 막대한 피해를 주고 있으며 그 피해의 규모는 점점 더 커질 것이다. 미세먼지는 질산염( $\text{NO}_3$ ), 암모늄 이온( $\text{NH}_4^+$ ), 황산염( $\text{SO}_4^{2-}$ ) 등의 이온 성분과 탄소 화합물과 금속 화합물 등으로 이루어져 눈에 보이지 않을 정도로 작은 먼지 형태로 대기에 떠다니는데 장기간 미세먼지에 노출될 경우 면역력이 급격히 저하되어 감기, 천식, 기관지염 등의 호흡기 질환은 물론 심혈관 질환, 피부질환, 안구질환 등 각종 질병에 노출될 수 있다. 특히 직경  $2.5 \mu\text{m}$  이하의 초미세먼지는 인체 내 기관지 및 폐 깊숙한 곳까지 침투하기 쉬워 기관지, 폐 등에 붙어 각종 질환을 유발한다.

또한, 제조공정에 먼지가 유입되면 불량률이 발생하는 반도체 및 디스플레이, 전자기기에 매우 치명적이며 실외 작업이 많은 조선, 건설, 농업의 경우 생산성 저하와 비용지출이 증가하고 가축의 경우 폐렴 등 호흡기 질병이나 결막염 같은 인구 질환 발생이 증가하며 항공기 운항이나 물류에도 악영향을 끼치고 있다.

이를 해결하기 위해서는 화력발전의 연료를 석탄에서 LNG 로 대체하고 전기자동차 확대, 사업장의 배출허용기준 대폭강화 및 먼지총량제 신규도입 등 다양한 노력과 대책이 필요하지만, 이는 막대한 비용과 상당한 시간이 필요하기 때문에 가장 현실적인 대책은 미세먼지 수치가 높아질 때 경보를 발령하고 이에 따라 대응하는 것이다. 그러기 위해서는 전국에 촘촘한 미세먼지 측정소를 배치해야 하는데 현재 면적에 비해 측정소가 적어 이들 측정소가 측정한 데이터가 해당 지역의 측정소에 대한 상황이라고 하기에는 문제가 많은 편이다.[1,2]

엣지컴퓨팅(Edge Computing)이란 클라우드 컴퓨팅의 변형된 컴퓨팅서비스로 통신망을 구축하는 경우 단말기와 통신하는 게이트웨이 또는 무선 기지국에 클라우드 및 컴퓨팅 기술을 적용한 서비스를 말한다. 즉, 센싱 및 서비스 실현되어야 하는 현장(엣지)에 현장의 센싱 및 서비스 처리에 적정한 정도의 컴퓨팅 자원을 설치하여 즉시성을 요구하는 서비스를 현장에서 처리하게 하는 모델이다. 엣지컴퓨팅에 대한 연구는 IoT 가 활성화된 모바일 환경에 많은 종류의 센서에 대한 정보 및 단말기의 증가로 인해 물리적으로 떨어져 있는 중앙 집중형 원격지 클라우드에 정보를 전송하는 경우 많은 양의 트래픽이 발생하게 되고 이로 인한 모바일 코어망에 대한 트래픽의 부담이 생기게 된다. 따라서 이러한 문제를 해소하기 위하여 연구 개발된 컴퓨팅 기술이다. 엣지 컴퓨팅 기술은 프론트 홀(Fronthaul)/백 홀(backhaul) 상에 지연시간(Latency Time)을 줄이고 높은 대역폭(High bandwidth)을 가지게 하여 망의 트래픽을 줄일 수 있도록 설계된 서비스이다.[2,3]

따라서 미세먼지와 초미세먼지 그리고 온·습도를 측정하여 누구나 활용할 수 있게 클라우드에 저장하고 특정 수치 이상일 때 문자로 알리는 기능과 모니터링 하는 시스템이 필요하다.

본 논문에서는 미세먼지 및 초미세먼지 그리고 온·습도를 측정하고 이를 엣지컴퓨팅으로 처리하여 미세먼지 실시간 관제와 대국민서비스할 수 있는 시스템을 제안하였다. 미세먼지 데이터의 수집과 공공 및 민간데이터를 활용하여 미세먼지 등급을 서비스하는 것은 데이터양이 크지않고 처리부하가 크지 않기 때문에 라즈베리파이를 이용한 소형엣지컴퓨팅 기법으로 처리하는 것이 비용과 처리성능측면에서 효율적이다.

