

대한기계학회 주최

제9회 전국학생설계경진대회(2019년)

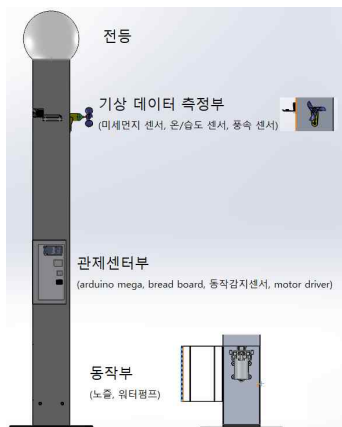
설계 최종 보고서

참가부	대학부 (○)				
참가분야	공모주제 (○) / 자유주제 ()				
참가팀명	가로등불아래 서면				
설계제목	IoT 스마트 가로등				
지도교수/교사	(송실대학교 기계공학부 / 송기영 / mechanical.song@ssu.ac.kr				
대표자 (신청인)	성명	소속	연락처 (휴대폰)	E-mail	주소
	민세웅	송실대학교 기계공학과/4학년		sae0121@mensakorea.org	

참가팀원 인적사항

NO	성명	소속 / 학년	E-MAIL
1	민세웅	송실대학교 기계공학과/4학년	sae0121@mensakorea.org
2	이하준	송실대학교 기계공학과 / 4학년	kia1813@naver.com
3	최병록	송실대학교 기계공학과 / 4학년	chlqudfhr93@naver.com
4			
5			
6			

설계 요약문

참가분야	공모주제 (○) / 자유주제 ()
참가팀명	가로등불 밑에 서면
설계제목	IOT 스마트 가로등
대표자명	민세웅
요약문	다양한 방면으로 기술이 복잡하게 발전하면서 사람들은 기술이 집합된 도시속에서 많은 편의를 누리고 있다. (그러나) 이에 대한 대가로 미세먼지, 폭설, 폭염과 같은 여러 가지 환경문제가 발생하고 있으며 또한 방법, 도로 안전 등에서도 문제가 발생하고 있다. 이러한 복합적인 문제들을 효율적으로 해결하며 더 나은 편의를 (사람들에게) 제공하기 위해 IoT기술을 접목한 ‘IoT 스마트 가로등’을 설계하였다. ‘IoT 스마트 가로등’은 넓게는 이를 통제하는 ‘관제센터’ 그리고 협력관계인 ‘소방서, 기상청등 각 부처’를 포함하는 개념이다. 우선 ‘IoT 스마트 가로등’은 센서부, 제어부, 동작부로 구분이 된다. 센서부는 미세먼지, 온/습도, 우적, 동작감지센서로 구성되며 각각의 센서들이 수집한 데이터는 serial통신을 이용해 가로등을 통제하는 관제센터로 전달되고 이 데이터들은 Windows Forms 하에서 관리가 된다. 동작부인 여름엔 물을 겨울엔 염수를 분사하는 물펌프는 제어부의 통제를 받아 동작한다. 제어부는 아두이노 메가를 사용하였다. 아두이노의 소프트웨어는 아두이노IDE를 이용하여 프로그램을 설계했으며 이를 관제센터에서 아두이노와 Windows Forms의 시리얼통신을 통하여 제어를 한다. ‘IoT 스마트 가로등’은 센서들을 통해 다양한 정보수집이 가능하며 여름엔 온도조절과 겨울엔 제설까지 가능하다. 현재 많은 관심을 받고있는 스마트 도시의 핵심이 되어 앞으로의 미래도시에서 사람들의 편의에 큰 기여를 할 수 있을 것이다. 

스마트IOT 가로등

민세웅, 이하준, 최병록
 숭실대학교 기계공학부

Smart IOT Lamps

Sae Woong Min, Ha Jun Lee, and Byeoung Rok Choi

* School of Mechanical Engineering, Soongsil University

*

(Received september 19, 2019 ; september 19, 2019 ; Accepted september 19, 2019)

Key Words: arduino(아두이노), data mining(데이터 마이닝), IoT(사물인터넷), smart city(스마트시티), street lamp(가로등), windows forms(윈도우폼), big data(빅데이터)

초록: 날씨에 따라 도시환경을 조절할 수 있는 스마트도시시스템이 최근 주목받고 있다. 그러나 현재의 시스템으로 기상문제에 재빠르게 반응하기는 어렵다. 이를 해결하기 위해 곳곳에 설치되어있는 가로등을 이용해 즉각적이고 세분화된 대처가 가능한 ‘IoT 스마트 가로등’을 설계하였다. 마이크로프로세서보드와 컴퓨터간의 통신을 이용해 데이터마이닝하여 날씨정보를 수집해 빅데이터화하고 물/친환경제설액을 분사하는 기계시스템을 제어하여 주변환경을 조절하는 것을 목적으로 설계하였다. 설계의 중점은 IoT 가로등으로 원활한 기상데이터수집 및 저장하여 분석하고 이를 이용해 지역적으로 세분화하여 주변환경을 제어하는 것에 있다.

Abstract: Smart city system that can control the urban environment according to the weather has recently attracted attention. However, with current systems it is difficult to react quickly to weather problems. To solve this problem, we designed “IoT Smart Street Light” that can be immediately and granularly coped with street lamps installed everywhere. It is designed to control the surrounding environment by controlling the mechanical system that collects weather information by data mining using the communication between the microprocessor board and the computer to make big data, and spray water / environmental snow removal solution. The focus of the design is on IoT streetlights to collect, store and analyze weather data smoothly and use it to control the surrounding environment by segmenting it locally.

1. 서 론

2019년 1월 31일 전국에 기습폭설이 내렸다. (이에 따라) 도로 곳곳에서 크고 작은 눈길 사고가 발생했고 시민들은 빙판길에 넘어지며 낙상사고로 인한 인명피해와 일부학교는 휴교 및 등교시간을 늦추는 일이 발생하였다. 특히 부산의 한 어린이집원생 375명은 제설작업이 끝날 때까지 두 시간넘게 발이 묶여있었다. 폭설뿐만 아니라 빗발몰이 지표면 등에 얼어붙는 최악의 결빙현상인 ‘어는 비(freezing rain)’가 내리면 도로가 바로 얼어붙기 때문에 인명사고

† Corresponding Author, sae0121@mensakorea.or.kr

를 유발할 수 있다. 이를 해결하기 위해 <Fig 1>에서 확인할 수 있듯이 겨울철 폭설을 대비하기 위해 24시간 상황실을 운영하며 제설작업을 준비한다. 하지만 이와 같은 노력에도 사고가 잇따라 보다 효율적인 제설방법을 세울 필요가 있다.



Fig 1. 자료사진 뉴스

2019년 여름 서울 기온이 37도까지 치솟으며 이에 따라 도로온도는 50도에 육박했다. 이에 서울시는 폭염기간 8~9월 오후 12시~4시 주요도로 1만4984km에 물청소차 120대를 운영하며 도로 온도에 신경쓰고 있다. 이유는 노인 등 취약계층은 온열질환 관련 인명피해위험이 존재하며 도로면의 복사열과 자동차 등에서 나오는 열로 인해 아스팔트 도로의 변형으로 시민의 안전을 위협할 수 있기 때문이다.



Fig 2. 자료사진 뉴스



Fig 3. 자료사진 뉴스

여름과 겨울에 도로에 발생하는 문제들을 효율적으로 해결하기 위한 수단으로 도로 곳곳에 배치되어 있는 가로등을 생각하였다. 가로등은 몸집이 큰 물분사차와 제설차가 가기 힘든 지역도 관리가 가능하며 가로등에 IoT기술을 이용한 온/습도, 우적, 풍속, 미세먼지, 동작감지센서들 설치하여 데이터를 수집해 빅데이터를 만들 수도 있다. 이를 이용해 기상청, 구청등과 같은 각 부처와 협력한다면 효율적인 도로관리가 가능하게 된다. 즉, 'IoT 스마트 가로등' (이하 IoT 가로등)은 도로환경을 개선해 사람들의 안전과 편의를 증진시키는 것을 목적으로 한다.

1.1. 기상데이터 수집

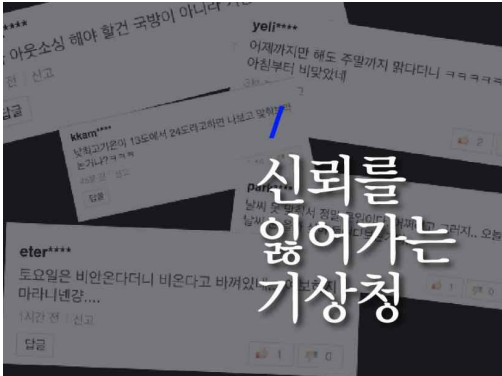


Fig 4. 자료사진 뉴스

기상청에서 제공하는 기상정보는 동/읍/면(예: 상도동) 지역단위로 제공한다. 이러한 기상정보를 이용한 빅데이터는 농림수산, 문화, 체육, 보건, 환경, 교통, 물류, 방재, 기후 등 다양한 분야에서 활용이 되고 있다. 그러나 같은 동/읍/면의 지역에 있더라도 기상의 차이가 발생할 수 있으며 이로 인한 피해가 발생할 수 있다. 이를 해결하고자 곳곳에 설치된 가로등에 기상정보데이터를 수집할 수 있는 센서들을 설치하면 동/읍/면에서 도로명단위의 기상정보를 수집해 활용할 수 있다. 또한 세분화된 기상데이터를 수집해 연구한다면 그림과 같이 신뢰를 잃어가는 기상청의 명예를 회복할 수 있다.