## II. 엣지컴퓨팅을 이용한 미세먼지 관리 시스템

### 2.1 엣지 컴퓨팅

엣지컴퓨팅은 엣지 컴퓨팅 기술은 클라우드컴퓨팅의 프론트 홀(Fronthaul)/백 홀(backhaul) 상에 지연시간(Latency Time)을 줄이고 현장과 컴퓨팅서버 사이의 높은

대역폭(High bandwidth)을 가지게 하여 망의 트래픽을 줄일 수 있도록 설계된 서비스이다. 그림 1 은 ESTI 에서 제안한 모바일 엣지 컴퓨팅에 대한 플랫폼의 구조를 보여주고 있다[2,3,4].

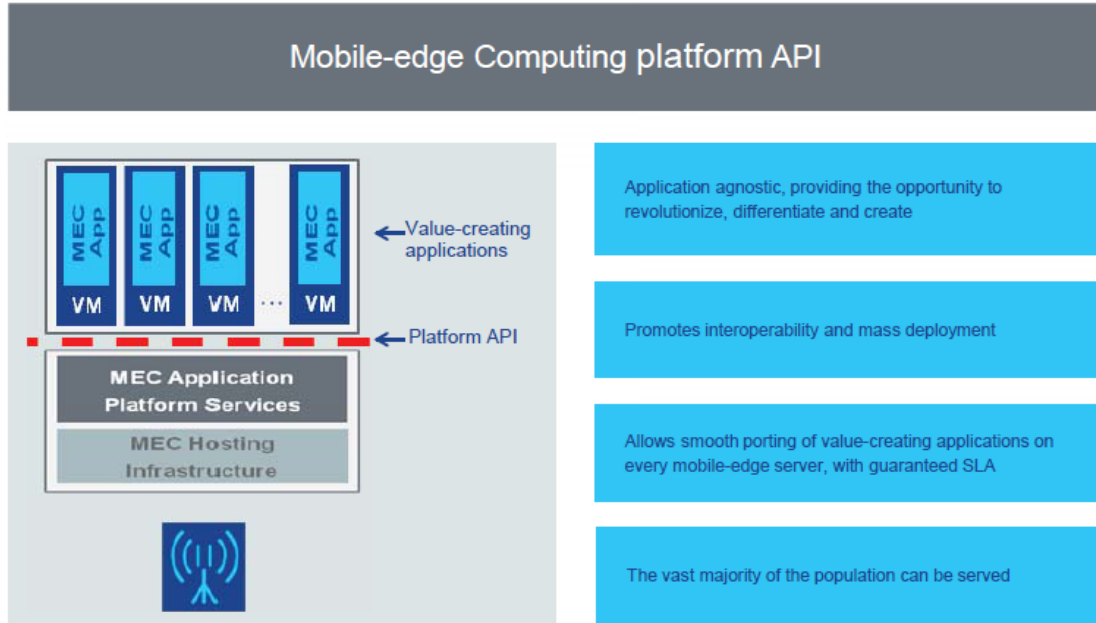


Figure 1.

그림 1 과 같이 엣지 컴퓨팅의 플랫폼은 개별기기마다 데이터를 분석하고 처리할 수 있는 응용프로그램을 VM(Virtual Memory)내에 두어 실시간으로 데이터를 분석하고 처리할 수 있도록 하는 플랫폼이다. 클라우드컴퓨팅 플랫폼에 비해서는 VM 수 등 규모가 적지만 제공하는 서비스와 엣지단의 처리부하를 고려하여 확장성을 가질수 있기 때문에 매우 유용한 개념이다.

클라우드 컴퓨팅 방법은 사용량에 기반을 둔 합리적인 가격모델을 가지고 있으나 엣지 컴퓨팅은 엔드 포인트에서 직접 데이터가 수집되므로 외부망이 필요하지 않다. 클라우드 컴퓨팅 방법에서는 서버에 대한 안전성 및 보안 문제에 대한 우려가 있다. 반면 엣지 컴퓨팅의 컴퓨팅 특성은 기존의 클라우드가 가지고 있는 MPU 의 성능에 비해 상대적으로 성능이 떨어지므로 처리속도에 한계를 가질 수 있다.[5,6]

엣지컴퓨팅은 클라우드컴퓨팅의 축소형이지만 중소도시에 미세먼지 서비스를 위한 미세먼지 센싱 및 관제시스템으로는 규모와 비용측면에서 낭비가 있으므로 본 논문에서는 라즈베리파이 정도의 소형서버로 센싱과 서비스 처리를 현장에서 할수 있고 데이터는 클라우드로 전송하여 저장할 수 있는 소형 엣지컴퓨팅을 설계하였다.

## 2.2 IoT 센싱설계

미세먼지 등 다수의 센서를 이용하는 모니터링 시스템의 데이터 수집모듈의 기본구조는 그림 2 와 같이 센서에서 데이터를 수집하는 IoT 센서 인터페이스 부, MCU(Micro Controller Unit) 그리고 클라우드 및 데이터포털 등과 데이터를 송수신하는 유무선 네트워크 인터페이스 부로 구성된다.

IoT 데이터 수집모듈은 센서에서 수집된 정보를 이기종간의 프로토콜 변환하여 원격지에 있는 클라우드와 데이터포털과 송신하는 기능을 가지고 있으며 원격지에서 수신된 명령을 처리하거나 센서에 정보를 전송하는 기능을 가진다. [7,8]

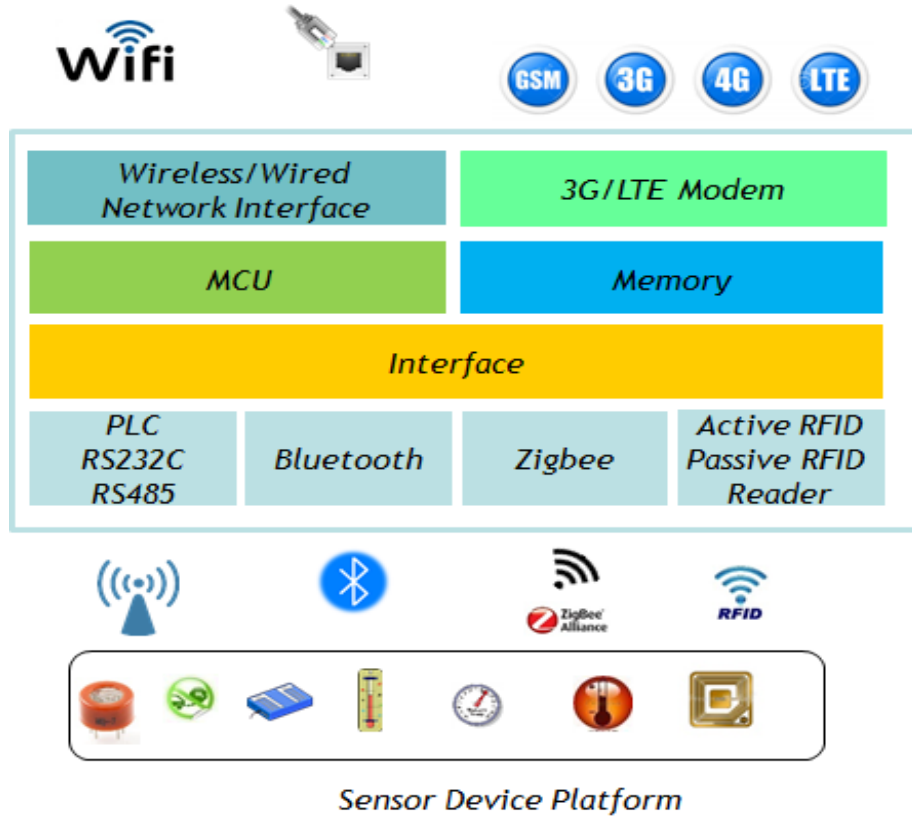


Figure 2.