1.2. 방법시스템

밤거리가 불안하다고 생각하는 비율 (단위: %)

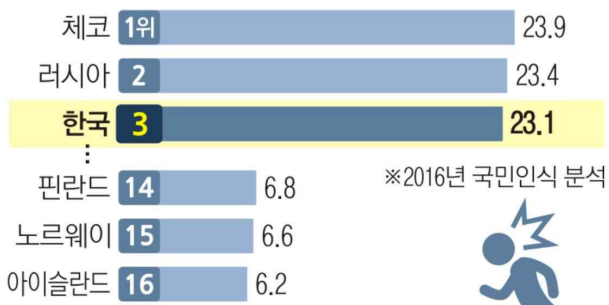


Fig 4. 자료사진 뉴스

그림에서 알 수 있듯이 우리 국민들은 밤거리의 안전에 대해 불안감을 많이 느끼고 있다. 거리 곳곳에 설치된 가로등에 방법버튼을 설치해 작동시 사이렌이 울리며 관제센터에서는 경찰청과 가로등이 위치를 공유하여 관리한다면 국민들의 불안감을 감소시킬 수 있다.

1.3. 에너지 관리

세계적으로 에너지의 효율적인 관리가 중요한 문제이다. 현재 한국은 OECD 34개국 중 에너지효율 측면에서 30위이다. 아래의 그림을 보면 한국은 가정용보다 산업용과 상업용에서의 효율을 높여야 함을 알 수 있다. 서울에만 50만개가량이 있는 일반 가로등을 IoT 가로등으로 교체하여 LED전구를 사용하며 동작감지센서를 활용해 상황에 맞는 밝기를 조절한다면 산업용 에너지의 효율적으로 절감 및 관리가 가능하다.

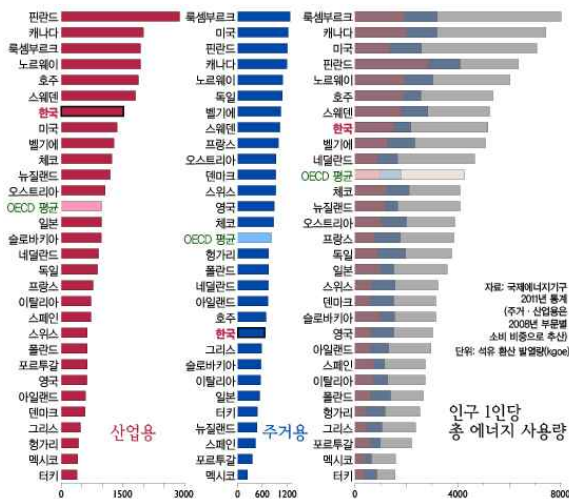


Fig 5. 자료사진

(단위 : 개소/등)

구 분	계	공간조명			광고조명	장식조명
		가로등	보안등	공원조명		
수 량	1,320,048	234,663	226,849	34,119	800,000	24,417

Fig 6. 자료사진

2. 설계핵심내용

2.1 설계 문제의 정의

쾌적하고 안전한 도시를 위해서 기상청, 소방청, 경찰청, 구청 등이 업무를 분담하여 노력하고 있다. 그러나 행정부처들의 만연한 떠넘기기식 행정으로 인해 협력이 필요한 문제들이 제대로 해결되지 않고 있으며 협력하여 국민들의 안전과 편의를 증진시킬 수 있는 방법들이 만들어지지 않고 있다. 이를 해결하기 위해 IoT기술을 이용하여 각 부처들을 유기적으로 연결하면 보다 효율적이고 즉각적인 관리가 가능하게 되며 국민들의 안전과 편의를 더욱 더 증진시킬 수 있다. IoT 가로등은 이를 통제하는 관제센터와 협력관계인 각 부처들을 포함하는 개념으로써 폭설과 폭염에 대한 대처, 센서들로부터 수집된 빅데이터의 활용, 전력 에너지 절감, 방법의 기능들을 수행하며 다양한 문제들을 포괄적으로 관리한다.

2.2 설계의 독창성 및 접근 방법

IoT 가로등은 온/습도, 미세먼지, 우적, 풍속센서들로 기상데이터를 수집하는 기상데이터수집부와 펌프, 노즐, 전구, 방법버튼 및 사이렌으로 이루어진 동작부 그리고 기상데이터수집부와 동작부를 통제하고 관제센터와 각 부처들을 연결하는 제어부로 설계하였다.

앞서 말한 세 가지 부서의 설계방법을 논하기 전에 IoT 가로등의 본체 설계에 대해 논하겠다.

2.2.1 동작부

-친환경제설액과 물 분사

도로교통공단과 기상청의 자료를 보면 적설일 수와 사고 건수는 <그림 2>와 같이 비례하여 나타난다. 회귀분석결과, 월간 강설 일이 1일 늘어날수록 눈에 의한 교통사고는 월간 164건 늘어나고, 월간 하루평균 적설량이 1cm 늘어나면 월간 교통사고는 74건 늘어나는 것으로 나타난다.

매년 겨울철 비싼 제설장비를 구매하고도 사용을 하지 않는 뉴스, 제설예산 부족, 인력부족 등의 이유로 안전사고에 관한 기사가 끊이지 않고 있다. 또 제설차가 투입되더라도 빠른 대처가 이루어지지 않으면 무용지물 되는 경우가 많다. 이처럼 국민의 도로 안전을 위해 가로등에 제설액을 뿌리는 장치를 설치한다.

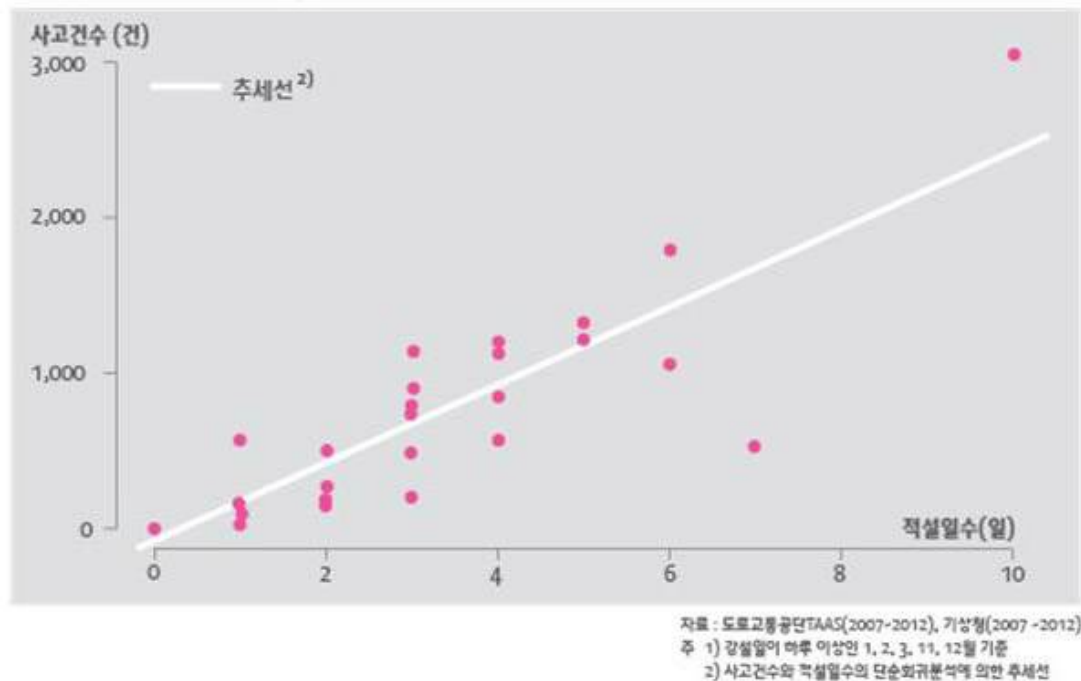
월간 강설일수와 눈 올 때 교통사고 건수¹⁾

Fig 7. 자료사진

2.2.1.1. 설계 배경

물과 친환경제설액을 분사하기 위한 펌프의 종류는 터보형, 용적형, 특수형이 있다. 효율적인 분사를 위해 일정 용적의 유체를 흡입하고 흡입구를 닫아 토출구로 압출하는 방식의 용적형을 선정하였다.

>입구와 출구의 면적을 다르게 하면 유체의 속도를 각각 다르다. 노즐은 출구의 넓이를 입구보다 좁게 하여 출구의 속도를 증가시킨다.

$$Q = AV$$

Q : 유량

A : 면적

V : 유체의 속도

>유량, 넓이, 유체의 속도는 위와같은 식을 따른다. 입출구의 유량은 일정하므로 면적이 좁을수록 속도는 커지게 된다. 노즐은 이를 이용하여 속도를 증가시킨다.

손실

관에서 입구의 에너지와 출구에 에너지는 관의 모양과 재질에 따라 각각 다르다. 이러한 손실을, 주손실과 부손실이라고 부른다. 주손실과 부손실의 식은 아래와 같다.

$$\text{주손실} = f \frac{L}{D_h} \frac{V^2}{2g}$$

$$\text{부손실} = \sum K \frac{V^2}{2g}$$

2.2.1.2. 설계방법

도로관리를 물과 친환경제설액은 최대한 많은 도로면적에 분사되어야 한다. 한 차선의 폭은 약 3m로 3차선까지 분사가 가능하도록 최대분사거리를 10m가 되도록 설계하였다. 또한 노즐의 높이는 보도블럭의 최대높이인 150mm로 설계하였다. 그 이유는 노즐을 높은위치에 설치하여 물을 분사할 경우 IoT 가로등이 설치된 차도와 보행도로에 물이 흩날려 주행자와 보행자에 불편을 초래할 수 있어서 보도블럭의 최대높이로 설계하였다. 또한 설계에 사용할 노즐과 펌프를 선정하기 위한 계산이 필요하다.

공기저항과 바람의 영향은 펌프에 미미할것으로 판단하여 두 변수를 무시하고 $d_{spray} = t \times v$ (l_{spray} : 분사거리, t : 분사시간, v : 분사속력)의 식으로 분사거리를 계산할 수 있다. 여기서 물을 지면과 평행한 방향으로 분사했을 때 시간 t 와 속도 v 는 다음의 식으로 표현이 된다.

$$t = \sqrt{\frac{2h}{g}} \quad (h : \text{분사높이}, g : \text{중력가속도})$$

$$v = \frac{Q}{A} \quad (Q : \text{유량}, A : \text{출구의 넓이})$$

노즐은 많은 면적에 분사하기위해 2개를 사용하므로 출구의 넓이는 $A = 2 \times \frac{\pi}{4} D^2$ (D : 노즐의 직경)로 표현이 된다. 따라서 분사거리는

$$d_{spray} = t \times v = \sqrt{\frac{2h}{g}} \times \frac{Q}{A} = \sqrt{\frac{2h}{g}} \times \frac{Q}{2 \times \frac{\pi}{4} D^2} = \sqrt{\frac{2h}{g}} \times \frac{2Q}{\pi D^2} \text{로 나타낸다.}$$

이를 정리하면 $d_{spray} = \sqrt{\frac{2h}{g}} \times \frac{2Q}{\pi D^2}$ 로 나타낼 수 있다.

노즐의 높이는 보도블럭의 최대높이인

$h = 150\text{mm}$, $g = 9.807\text{m/s}^2$ 으로 설정하였고 펌프의 유량 Q 와 노즐의 직경 D 를 변수로 놓고 펌프의 (4,6,8,10 lpm), 노즐을 설계하면 아래 표와 같다.

Flux(lp m)	4lpm					6lpm				
Diameter(mm)	0.8	1	1.5	2	3	0.8	1	1.5	2	3
Distance(m)	11.60	7.42	3.30	1.86	0.82	17.40	11.13	4.95	2.78	1.24
Flux(lp m)	8lpm					10lpm				
Diameter(mm)	0.8	1	1.5	2	3	0.8	1	1.5	2	3
Distance(m)	23.20	14.85	6.60	3.71	1.65	29.00	18.56	8.25	4.64	2.06

자동차 1차선의 길이는 3~3.5m이고 손실과 공기저항을 고려하였다. PWM제어를 고려했을 때 허용 분사거리를 20m로 잡았다. 식 $d_{spray} = \sqrt{\frac{2h}{g}} \times \frac{2Q}{\pi D^2}$ 을 고려해 위 표를 고려한 결과 8lpm 펌프, 직경 0.8mm 일 때 23.20m, 10lpm펌프, 직경 0.8mm일 때 29.00m로 나왔다. 따라서 허용 분사거리 범위에 있고 경제성을 고려하여 8lpm의 펌프, 0.8mm 직경의 노즐을 구매하였다.

최종적으로 8lpm인 diaphragm pump Pro XTL-3210(용적식 펌프기계)를 선정하였다. XTL-3210(그림x)의 스펙은 아래와 같다.

Voltage	Power(Max)	Pressure(Max)	Flow rate	Pumping head
DC 12V	100W	1.1Mpa	8L/min	70m

2.2.1.3. 설계의 독창성

-High speed, economical, Comfort

기존의 도로환경 관리는 겨울이나 여름철 제설차나 물분사차량으로 직접 도로환경을 관리하는 방식이다. 그러나 기상청으로부터 정보를 받은 관제센터가 IoT 가로등을 활용하면 즉각적으로 도로환경 관리가 가능하며 제설차와 물분사차량이 쉽게 접근하지 못하는 도로의 환경관리가 가능하다. 또한 제설차와 물분사차량을 이용하는 방식보다 경제적으로 도로환경을 관리할 수 있다.

2.2.2 기상 데이터수집부

2.2.2.1. 설계 배경

센서로는 미세먼지센서, 우적센서, 온습도 센서, 동작감지센서를 넣었다. 먼저 동작감지 센서의 경우 펌프가 작동시 사람에게 피해가 가지 않도록 하기 위하여 사용을 하였다. 다른 이유로는 에너지 전략을 위하여 사용하였다. 이는 사람이 없을때는 30퍼 정도의 밝기를 하다가 사람이 지나가면 100퍼의 밝기를 사용하면 된다. 미세먼지 센서는 몇 년동안 이슈가 되어지는 미세먼지를 측정하기 위하여 사용을 하였다. 그리고 이정보를 수집을 하면 특히 어느지역에서 미세먼지농도가 높은지를 확인이 가능하며 그에따라 도시환경을 어떻게 변화를 줄수있다는 기대효과 또한 있다. 우적 센서의 경우 비의 유무만을 확인해주는 역할을 하지만 기상청에서 우적의 예측이 항상 맞는 것이 아니며 이러한 이유 때문에 즉각적으로 현재 비의 유무를 알려줄수 있다는 장점이 있기 때문에 사용을 하였다. 온습도센서의 경우 한국처럼 4계절의 특징이 분명한 지역에서 각 날짜마다 이러한 데이터가 필요하다고 느껴졌으며 가장 큰 이유는 폭염의 상황을 알기 위해서 이다. 이 모든 센서는 가로등의 동작부 제어를 하기위하여 단일 데이터분석이 아니라 여러데이터를 분석을 하여 동작부의 작동을 제어하는것에 중요한 역할을 한다. 또한 정확한 기상정보를 수집하기 위해서는 지열과 같은 외부 요소의 영향을 최소화 하도록 설계해야 한다. 특히 미세먼지센서는 통풍이 잘되어야 하고 우적센서는 지면과 수평이 되도록 설치해야 하므로 지면으로부터 1.7m지점에 공간을 만들어 온/습도, 미세먼지, 우적센서들을 설치하였다. 또한 풍속센서는 바람개비모양의 센서의 공간확보를 위해 IoT 가로등의 우측면에 설치하였다.

각각의 일련번호로 정확한 위치파악이 가능한 IoT 가로등은 설치지역 단위로 세분화된 기후데이터 수집이 가능하다. 수집된 데이터들은 관제센터에 전송되어 빅데이터를 형성한다. 이 빅데이터는 지역별 세부적인 위치를 기반으로한 데이터이므로 현재 농림수산, 문화, 체육, 보건, 환경, 교통, 물류, 방재, 기후등에 사용되는 빅데이터의 질을 높일 수 있다.

2.2.2.2.설계방법

정확한 기상정보를 수집하기 위해서는 지열과 같은 외부 요소의 영향을 최소화 하도록 설계해야 한다. 특히 미세먼지센서는 통풍이 잘되어야 하고 우적센서는 지면과 수평이 되도록 설치해야 하므로 지면으로부터 1.7m지점에 공간을 만들어 온/습도, 미세먼지, 우적센서들을 설치하였다. 또한 풍속센서는 바람개비모양의 센서의 공간확보를 위해 IoT 가로등의 우측면에 설치하였다.

2.2.2.3설계 독창성

-Very easy data collection

서울에만 50만개의 가로등이 설치되어 있다. 이러한 거리 곳곳에 설치된 IoT 가로등을 이용해 보다 손쉽고 빠르게 기상데이터들을 수집할 수 있다.

-Responsibility

기상청에서 예보하는 기상정보들은 지역의 세부적인 기상정보가 아닌 전체적인 기상정보이다. 게다가 지역별로 실제 기상은 이와 다른 경우가 많아 신뢰성이 떨어진다. 이를 도로 곳곳에 있는 IoT 가로등을 활용해 수집한 데이터들로 기상정보를 만들고 IoT기술을 이용해 국민들에게 제공하면 보다 현실적인 정보로써 국민들의 신뢰성을 높일 수 있다.

2.2.3. 관제센터부(프로그래밍)

2.2.3.1설계 배경

마이크로프로세서보드를 이용하여 'IoT 스마트 가로등'을 통제하며 지역별 각각의 고유번호를 가진 가로등이 측정한 데이터값을 효율적으로 축적하고 저장하는 것이 본 설계의 핵심기술이다. 데이터수집을 위해서는 여러 가지 센서들을 사용해야한다. 이를 위해 범용적이고 확장성이 좋은 아두이노를 사용하였다. 통신방법으로 직렬통신방법을 이용하며 이를 통제하기 위해 시리얼통신으로 설계하였다. 또한 'IoT

스마트 가로등'에 효과적으로 명령을 내리기 위해 아두이노메가와 컴퓨터(관제센터)와의 통신은 비주얼베 이직의 윈도우폼즈를 이용하여 어플리케이션을 설계하여 가로등통제를 가능하게 하였다. 윈도우폼즈는 그래픽인터페이스가 잘 구성되어있어 쉽게 컴퓨터용 어플리케이션제작이 가능하며 각각의 기능을 알면 c언어와 c#을 이용하여 어플리케이션을 만드는 것이 가능하기에 선정하였다. 또한 제작자가 프로그램된 내용을 보고 버그등 시스템상의 결함을 수정하기 쉬우며, 보안설계가 가능하여 프로그램의 보안성또한 우수하다.

2.2.3.2 설계 방법

아두이노와 윈도우 폼즈는 기본적으로 c++와 c#언어를 사용하여 프로그램을 설계가 가능하다. 여기서 아두이노와 윈도우폼즈가 서로 통신하면서 일련의 문제가 발생하게 된다. 각 센서가 아두이노에서 측정하여 저장한 데이터들을 윈도우폼즈로 보낼 시 데이터들의 구분이 불가능한 문제가 발생한다. 이를 해결하기 위해 프로그램설계시 데이터들의 범위를 아래와 같이 나누어서 구분이 가능하도록 하였다.

```
68. humi_serial_data=hum+400; //400<=humi<=500사이로 설정
71. temp_serial_data=temp+200; //temp값은 200+-50을 생각하여 저장
89. dust_serial_data=dustDensity; //0부터 100의 값을 준다.
```

윈도우폼즈는 구분이 되지 않는 문자값들로 데이터를 받는다. 이를 위해 각 데이터의 범위를 68번 습도의 경우 400에서 500사이, 71번 온도는 150에서 250사이, 89번 미세먼지는 0에서 100사이의 숫자값의 범위들을 주어 float형태로 소수점까지 표현하게 해주는 변수들로 설계하였다. 숫자값들은 윈도우폼즈로 넘어가서 다시 원래의 데이터값으로 돌려주는 함수를 적용시켜 원래 값을 저장하도록 설계했다.

```
96. rain_serial_data=1000;
102. rain_serial_data=1001;
```

이러한 숫자값들은 윈도우폼즈로 넘어가서 다시 원래값으로 돌려주는 함수를 적용하여 원래값을 저장하도록 만들었지만 우적센서의 경우 우적의 유/무만이 중요하므로 96번과 102번처럼 설계하였다. 아두이노에서 비가오면 HIGH('1')로 비가 안오면 LOW('0')로 표시되도록 간단하게 설계하여 윈도우폼즈에서 원래 데이터값을 다시 구하는 식을 넣지 않도록 하였다. 이러한 설계는 프로그램자체의 무게를 줄이는 효과도 있다.