### 2.3 소형 엣지컴퓨팅을 이용한 미세먼지 관리시스템 설계

본 논문에서는 중소규모 도시나 산악지역과 같이 외진 지역에 대한 저비용 미세먼지 관제를 구현하기 위하여 소형엣지컴퓨팅을 이용한 미세먼지 관리시스템을 설계하였다. 제안된 시스템은 미세먼지 등 4개의 센서로부터 데이터를 센싱하고 이를 클라우드로 전송하는 기능과 센싱된 데이터를 미세먼지 관제센터로 전송하는 기능과 공공데이터 포털과 민간데이터 포털로부터 필요한 데이터를 가져와 처리하는 기능과 정해진 등급 이상의 미세먼지가 측정되는 경우, 엣지컴퓨터에 포함된 GPS 와 정보와 GIS 정보 그리고 통신사 기지국정보를 종합하여 기지국 셀내의 국민들에게 미세먼지 경고 문자를 자동으로 보내주는 기능이 있다.

그림 3은 본 논문에서 설계된 소형 엣지컴퓨팅을 이용한 미세먼지 관리시스템이다.

제안된 소형 엣지컴퓨팅 시스템은 미세먼지 및 온·습도를 측정하는 센서모듈과 측정된 Data 를 Parsing 하여 HTTP Protocol 로 전송하는 서버 기능을 하는 컴퓨팅 모듈로 구성되며 센서에서 측정된 데이터는 클라우드에 저장하고 서버는 클라우드에 저장된 데이터와 공공데이터 포털과 민간데이터 포털에서 제공하는 타 측정소의 값을 가져와 웹이나 앱에서 비교할 수 있도록 정보를 시각화하여 출력한다. 또한 센서의 위치를 알려주는 GPS 를 추가하여 좌표값 데이터를 이용하고 네이버나 다음 지도 같은 민간지도 데이터를 이용하여 네이버 지도위에 표시하는 GIS 기능도 포함하고 있다.

제안된 시스템은 GPS 모듈을 포함하고 있기 때문에 센서위치를 알 수 있으며 이 좌표 데이터와 기지국 데이터를 이용하면 기지국셀 내부에 속해있는 국민들에게 미세먼지 경보문자를 보내는 기능이 가능하다.