아두이노와 컴퓨터(관제센터)와의 직렬통신에서는 문제가 있다. 아두이노에 저장되는 데이터값들은 int, float, long(정수, 실수, 실수를 저장하는 명령어)와 같은 형태이다. 그러나 윈도우폼즈에서는 string(모든 형태의 문자열을 저장하는 명령어)형으로 데이터값을 가져와 저장된다.

```
58.string s = myport.ReadLine();
69.float number_1;
70.number_1 = Convert.ToInt32(s);
```

이를 해결하기 위해 58번에서 볼 수 있듯이 's'문자열값을 변수로 선언하였고 69번줄에서 'number_1'로써 실수형 수를 저장하도록 변수를 선언하였다. 70번에서 'Convert.ToInt32(s)'는 문자형으로 저장된 변수를 실수형으로 변환시켜주는 명령어이다. 이렇게 설계하여 먼저 's'에는 아두이노로부터 받은 데이터들을 문자형으로 저장되게하고 'number_1'에는 실수형태로 저장하여 문제를 해결하였다.

앞의과정으로 윈도우폼즈에서 데이터를 숫자열로 바꾸었고 데이터의 범위를 통하여 데이터값을 구분하는것까지 해결하였다. 그러나 여전히 윈도우폼즈에서 아두이노로부터 온 데이터들의 범위를 구분하여 각 범위에 알맞은 위치에 저장을 해야한다. 이를위해 's'라는 변수에 아두이노로부터 온 데이터를 시간에 따라 저장하였다. 전체적인 과정을 아두이노의 프로그램 명령과 윈도우폼즈의 프로그램을 비교해보자면 아래와 같다.

```
90. delay(1000);
```

아두이노에서는 90번의 명령어는 1초를 기다린 후 다음을 진행하도록하는 명령어이다. 이를 통하여 다양한 데이터를 받아야하는 윈도우폼즈에게 시간순차적으로 데이터를 주어 혼선을 피하도록 하였다.

```
59. this.BeginInvoke((new Action(delegate { showValues_1(s); })));
60. this.BeginInvoke((new Action(delegate { pump_control(s,sender,e); })));
61. this.BeginInvoke((new Action(delegate { temp_data(s); })));
62. this.BeginInvoke((new Action(delegate { humi_data(s); })));
63. this.BeginInvoke((new Action(delegate { dust_data(s); })));
64. this.BeginInvoke((new Action(delegate { rain_data(s); })));
```

윈도우폼즈에서는 59~64번의 각 지역함수가 있다. 각 함수에 아두이노에서 받은 데이터값 's'를 할당하

도록 하는 함수이다. 각각의 함수들은 범위에 맞는 역할을 수행한다.

```

94. private void temp_data(string s)//온도값을 표시해주는 함수
    95. {
        96. float number_temp;
        97. float number_temp_1;
98. number_temp = Convert.ToInt32(s); //temp는 150에서 250사이의 값을 갖는다.
    99.
100. if (number_temp >= 150 && number_temp <= 250)
    101. {
        102. number_temp_1 = number_temp - 200;
        103. text_temp.AppendText(number_temp_1 + "\n");
    104. }
    105. }

```

위의 윈도우폼즈 ‘최종’에서 94~105번은 온도값을 표시해주는 지역함수이고 앞서 언급된 문제를 해결을 하는 함수의 하나의 예이다. 데이터 ‘s’를 받아 이를 숫자값으로 변환시킨 뒤에 범위에 맞는 변수에 저장하여 앱상에서 보여주고 데이터값을 저장할 수 있도록 만든 것이다. 이와같이 지역함수에 아두이노로부터 온 데이터값인 ‘s’를 할당하고 할당된 값을 적절한 범위안에 들면 데이터를 보여주도록 설계하였다.

윈도우폼즈에서 아두이노로 보낼 때는 1바이트짜리 문자값만 전송이 가능하다. 그러나 문자값 자체는 아두이노에서 숫자값으로 인식을 할 수가 없으므로 얼마나(시간설정) 어느정도(세기설정)을 원하는지를 아두이노는 알 수가 없다. 이러한 문제는 ‘IoT 스마트 가로등’의 기계적움직임을 제어할 수 없는 문제점이 될 수 있다. 이를 해결할 수 있도록 윈도우폼즈에서 아두이노로 명령을 내리는 프로그램을 설계해야 한다.

```

67.private void pump_control(String s,object sender, EventArgs e) //펌프를 제어하기 위한 함수
    68. {
        69. float number_1;
        70. number_1 = Convert.ToInt32(s);
        71.
72. Timer=new System.Windows.Forms.Timer();
        73. //Timer.Tick += Timer_Tick;
        74.
        75. Timer.Interval = 3000;
        76. Timer.Start();
        77.
78. if(number_1>=233 && number_1<=300)
    79. {
        80. try
        81. {
82. MessageBox.Show("현재가로등COM3지역은폭염입니다.");
        83.
        84. Timer.Stop();
        85. myport.Write("a");
        86. Timer.Start();
        87. }
88. catch(Exception ex4)
    89. {

```

```

90. MessageBox.Show(ex4.Message, "Error");
    91. }
    92. }
    93. }

```

67~93번은 특정조건하에서 펌프를 제어하기위해서 윈도우폼즈가 아두이노를 작동시키는 지역함수이다. 폭염주의보 기준온도인 33도이상으로 온도가 측정될 때 85번에서처럼 윈도우폼즈는 아두이노로 'a'라는 신호를 보내게된다.

```

/*펌프 속도 제어*/
141. if(ch=='a')
    142. {
143. pump_move_loop();
    144. }

166. void pump_move_loop()
    167. {
    168. int i;
169. digitalWrite(pump_direction,HIGH); //모터의 방향제어
    170. for(i=0;i++;i<510)
    171. {
    172. int motor_speed;
173. motor_speed=-8*i+8*i+50;//펌프가 속도가 빨라졌다가 느려지는 식을 특정 변수에 저장을
        한다.
    174. analogWrite(pump,motor_speed);
    175. delay(300); //0.3초에 i=i+1의 식을 행하여 펌프속도를 제어한다.
176. if(sensor==true) //사람이 지나가면 펌프는 물을 뿌리는것을 멈춘다.
    177. {
    178. analogWrite(pump,0);
    179. delay(3000);
    180. }
    181. }
    182. }

```

아두이노는 'a'라는 신호를 받으면 아두이노 자체에 설계되어있는 프로그램을 실행하도록 설계되어있다. 즉 특정한 문자값이 오면 그에대응하는 조건을 적용하여 기계제어가 되도록 한 것이다. 141~144번은 신호를 읽는 명령어이고 166~181번은 펌프를 제어하는 지역함수이다. 펌프제어는 사람이 지나갈 때 물을 분사하지 않도록 동작감지센서의 데이터를 이용해 사람이 지나갈 때 멈추도록 프로그램을 설계하였다.

2.2.3.3.설계 독창성

첫째, 아두이노와 윈도우폼즈사이의 통신관련 문제를 'IoT 스마트 가로등'의 센서로부터 수집된 데이터를 숫자값으로 저장하고 이를 숫자값 그대로 윈도우폼즈로 보낸다. 이때 윈도우폼즈는 아두이노로부터 받은 숫자값의 데이터를 문자값으로만 저장함으로 우리가 원하는 데이터저장방식에서 벗어난다. 이러한 문제를 해결하기 위해 아두이노와 윈도우폼즈에서 각각에 맞게 데이터를 처리하도록 설계하였다. 전송되어지는 데이터들을 특정한 범위로 분류하고 아두이노에서는 데이터들을 각 범위에 맞게 데이터값을

변환한다. 예를 들어 온도값은 150~250사이의 값으로 정하였고 아두이노는 이러한 범위값으로 수집된 숫자값의 데이터를 윈도우폼즈로 전송한다. 온도값이 숫자값 225로 되었다면 윈도우폼즈는 본 설계에 의해 문자값으로 인식하지 않고 숫자값으로 변환해 이 데이터가 150~250사이의 온도값이라는 것을 확인한다. 그 다음 이 값을 원래의 값으로 돌리기 위해 150에서 250사이의 데이터값은 -200을 하여 원래 값으로 되돌려주는 프로그램설계를 설계하여 최종데이터값인 25를 온도값으로 표시하도록 하는 방법을 사용하였다.

둘째, 'IoT 스마트 가로등'에서 수집한 데이터들을 아두이노는 한번에 관제센터로 전송을 한다. 이때 윈도우폼즈에서 데이터값을 읽는데 혼선이 생기게된다. 이를 해결하기 위해 윈도우폼즈에서는 데이터를 받는 시간간격을 5초간격으로 설계하고 아두이노에서는 1초간격으로 데이터를 보내는 방식으로 설계를 한다. 이러한 설계를 통해 윈도우폼즈에서는 5초간 1초간격으로 각 데이터들을 순차적으로 표시해줄 수 있다.

셋째, IoT를 이용하여 사람들의 편의를 증진시키기 위해 여름엔 물을, 겨울엔 친환경제설액을 분사한다. 물펌프의 동작을 제어하는데 있어서 여러 조건을 설계했다. 물을 분사함에 있어서 관제센터는 기상청과의 협력하에 폭염주의보/경보의 조건인 33도/35도가 되면 물을 분사하도록 한다. 친환경제설액을 분사함에 있어서는 대설주의보/경보의 조건인 신적설이 5cm/20cm이상일 때 분사하도록한다. 이때 도로상황에 맞게 분사할 수 있도록 동작감지센서를 이용해 도로상황을 확인한 후 분사하도록 하였다. 또한 전체 도로에 골고루 분사되도록 펌프의 세기를 약한신호부터 강한신호까지 점차적으로 증가시켜 분사하도록 하였다. 물/친환경제설액의 분사량은 시간을 통제를 한다. 윈도우폼즈는 각 시간에 할당된 1byte 문자값을 전송하고 아두이노는 특정 문자값에 해당하는 함수를 실행하여 얼마간의 시간동안 분사할지 결정한다.

2.2.3. 설계의 제약조건 및 문제 해결 방법

2.2.3.1. 안전성

풍력센서는 바람개비 형태의 날개가 회전하며 데이터를 수집한다. 이를 고려하여 사람의 손이 닿지 않도록 IoT 가로등 우측면 위쪽에 부착하여 사람들의 안전성을 고려하였다. 또한 기존의 제설제인 염화칼슘은 눈을 빠르게 녹이지만 도로와 자동차의 부식, 환경오염등의 부정적인 측면이 있다. 이를 해결하기 위해 친환경제설액을 사용하였다. 물/친환경제설액을 분사하는 과정에서는 동작감지센서로 전방의 상황을 파악해 분사를 작동하거나 멈추도록 설계하여 도로위 사람들의 안전을 고려하였다.

2.2.3.2. 효율성/편의성

곳곳에 설치된 가로등에 기상측정을 위한 센서들을 설치하여 효율적이고 경제적으로 기상데이터를 수집할 수 있다. 또한 몸집이 큰 제설차/물분사차량은 가파르거나 좁은 도로 및 골목길에 접근이 어렵다. 이를 IoT가로등을 이용한다면 효율적이고 편리하게 도로환경을 개선할 수 있다.

2.3. 설계 내용

IoT 가로등은 기상데이터를 수집하는 기상데이터수집부와 도로환경을 위해 직접적인 활동을 하는 동작부, 이를 통제하는 제어부로 설계하였다. IoT 가로등의 최상단에 전구를 설치하였고 기상데이터수집부는 효율적인 데이터수집을 위하여 상단에 설치하였다. 동작부는 분사시의 상황을 고려하여 하단부에 설치하였고 제어부는 전선배치를 고려하여 중앙쪽에 설계하였다. 전체적인 설계도는 그림에서 확인할 수 있다.

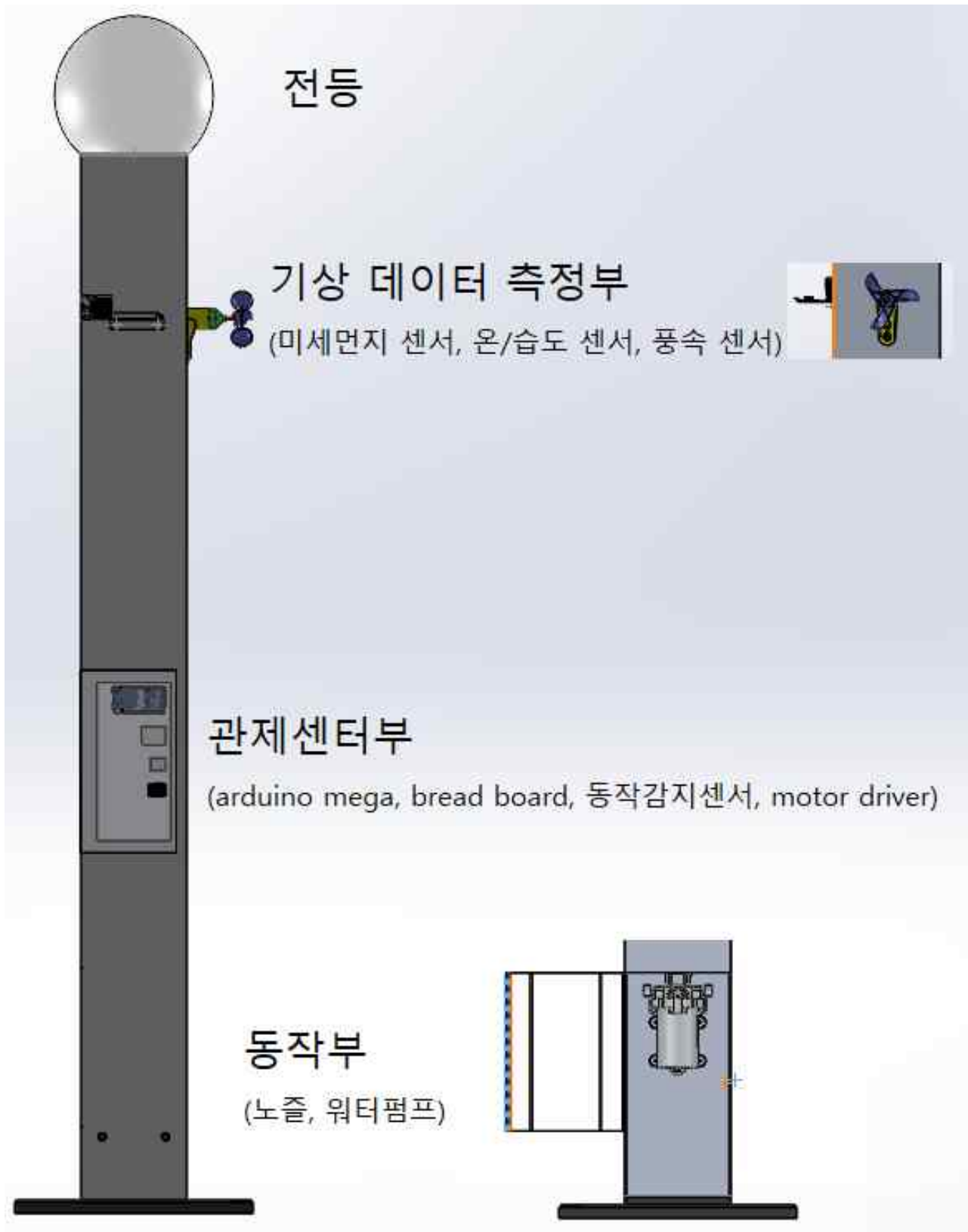


Fig 9.

2.3.1. 기상데이터수집부

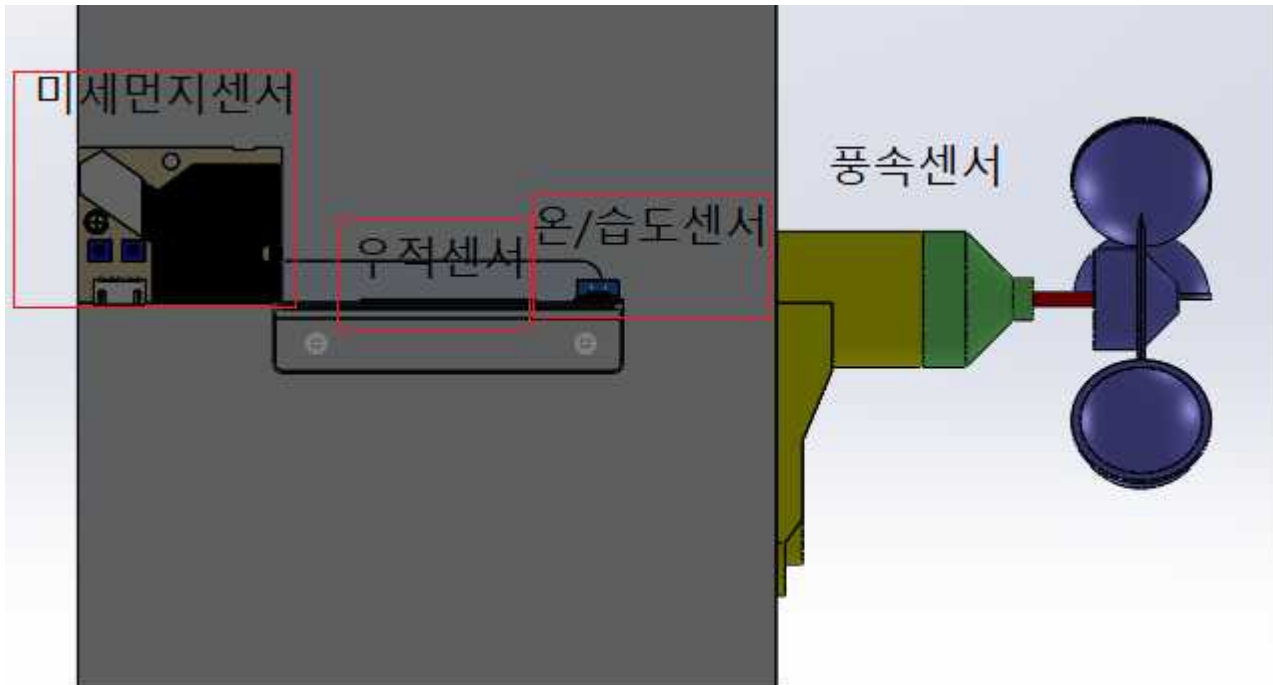


Fig 10.