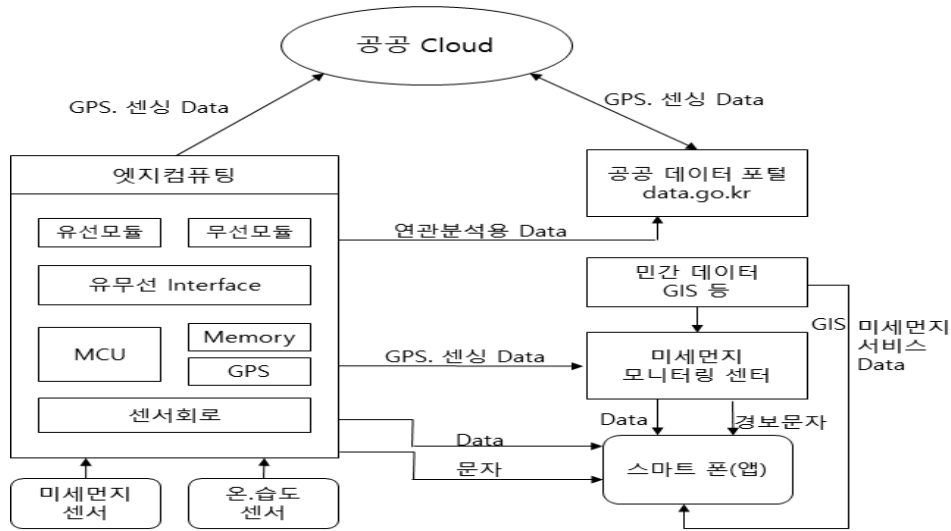


Figure 3. 제안된 소형 엣지컴퓨팅을 이용한 미세먼지 관제시스템

### III. 실험 및 고찰

#### 3.1 하드웨어 설계 및 구현

본 논문에서는 미세먼지와 온·습도 측정을 위해 PMS7003M 미세먼지 측정 센서와 DHT22 온·습도 측정 센서를 사용하였다. 레이저 측정 방식인 PMS7003M 센서는 미세먼지 PM10( $10 \mu\text{g}/\text{m}^3$ )와 초미세먼지 PM2.5( $2.5 \mu\text{g}/\text{m}^3$ ) 그리고 극초미세먼지 PM1.0( $1 \mu\text{g}/\text{m}^3$ )을 동시에 측정할 수 있고 센서 내부에 공기순환 FAN 이 있어 측정 값이 균일하며 센서값을 디지털 데이터로 변환해서 출력해 주기 때문에 노이즈 영향이 거의 없어 매우 안정적으로 동작한다. PMS7003M 센서는 샤오미 공기청정기에 적용될 만큼 신뢰성이 높은 편이다. DHT22 온·습도 센서는 측정 범위가  $-40\sim 80^\circ\text{C}$ ,  $0\sim 100\%RH$ 며 오차는  $\pm 0.5^\circ\text{C}$ ,  $\pm 2\%RH$ 이다. 센서 장치에 사용된 모듈은 WiFi 기능이 내장된 ESP8266 NodeMCU 를 사용하였다.

Table 1. PMS7003M 미세먼지측정 센서 사양표

Maximum Consistency Error (PM2.5 standard data)*	$\pm 10\% @ 100\sim 500 \mu\text{g}/\text{m}^3$ $\pm 10 \mu\text{g}/\text{m}^3 @ 0\sim 100 \mu\text{g}/\text{m}^3$	
Standard Volume	0.1	Litre (L)
Single Response Time	<1	Second (s)
Total Response Time	$\leq 10$	Second (s)
DC Power Supply	Typ:5.0 Min:4.5 Max: 5.5	Volt (V)
Active Current	$\leq 100$	Milliampere (mA)
Standby Current	$\leq 200$	Microampere ( $\mu\text{A}$ )
Interface Level	L < 0.8 @ 3.3 H > 2.7 @ 3.3	Volt (V)
Working Temperature Range	$-10\sim 60$	$^\circ\text{C}$
Working Humidity Range	$0\sim 99\%$	
Storage Temperature Range	$-40\sim 80$	$^\circ\text{C}$
MTTF	$\geq 3$	Year (Y)
Physical Size	$48 \times 37 \times 12$	Millimeter (mm)

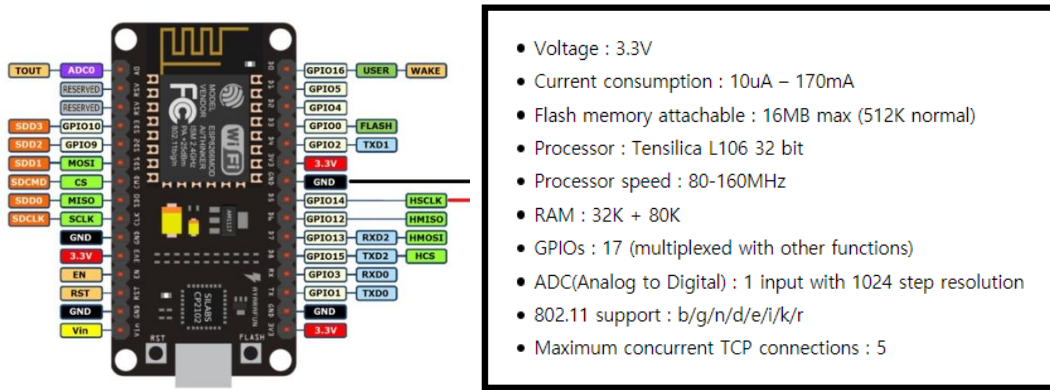


Figure 4. ESP8266 NodeMCU/스펙

하드웨어 구현에서 가장 중점을 둔 부분이 측정된 데이터의 신뢰성이었다. 신뢰성 확보를 위해 미세먼지와 온·습도 값을 1분 간격으로 총 10 번을 측정한 후 평균을 구한 다음 10분마다 그 값을 Cloud에 전송하였다. 그리고 매일 새벽 2시에 센서를 리부팅하여 메모리 문제나 기타 이유로 센서가 Freezing 되는 것을 방지하고 항상 안정된 상태에서 측정할 수 있게 구현하였다.

### 3.2 소프트웨어 설계 및 구현

그림 5와 같이 ESP8266 NodeMCU 모듈에 내장된 WiFi AP 기능을 이용, 센서에 접속하여 AP 이름, 접속할 SSID, 정확한 시간을 받아 오기 위한 NTP 서버, Cloud 접속 정보, 그리고 문자 메시지 발송 여부와 발송 정보, 경고 수치 등을 설정할 수 있도록 하였으며 설정된 정보는 그림 6과 같이 함수를 사용하여 ESP8266 NodeMCU 모듈의 EEPROM에 저장되도록 하였다.

AP Name	Smart-IoT
비밀번호	설정불가
SSID	[Redacted]
비밀번호	••••••••
시간서버	kr.pool.ntp.org
동기화 주기	1h
ThingSpeak™	
ID	[Redacted]
Write KEY	••••••••••
Nexmo Setting	
API KEY	[Redacted]
Secret KEY	••••••••••

Figure 5. 센서 설정 화면

EEPROM에 저장된 값을 수정할 때 먼저 저장된 값의 길이와 수정된 값의 길이가 다른 경우 에러가 발생 되므로 수정하기 전에 저장된 값의 길이를 읽어 기존 값을 모두 삭제한 후 다시 쓰도록 정교하게 구현하였다.

```

void writeEeprom(String get_char, int Snum, int Enum)
{
  int char_size = -1;
  for(int i = Snum; i <= Snum+get_char.length(); i++)
  {
    char_size ++;
    EEPROM.write(i, get_char[char_size]);
  }

  for(int j = Snum+get_char.length()+1; j < Enum; j++)
    EEPROM.write(j, 0);

  EEPROM.commit();
}

```

Figure 6. EEPROM 저장 함수

그리고 센서 점검이 필요할 경우를 대비하여 그림 7 과 같이 PC 나 스마트폰으로 센서에 직접 접속하여 센서의 상태나 설정 정보, 측정된 값을 모니터링 할 수 있도록 하였다.



Figure 7. 센서 모니터링 화면

센서에서 측정된 미세먼지나 초미세먼지 수치가 설정한 값보다 높을 경우 이를 문자 메시지로 발송하는 기능을 추가했으며 문자 메시지 경고 알림은 그림 5 의 센서 설정 화면에서 등록 Key 와 Secret Key 를 입력한 후 사용여부를 체크하면 바로 사용이 가능하도록 프로그래밍하였다. 국내 문자 메시지 발송 대행사의 경우 C 언어 환경에서

사용할 API 를 제공하지 않고 국내법에 의해 사전에 발송번호를 등록해야 하는 등 여러가지 제한 사항이 많아 해외 Nexmo 사의 서비스를 이용하였다.

한편, 본 연구에서 제안하는 소형 엣지컴퓨팅을 구현하기 위하여 Cloud 에 저장된 Data 와 공공데이터 포털에서 제공하는 타 측정소의 값을 가져와 Web browser 로 출력하기 위한 소형엣지컴퓨팅 Server 는 Raspberry Pi 3B 를 사용하였다. Raspberry Pi 3B 에 CentOS 를 설치하고 Apache 와 PHP 를 세팅하여 Server 환경을 세팅했으며 ThingSpeak Cloud 와 공공데이터 포털에서는 필요한 Data 를 XML 형태로 제공하기 때문에 PHP 웹 프로그래밍 언어를 이용하여 그림 8 과 같이 XML 을 Parsing 하여 원하는 부분만 사용하였다.

```
//thingspeak.com
$ch = curl_init();
$url = 'https://thingspeak.com/feeds/1234567890';
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 60);
$response = curl_exec($ch);
curl_close($ch);

//var_dump($response);
$xml = simplexml_load_string($response);
$tmp = $xml->feeds->feed->field1;
$hpm = $xml->feeds->feed->field2;

$tmp = substr($tmp,0,4);
$hpm = substr($hpm,0,4);
```

Figure 8. Cloud 에 저장된 Data 를 읽어오는 PHP code

```
// 공공데이터 미세먼지
$ch = curl_init();
$url = 'http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforIn';
$queryParams = '?' . urlencode('ServiceKey') . '=1YKh0U7yMvhQSTZsu5nAfd1M';
$queryParams .= '&' . urlencode('numOfRows') . '=' . urlencode('1');
$queryParams .= '&' . urlencode('pageNo') . '=' . urlencode('1');
$queryParams .= '&' . urlencode('stationName') . '=' . urlencode('가평동');
$queryParams .= '&' . urlencode('dataTerm') . '=' . urlencode('DAILY');
$queryParams .= '&' . urlencode('ver') . '=' . urlencode('1.3');

curl_setopt($ch, CURLOPT_URL, $url.$queryParams);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_HEADER, FALSE);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
$response = curl_exec($ch);

curl_close($ch);
//var_dump($response);
$xml = simplexml_load_string($response);
$pm10 = (string)$xml->body->items->item->pm10Value;
$pm25 = (string)$xml->body->items->item->pm25Value;
```