기상데이터수집부에는 온/습도, 미세먼지, 풍속, 우적센서가 설치되어있다. 정확한 기상정보데이터수집을 위해 외부영향을 최소화하며 사람의 손이 닿지 않도록 지면으로부터 IoT 가로등의 1.7m지점에 설치하였다. 외부영향이란 온/습도센서에 영향을 줄 수 있는 지열이 있다. 또한 원활한 통풍을 통해 정보를 수집해야하는 풍속센서와 미세먼지센서가 지면과 가깝게 설치되어 있을 시에 영향을 받을 수 있다. 마찬가지로 우적센서도 지면과 가깝게 설치되면 우적인 아닌 다른 외부요인으로 인해 작동하게 되는 영향을 받을 수 있다. 이를 고려하여 1.7m의 높이를 선정하였고 부피가 크고 바람을 잘 맞아야하는 풍속센서는 IoT 가로등의 우측면에 설치하였다.

2.3.2. 제어부

제어부는 동작부와 기상데이터수집부와 유선으로 연결되어 이들을 통제한다. 제어부와 각부를 연결하는 배선문제를 고려하여 공간을 효율적으로 사용하도록 지면으로부터 705mm지점에 직사각형 모양의 아크릴판을 배치해 제어부를 설치하였다. 이 아크릴판에 arduino mega, bread board, motor driver, 동작감지센서를 설치하기위해 아크릴판의 크기는 가로 140mm, 세로 300mm로 설계하였고 배치는 그림과 같이 하였다. 또한 IoT 가로등에 문제가 발생할 시에 수리에 용이하도록 아크릴판을 미닫이 문 형태로 설계하였다.



Fig 11.

IoT 가로등은 아래와 같은 언어를 사용하여 제어하였다.

```

1. #include <DHT.h>
2. #include <DHT_U.h>
3.
4. /*미세먼지감지 센서 지정*/
5. int V0=A1; //미세먼지 감지를 A1으로 해준다.
6. int V_LED=5; //미세먼지에 있는 LED을 5번핀으로 지정해준다.
7.           //이 LED가 미세먼지양을 측정을 해주는 LED이다. LED가 켜지면 반대편에 있는
수신소자가 반사되는 미세먼지양을 인식해주는 구조이다.
8.           //적외선 LED을 켜고 끄는데 10ms가 소요된다.
9.           //0.32ms동안 켜고, 9.68ms동안 led을 꺼야한다.
10.          //이 동작 반복
11. float V0_value=0; //먼저 켜졌을때의 미세먼지 측정값을 0으로 맞춰준다.
12. float V0_voltage=0; //미세먼지에 걸리는 5v값을 측정하기 위하여 지정해준다.
13. float dustDensity=0; //미세먼지양을 측정하기 위하여 지정
14.
15. /*온습도 센서 지정*/
16. int temp_hum=4; //온습도 센서를 4번핀으로 지정
17. #define DHTTYPE DHT11 //DHT타입을 11로 정의한다.
18. DHT dht(temp_hum,DHTTYPE); //dht설정 -dht(디지털신호4핀,dht11)
19.
20. /*사람감지 센서 지정*/
21. int motion=3; //인체감지 센서 3번 핀으로 설정
22.
23. /*펌프*/
24. int pump=13; //13번 핀은 펌프 속도제어
25. int pump_direction=12; //12번 핀은 펌프의 방향을 제어하기 위하여 사용한다.
26. int LED_red=2; //임시확인을 위한 LED
27.
28. /*우적센서 지정*/
29. int rainsensor_pin=A0; //우적센서를 Analog0로 지정
30.
31. void setup() {
32.   Serial.begin(9600);
33.
34.   pinMode(V_LED,OUTPUT);
35.   pinMode(V0,INPUT); //미세먼지 감지 input
36.   pinMode(LED_BUILTIN, OUTPUT);
37.
38.   pinMode(pump_direction,OUTPUT); //펌프 방향 output
39.   pinMode(pump,OUTPUT); //펌프 속도 output
40.
41.   pinMode(motion,INPUT); //사람감지 센서를 지정한다.

```

```

42. pinMode(LED_red,OUTPUT);
43. }
44.
45. void loop() {
46.   char ch=Serial.read();//windows forms에서 받는 모든 값을 여기서 받는다.
47.
48.   /*시간에 따른 등주제어*/
49.   if(ch=='b')
50.   {
51.     analogWrite(LED_red,250);//나중에 시간값을 받아오면 사용
52.   }
53.   if(ch=='c')
54.   {
55.     analogWrite(LED_red,0);
56.   }
57.   /*사람감지 센서*/
58.   int sensor=digitalRead(motion); //사람감지 센서
59.   if(sensor==true) //사람감지에 따라서 펌프또한 멈추게 한다.
60.   {
61.     analogWrite(LED_red,250*30/100);
62.     delay(300000);
63.   }
64.   /*온습도 센싱와 시리얼 통신*/
65.   float temp_serial_data; //원도우폼에 보낼 온도값
66.   float humi_serial_data; //습도값
67.   float hum=dht.readHumidity(); //습도값을 hum에 저장을 한다.
68.   humi_serial_data=hum+400; //400<=humi<=500사이로 설정
69.   Serial.print(humi_serial_data);
70.   float temp=dht.readTemperature(); //온도값을 temp에 저장을 한다.
71.   temp_serial_data=temp+200; //temp값은 200+-50을 생각하여 저장
72.   Serial.print(temp_serial_data);//150<temp<250의 값을 가진다.
73.
74.
75.   float dust_serial_data; //미세먼지
76.   char rain_serial_data; //
77.
78.   /*미세먼지 감지*/
79.   digitalWrite(V_LED,LOW);
80.   delay(280000); //dalay는 ms를 나타내고 delayMicrosecond는 us을 나타낸다. 1000ms=1us
81.   V0_value=analogRead(A1);
82.   delay(40000);
83.   digitalWrite(V_LED,HIGH);
84.   delay(9680000);
85.

```

```
86.  V0_voltage=V0_value*5.0/1024.0; //미세먼지를 5v로 환산하여 표시를 해 준다.
87.  dustDensity=(V0_voltage-0.3)/0.005; //미세먼지양을 측정하기 위한식
88.  dust_serial_data=dustDensity; //0부터 100의 값을 준다.
89.  Serial.print(dust_serial_data); //0<dustDensity<100 어차피 퍼센트 값이기 때문에 이 범위 안에서 해결가능
90.  delay(1000);
91.
92.
93.  /*우적센서*/
94.  if(analogRead(A0)<500) //우적센서의 센싱값을 우선은 LED_BUILTIN으로 표시를 해 준다.
95.  {
96.      rain_serial_data=1000;
97.      Serial.print(rain_serial_data);
98.      digitalWrite(LED_BUILTIN,HIGH);
99.  }
100. else if(analogRead(A0)>500)
101. {
102.     rain_serial_data=1001;
103.     Serial.print(rain_serial_data);
104.     digitalWrite(LED_BUILTIN,LOW);
105. }
106.
107. /*펌프 속도 제어*/
108. int i;
109. if(ch=='1')
110. {
111.     digitalWrite(pump_direction,HIGH);
112.     analogWrite(pump,250);
113.     if(sensor==true)
114.     {
115.         analogWrite(pump,0);
116.     }
117.     delay(5000);
118. }
119. }
```

위와 같은 방식으로 기상관측을 하게 되면 <그림>와 같이시간에 따른 온도, 습도, 미세먼지농도, 우적유무를 관제센터에 전송하게 된다.



Fig 12. windows forms application

2.3.3. 동작부



Fig 13. 동작부 도면

동작부는 그림과 같이 설치하였다. IoT 가로등 정면에는 노즐을 배치하고 후면에 펌프를 배치하였다. 노즐은 지면으로부터 145mm높이에 왼쪽에서부터 17mm, 오른쪽에서부터 17mm 이격시킨거리에 노즐을 고정시킬 수 있도록 구멍으로 만들어 설치하였고 진동을 고려하여 너트를 이용해 고정시켰다. 펌프는 IoT 가로등 내부에 위치하며 물이 고이지 않도록 지면으로부터 27cm높이에 설치하여 물을 조달하는 배관이 최대한 꺾이지 않도록 설계하여 손실을 최소화 하였다.

3. 설계 수행 일정

설계 진행 내용	4월	5월	6월	7월	8월	9월	10월	11월
브레인 스토밍 및 설계 계획	■							
자료 조사, 주제선정	■							
아이디어 구체화	■							
설계제안서 작성		■						
설계 계획 수정 및 보완			■					
부품 선정 및 구입			■					
중간보고서 작성			■					
프로토 타입 제작				■				
‘IoT 스마트 가로등’ 제작					■			
제작 수정						■		
최종 보고서 작성						■		
문제점 파악 및 개선							■	■

4. 설계 결과물

그림은 IoT 가로등의 완성된 모습이다. 본 장에서는 최종적으로 설계된 기상데이터수집부, 동작부, 제어부에 대해 논하겠다.

4.1동작부

관제센터에서 아두이노를 통해 펌프를 가동시키게 되면 펌프는 유체를 흡입하여 노즐을 통해 나오게 된다. 펌프의 세기를 PWM을 통해 제어한다.

-동작부

펌프와 호스, 관은 아래 사진과 같이 되어있다. 펌프의 왼쪽에서 유체가 흡입되어 오른쪽으로 유체가 나오게 되어있다. 이후 호스가 연결되고 Y자 분배기로 유체를 각각 분배하였다.



Fig 14. 완성된 모습
좌, 우측 각각 고무호스-신주-관-노즐 순으로 연결하였다. 각 연결부에는 호스밴드, 테프론 테이프를 사용하여 물이 새지 않게 설치하였다.



Fig 15. 펌프

최종 설계된 8l/min의 펌프의 실제 유량은 2L의 물을 몇 초 동안 토출해내는지로 측정하였다. 이 결과는 아래 표와 같다.

유량실험		
2L	t(s)	유량(lpm)
1차	43	2.79
2차	42	2.86
3차	45	2.67
4차	43	2.79
5차	45	2.67
평균	44	2.71
표준편차	1.20	0.0756

평균 2.71lpm의 유량이 측정 되었다. 오차율은 66%로 굉장히 큰 값이 나왔다.

1분동안 좌, 우측의 노즐을 통해 얼마나 나오는지 각각의 유량을 측정해본 결과 아래 표와 같다.

좌/우측 유량 실험		
1min	좌측(l)	우측(l)
1차	1.41	1.29
2차	1.43	1.24
3차	1.43	1.37
4차	1.42	1.48
5차	1.4	1.4
평균	1.42	1.36

평균 유량은 좌측 1.42lpm, 우측 1.36lpm 으로 좌측이 더 유량이 많이 나오는 것을 확인할 수 있었다.

분사거리실험(m)	
1차	6.12
2차	5.91
3차	6.15
4차	5.95
5차	5.9
평균	6.006
표준편차	0.0977

실제적인 평균 분사거리는 6.006m로 결과가 나왔다. 실험을 하기 전 이론상 거리인 23.20m보다 작은 값이 나왔다. 이는 유량의 오차율 때문이라고 생각된다. 8lpm의 펌프를 구매하였지만 2.71lpm의 유량이 측정되었다. 초기에는 구조적인 설계문제로 판단하고 조사해보았지만 문제를 찾을 수 없었다. 그래서 측정된 유량 2.71lpm를 대입

하여 $d_{spray} = \sqrt{\frac{2h}{g}} \times \frac{2Q}{\pi D^2}$ 식을 이용하면 7.86m의 분사거리가 나왔다. 실험 평균

6.006m과 오차율은 -23.6%로써 기존의 8lpm펌프의 이론값보다 오차율이 적게 나타났다. 시간관계상 펌프를 교체해 분사해보지 못했지만 펌프 자체의 불량문제로 생각된다. 이 문제를 해결하기 위해 여러 스펙의 펌프를 구매해 분사거리를 측정해볼 것이다. 결론적으로 이론적으로 예상한 거리보다 작은 값이 나왔지만 최종적으로 측정된 최대분사거리는 6m로써 2차선도로까지는 충분히 분사가 가능하다. 또한 분사시 사람들에게 피해가 가지 않도록 설계한 동작감지센서도 그림에서와 같이 사람이 감지되면 분사를 멈추는 것을 확인할 수 있다.

4.2 기상데이터수집부와 제어부



Fig 17. 아두이노 센서부

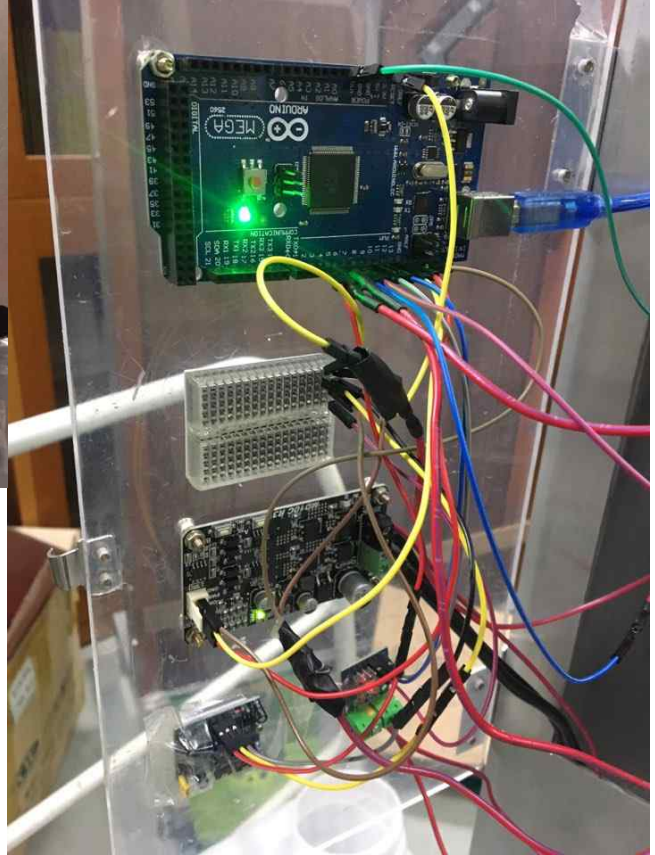


fig 18. 아두이노 제어부

현재시간	온도값	습도값	미세먼지농도	우박유무
20:01:23	21.21	52.01	25	X
20:01:28	21.21	52.01	25	X
20:01:33	21.18	52.02	25	X
20:01:38	21.19	52.02	25	O
20:01:43	21.20	52.02	25	O
20:01:48	21.18	52.02	25	O
20:01:53	21.18	52.02	25	O
20:01:58	21.18	52.02	25	O
20:02:03	21.18	52.01	25	O
20:02:08	21.18	52.01	25	X
20:02:13	21.18	52.01	25	X
20:02:18	21.18	52.01	25	X
20:02:23	21.18	52.01	25	X
20:02:28	21.18	52.01	25	X
20:02:33	21.18	52.01	25	X
20:02:38	21.18	52.01	25	X

Fig 19. Window Forms 앱

이노가 받은 데이터는 다시 관제센터로 보내준다. 관제센터는 받은 데이터를 토대로 하여 현재 지역의 날씨상태를 실시간 파악을 한다. 파악한 날씨상태를 토대로 관제센터는 가로등에게 어떠한 행동을 할 것인지를 명령을 내려준다. 가로등은 명령을 받고 동작부에 어떠한 행동을 할 것인지를 명령해준다.

그림 16는 현 시스템의 간단한 관계도를 나타낸다. 센서는 자신이 측정한 값을 아두이노(스마트 IOT가로등)로 전송을 한다. 아두

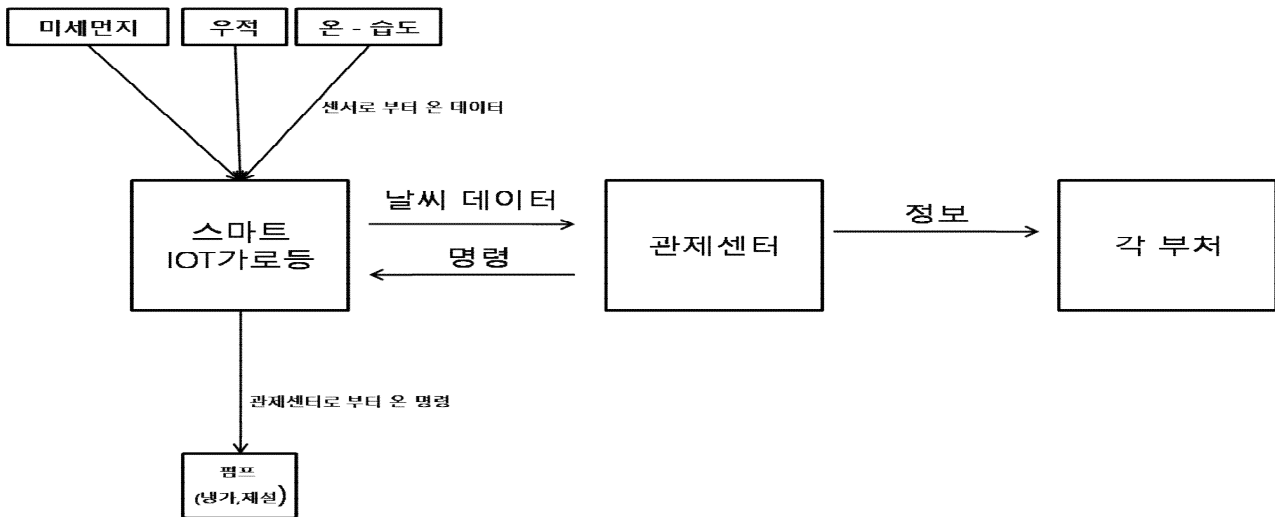


Fig 16. 통제부 알고리즘 정리

그림18는 아두이노의 운용모습이다. 각 센서에 전선을 연결하여 아두이노에 연결되어 있는 모습이다. 그림17은 센서가 운용되는 모습이다. 왼쪽부터 미세먼지 센서, 우적센서 그리고 온습도 센서이다. 미세먼지 센서는 내부에 적외선 감지기와 적외선 발생기가 존재한다. 검은색 부분은 비어있는 공간으로 공기가 지나가면 미세먼지가 같이 지나가고 적외선 발생기에서 적외선이 나와 미세먼지에 의하여 그림자가 지어지게 되고, 적외선 감지기는 적외선이 오는 양을 측정하여 미세먼지 데이터를 수집한다. 다음으로 우적센서는 센서와 쉴드로 이루어져있고, 쉴드는 센서가 물에 젖었을 때 쇼트가 나는 것을 방지한다. 센서는 물에 젖었을 때 전기선에서 저항이 0이 되고 이로인해 전기선에 지나가는 전류량이 무한히 높아지는 것을 측정을 하여 비의 유무를 확인한다. 다른 센서와 다르게 비의 유무만 파악할뿐 비의 양을 측정해주지는 못한다. 온습도센서는 먼저 습도감지와 온도감지가 내부에 따로 설계되어있다. 습도는 두 전극사이의 전기저항을 측정하여 수증기를 감지한다. 온도의 경우 NTC온도센서로 온도값을 측정한다. 하지만 아두이노에는 한가지 전선만 연결이 되어있고, 온습도는 하나의 전선으로만 그 값을 받는다.

그림19은 비주얼베이직에 있는 윈도우폼이라는 프로젝트를 사용하여 언어는 c#을 이용한 최종 모습이다. 현재 5초간격으로 각 데이터를 측정한 값이며 오른쪽 밑에 있는 Save Data라는 버튼으로 데이터 저장이 가능하다. 데이터는 '.Text'형식으로 저장이 된다.