Figure 9. 공공데이터 포털에서 Data 를 읽어오는 PHP code

### 3.3 실험결과

센서에서 측정된 값은 ThingSpeak 의 Cloud 에 저장된다. ThingSpeak 는 인터넷 또는 근거리 통신망을 통해 HTTP 및 MQTT 프로토콜을 사용하여 사물인터넷(IOT)에서 측정된 데이터를 자사의 Cloud 에 저장하고 검색 및 PC 및 스마트폰에서 모니터링 하는 환경을 제공한다.



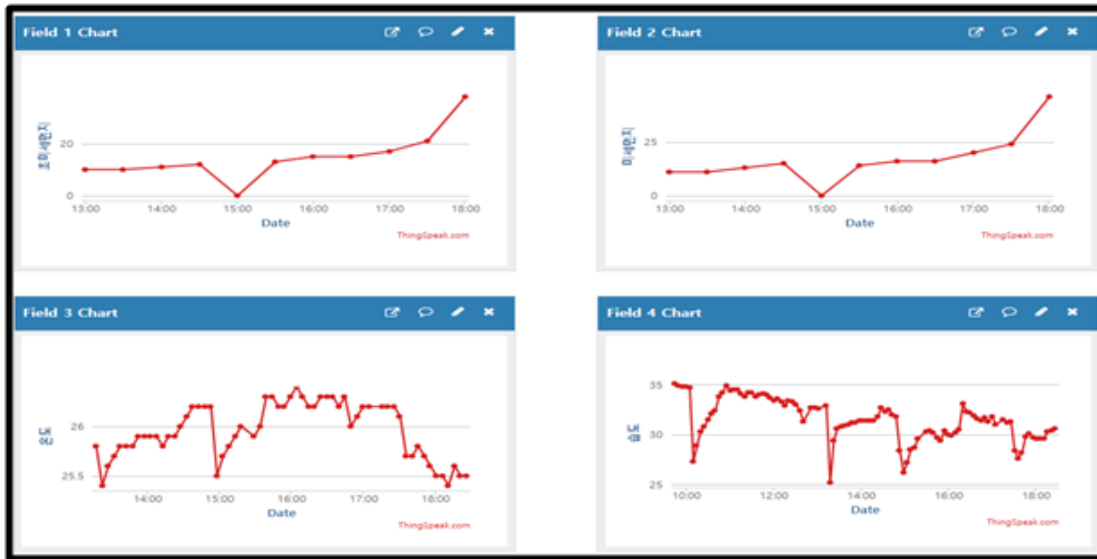


Figure 10. Cloud 모니터링 화면



Figure 11. 스마트폰 앱으로 접속한 Cloud

한편, GIS 기능을 구현하기 위해 센서에서 Cloud 에 Data 를 전송할 때 해당 그림 13 과 같이 센서의 좌표를 포함했으며 Server 에서 Javascript 를 이용하여 좌표를 지도에 대입하여 센서가 위치한 장소에 측정된 값이 출력되도록 하였다.

**Channel Settings**

Percentage complete 30%

Channel ID 00000000

Name 조미세먼지

Description

Field 1 조미세먼지

Field 2 미세먼지

Field 3 온도

Field 4 습도

Field 5 좌표

Figure 12. GIS 좌표 등록



Figure 13. 미세먼지 및 초미세먼지 생태 지도에 출력

본 논문을 위해서 시스템을 개발하여 국립백두대간수목원 방문자센터에 센서를 설치하고 운영한 결과 매우 안정적으로 미세먼지가 측정되는 것을 확인하였다.

#### IV. 결론

본 논문에서는 센서에 의하여 미세먼지를 측정하고 이를 센서와 함께 패키징된 소형엠티컴퓨팅 서버에서 처리하여 미세먼지 실시간 관제와 대국민서비스할 수 있는 소형 엠티컴퓨팅 기반 미세먼지 모니터링 시스템을 제안하였다. 이를 위하여 미세먼지 데이터의 수집과 공공 및 민간데이터를 활용하여 미세먼지 등급을 서비스하는 기능을 처리하는 엠티컴퓨팅 부분을 라즈베리파이를 이용하고 Thinkspeak 를 이용하여 시스템을 설계하였다. 설계된 시스템을 라즈베리 하드웨어 소프트웨어를 구현하여 경북북부권지역에 대한 미세먼지 측정을 실험한 결과, 미세먼지 측정결과가 실시간으로 모니터링 됨을 확인하였다. 특히 민간의 GIS 데이터를 소형엠티컴퓨팅기반으로 처리하여 시간에 따라 변화하는 미세먼지 측정결과가 지도상에 시각화하여 모니터링 됨을 확인되었다.

본 연구결과는 클라우드와 멀리 떨어져 있는 중소규모 도시지역이나 농촌지역 그리고 산촌지역의 미세먼지 측정과 원격모니터링 시스템으로 활용될 수 있을 것으로 기대된다. 특히, 저렴한 비용으로 민간 GIS 시스템과 연계하여 지도상에 미세먼지 모니터링을 할 수 있고 센서와 결합되어 있는 GPS 센싱데이터와 연계하는 기지국 셀 내에 속한 국민에게 미세먼지 위험 등급일 경우 자동으로 안내하는 기능까지 가능하다는 것을 알 수 있었다.

## V. 참고문헌

- [1] LPWA-Based Long-Term Environmental Data Monitoring System Using IoT Wireless Network 2019.04118 - 119 (2 pages)
- [2] 이동한, 고대식, "A Study on the Design of IoT Network with Edge Computing for Forest Service", 한국정보기술학회, Vol. 16, No. 10, pp.101-109
- [3] Zeba Idrees, Zhuo Zou \* and Lirong Zheng, " Edge Computing Based IoT Architecture for Low Cost Air Pollution Monitoring Systems: A Comprehensive System Analysis, Design Considerations", & evelopment Sensors 2018, 18, 3021; doi:10.3390/s18093021
- [4] E. T. Kim, Y. C. Kim and S. Y. Kwak, "Iot based Indoor Air Quality Monitoring System," The Journal of the Korea Contents Association, vol. 16, no. 2, pp. 143-150, Feb. 2016
- [5] Korea Environment Corporation(KECO). Dictionary of the airkorea [Internet]. Available: [http://www.airkorea.or.kr/dictionary\\_3](http://www.airkorea.or.kr/dictionary_3).
- [6] G. H. Lee, Y. B. Kim and H. S. Kim, "A study of Measuring Position in Real-Time about Particulate Matter Density based on Scattering Method", The Transactions of the Korean Institute of Electrical Engineers, pp. 133-134, 2017
- [7] 윤훈주, 한광인, 조성호, 정병혁, "GIS-based PM10 Concentration Real-time Service", Journal of Korean Society for Atmospheric Environment, Vol. 31, No. 6, pp.585-592
- [8] 노진호, 탁한호, " The Implementation of the fine dust mreasuring system based on internet of things", 한국정보통신학회 논문지, Vol.21.no.4,pp829~835, 2017
- [9] S. Noorulhassan Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog from a Security and Resilience Perspective", IEEE Journal on Selected Areas in Communication, 35(11), p2586-2595, Nov, 2017.
- [10] E. GS, "GS MEC 003 - V1.1.1 - Mobile Edge Computing (MEC); Framework and Reference Architecture".
- [11] <https://www.thefastmode.com/technology-solutions/8577-at-t-to-start-commercial-pilot-on-cat-m1-narrowband-lte-iot-in-november>. [accessed: 12. 07, 2018]

## 저자 소개



황기환(KiHwan Hwang)

1999년 2월 경희대학교 전자공학과(공학박사)  
 2001년 3월 ~ 경북전문대학교 컴퓨터정보과 교수  
 2020년 4월 ~ 현재: 경북전문대학교 소방안전관리과 교수

관심분야 : 클라우드컴퓨팅, IOT, 영상처리, 환경안전