현 시스템은 날씨정보와 데이터를 세부적이고 즉각적으로 수집하는것이 큰 장점이다. 이를 토대로 주변환경을 원격으로 제어가 가능하며 대규모 날씨정보를 저장하기 때문에 먼 미래에 날씨를 예측하는것도 또한 가능하게 될 것이다. 날씨정보를 토대로 즉각적인 대응이 가능하다는 점이 이러한 설계의 필요성을 증명한다. 예를 들어 폭염과 같은 상황에서 도심에 있는 횡단보도에 파라솔을 설치 하는것만으로는 더위를 피하기 힘들다. 하지만 IoT 가로등을 이용해 즉각적으로 물을 분사한다면 폭염에 대처할 수 있다. 그리고 현재 고속도로에서 이러한 폭염에 대응을 하여 도로중앙에서 물을 뿌리는 시스템에 응용될 수 있다.

하지만 완성된 시스템의 문제도 존재한다. 첫 번째로 펌프동작을 위해 펌프가 움직이는 시간을 제어한다. 하지만 이 시간은 임의로 명령을 내릴수 없고 앱에 입력이 되어 있는 시간으로만 제어가 가능하다. 두 번째로 데이터의 혼선을 막기 위하여 데이터를

5초간격으로 오게 프로그램하였다. 이로 인해 5초 이내의 세부데이터수집은 불가능하다. 세 번째는 원격으로 제어가 가능하지만 무선통신이 아닌 유선통신인 점이다. 아두이노에 와이파이모듈을 사용하여 제어를 시도했지만 알 수 없는 오류로 인해 유선모드로 통신을 설계하였다. 네 번째는 아두이노와 펌프, 전구에 각각 별도의 전원이 필요한 점이다. 아두이노에는 5~9v, 펌프에는 12~15v, 전구에는 9v이상의 전압을 사용하기에 이러한 문제가 발생하였다.

위의 네가지 문제에도 불구하고 IoT 가로등은 의미가 있는 기계설계시스템이다. 원격으로 제어가 가능하며 원거리로 데이터를 주고받으며 수집 및 저장하여 도로환경을 개선한다. 또한 관제센터를 만들어 각부처와의 협력을 유도할 수 있도록 하여 국민들의 편의와 안전을 위한 효과적인 도로개선이 가능하다.

5. 결론

IoT 스마트 가로등은 IoT가로등과 관제센터, 기상청, 소방청과 같은 각 부처들의 유기적인 연결을 포함하는 개념으로써 기상데이터수집부, 동작부, 제어부로 구분되어 국민들의 안전과 편의를 위한 도로환경 개선을 위한 설계품이다. 최종설계물을 실제 도로에 적용한다면 운용에 있어서 크고 작은 문제가 발생할 것이다. 그러나 이 문제들은 시간이 지남에 따라 해결될 것이며 이후 기대되는 앞서 말한 본 설계의 긍정적인 효과들은 국민들의 안전과 편의를 극대화 할것으로 기대된다.

5. 활용방안 및 기대효과

IoT 가로등으로 지역 곳곳에서 수집한 데이터들을 빅데이터화하여 적용 할 수 있는 분야는 무궁무진하다. 우리가 설계한 온/습도, 미세먼지, 풍속 이외에 지진, 조도, 강수량, 적설양, 염도, 자외선양 등 지역 특색에 따라 다양하게 적용될수 있다.

농림,수산에서는 농작물 주산지의 날씨정보, 생육시기별 기상정보, 농산물 생산성 예측, 서리 발생 가능성(서리는 농작물에 치명적인 영향을 미친다.) 월별 오징어 어획량 예측 정보, 바다의 기상정보를 알 수 있다.

관광에서는 눈이나 비가 올때 관광지를 추천하는 방안으로 활용될수 있다.

적조 발생가능성을 예측하고 과거 이력을 확인할수 있다.

교통에서는 기상변화에 따른 교통사고 위험 전망을 알 수 있고과거 이력과 기상정보를 대조하여 분석할수 있다.

기상정보인 호우나 폭설을 예측할수 있고 이로인해 피해 이력 정보를 조회할수 있다.

가로등에 다양한 측정장치를 설치함으로써 손쉽게 데이터를 수집하여 적용할수 있다.

6. 참고문헌

2. 참고문헌

주식회사 세진기술개발, (2006) 개량형 이동식 염수분사장치, 특허청(kpris)

주식회사 에스알디코리아, (2016) 염수분사시스템, 특허청(kpris)

주식회사 자동기,(2011) 도로제설을 위한 염화칼슘용액 제조용기의 자동세척 장치,특허청

전공자를 위한 C 언어 프로그래밍, (2018), 주우석, 한빛아카데미

[폭설내린 동해안 도로 제설 안돼 운전자 '불편'] 중앙일보 2019.1.31

<https://news.join.com/article/23342478>

현재 겨울 폭설시 제설에 대한 대처

[국토교통부, 폭설에 대비한 24시간 상황실 운영] 한국교통공단 블로그 2015.12.3

<https://blog.naver.com/autolog/220558054634>

월간 강설일수와 눈 올 때 교통사고 건수

[눈이 하루 더 오면 월간 교통사고 164건 증가] 서울타임즈 2014.2.5

<http://www.seoultimes.net/news/articleView.html?idxno=23673>

서울시 미세먼지

[맑던 하늘이 다시 뿌옇게...겨울철 불청객 미세먼지 공습 재개] SBS 뉴스 2018.10.16

https://news.sbs.co.kr/news/endPage.do?news_id=N1004974723&plink=COPYPASTE&cooper=BSNEWSSEND

OECD 주요국 미세먼지 농도

[한국 미세먼지 농도 날로 급증...OECD 국가 중 최악 수준] 연합뉴스 2017.2.16

<https://www.yna.co.kr/view/AKR20170215159600009>

여성 안심 귀갓길

["여성 여러분, 더 안심하고 귀가하세요"]노컷뉴스 2016.6.17

<https://www.nocutnews.co.kr/news/4609562>

[제설장비 예산 턱없이 부족] 노컷뉴스 2013.1.2

<https://www.nocutnews.co.kr/news/994654>

폭설에 대기 중인 제설차

["눈 와도 걱정 없다" 성남시 설해 제설 장비 사전 정비·점검을 완료해 대기 중]한국방송뉴스 2018.11.16

<http://www.ikbn.news/news/article.html?no=65726>

[눈 와도 제설장비 주차장서 낮잠] YTN 2013.12.29

https://www.ytn.co.kr/_ln/0115_201312290010031012

[강원도 폭설, 각종 피해에 지자체 예산 집행] 2014.02.14

<http://news1.kr/articles/?1533227>

[출근길 쏟아진 눈, 정체된 도로서 제설차도 제 역할 못 해] KFM 2018.12.13

원문보기: <https://www.kfm.co.kr/?m=blog&blog=news&uid=9330209&cat=>

[춘천시, '자동 염수분사장치' 도입] 뉴스1코리아 2016.11.10

<http://news1.kr/articles/?2827698>

[기상청 날씨마루] (빅데이터 활용 참고)

<https://bd.kma.go.kr/>

[눈이 오면 왜 염화칼슘을 뿌릴까?...염화칼슘의 '두 얼굴']SBS 뉴스 2017.01.21

https://news.sbs.co.kr/news/endPage.do?news_id=N1004002594&plink=COPYPASTE&cooper=BSNEWSSEND

[제설작업 대기 17시간째, 드디어 명령이 떨어졌다] 중앙일보 2010.01.16

<https://news.joins.com/article/3971261>

[제설작업 늦장 주민불만 고조] 중앙일보 19.03.01

http://www.cktimes.net/board_read.asp?boardCode=board_international&boardNumber=6269

[에너지효율관리의 중요성] 한국에너지 2012.11.26.

<http://www.koenergy.co.kr/news/articleView.html?idxno=66359>

[한국, 에너지 효율 OECD 34개국 중 30위] 한겨레 2013.08.30

http://www.hani.co.kr/arti/society/society_general/601537.html

[종합 국회전국 가로등 절반가량 '불량']에너지 time뉴스 2017.9.21

<http://www.enertopianews.co.kr/news/articleView.html?idxno=1528>

[과학백과사전 잠열]사이언스올

<https://www.scienceall.com/%ec%9e%a0%ec%97%b4latent-heat/>

[서울시, 폭염기간 ‘물 청소차’ 집중운영...도로온도 6.4℃ 낮춘다]

<http://www.newspim.com/news/view/20190812000801> 뉴스핌 2019.08.12.

[서울 올 최고 37℃, 도로 표면 온도가 무려...] YTN 2019.08.10.

https://www.ytn.co.kr/_ln/0108_201908101154198846

[기상청, 여름철 살수차·쿨링포그 효과 규명 나서] 스카이드일리 2019.08.05

http://www.skyedaily.com/news/news_view.html?ID=88210

[전국 기습 폭설에 '화들짝'...눈길·빙판에 '설설'기고 사고 속출] 연합뉴스 2019.01.31.

<https://www.yna.co.kr/view/AKR20190131197700062?input=1195m>

[‘어느비’ 예측으로 빙판길 사고 감소 시킨다] 라이드매거진 2016.11.22.

<https://m.post.naver.com/viewer/postView.nhn?volumeNo=5531590&memberNo=32155686&vType=VERTICAL>

[‘사상 최악’ 폭염·폭설...작년 이상기후 피해 컸다] mbc뉴스 2019.02.07

http://imnews.imbc.com/replay/2019/nw1700/article/5153386_24656.html

[사상 최고 폭염과 폭설 기록한 2018년, 왜일까?] 메디컬월드뉴스 2019.03.31.

<http://medicalworldnews.co.kr/news/view.php?idx=1510929990>

[무선 IoT 데이터 전송을 위한 사이언스로라 프로젝트 추진] KISTI 2018.09.27.

<https://blog.naver.com/withkisti/221366428112>

[범죄율 낮은데...한국인 '밤길 불안' 왜 클까] 서울신문 2018.08.06.

<https://www.seoul.co.kr/news/newsView.php?id=20180806009031#csidx7a361c8176b663ab8880e86d20dd1ba>

[[국감현장] "기상청이 아니라 '오보청'·'구라청'" 질타] 연합뉴스 2018.10.15

<https://www.yna.co.kr/view/AKR20181015065000004?input=1195m>