

1. 개요

☐ 연구 환경

- 과학중점학교인 과천중앙고에는 몇 개의 과학관련 동아리가 있음
- 지구환경을 생각하고 연구하는 기후 환경 동아리(IOS)
- 스마트폰 관련 앱 및 아두이노를 연구하고 개발할 수 있는 역량을 가진 컴퓨터 동아리(APP)
- 물리 관련 실험 및 융합과학 성격의 다양한 로봇과 아두이노 관련 작품을 제작하는 물리 동아리(PIRL)
- 통계수학으로 인간과 사회에 대한 연구를 하는 수학동아리(M4M)

☐ 연구 목적

- 미세먼지 센서와 아두이노를 이용하여 미세먼지를 측정하고 예보해, 학교와 같이 인원이 많아 일반 가정에 비해 여러 상황과 대처가 늦은 기관에서 보다 효율적으로 미세먼지에 대응 할 수 있는 미세먼지 예보 시스템을 개발하고자 함.

2. 연구주제 선정

☐ 문제의 착안점

- 황사와 더불어 중국에서 불어오는 미세먼지로 인한 피해가 심각함
- 학교와 같이 많은 인원이 생활하며, 체육 및 동아리 활동과 같은 여러 가지 야외활동을 하는 기관 들은 일반 가정에비해서 상대적으로 미세먼지에 대한 대처가 부족함

(뉴스에서 미세먼지 주의보가 발령이 된 후 학교에서 대처가 부족한 사례)

- 교실에 있으면 미세먼지를 피할 수 있는지 확인 필요
- 실내에서 환기여부, 마스크 착용여부가 인체에 주는 영향 확인 필요
- 관악산에 인접한 과천중앙고등학교의 지리적인 여건이 미세먼지에 영향을 주는지 확인 필요

- 학교에서 미세먼지에 대처해서 안전하게 친구들을 보호할 수 있는 예보 시스템 개발을 위해 기후환경+물리+컴퓨터+수학 동아리가 모임

분야	시스템 개발을 위한 연구 분야
기후환경	미세먼지의 농도에 영향을 주는 독립 변인의 종류와 특성
물리	미세먼지 센서, 온도센서 등 기후관련 센서의 작동원리
수학	미세먼지 예보 시스템의 통계적 자료 분석
컴퓨터	아두이노를 이용하여 센서 정보 수집 및 자료 공유 방안

□ 연구주제의 선정 과정

- 미세먼지 사전 실험 연구
 - 미세먼지가 심할 때 학교에서 어떻게 행동해야 하는지에 대한 적절한 안내가 필요
 - 학교에서 미세먼지에 영향을 주는 독립 변인들 설정 후 실험 연구 필요
 - 예상되는 독립 변인들
 - ✓ 장소에 따른 차이가 있는가?
 - ✓ 시간에 따른 차이가 있는가?
 - ✓ 보조 장치 및 행동이 효과가 있는가?
 - 사전 실험 연구의 주제와 기대되는 결론

연구 주제	미세먼지가 심할 때 학교에서 구체적으로 어떻게 행동해야 하는가?
기대 결론	미세먼지가 심할 때 어느 시간대에 어느 장소에서 어떤 행동을 하는 것이 도움이 된다.

3. 연구 내용 및 방법

□ 연구방법

1. 사전 실험

○ 시간대별 미세먼지 측정

- 미세먼지 측정 센서와 MBL을 이용해 시간대별 미세먼지 농도를 측정 한 뒤 그래프나, 엑셀 데이터로 나타냄
- 학교 내부의 전체적인 측정치를 종합해 평균 값 으로 표시
ex. AM 9:00 학교 내부 미세먼지 농도 평균 x.yy ppm
- 학교 외부(운동장 및 학교 부지)의 전체적인 측정치를 종합해 평균 값 으로 표시
ex. AM 9:00 학교 외부 미세먼지 농도 평균 z.xx ppm
- 과천시의 시간대별 미세먼지 농도와 비교 후 시간대별 특이성 탐구

○ 장소별 미세먼지 측정

- 미세먼지 측정 센서와 MBL을 이용해 장소별 미세먼지 농도를 측정 한 뒤 그래프로 나타냄
- 학교 내부의 공간들 (교실, 다목적실, 급식실, 교장실, 교무실 등)의 미세먼지 농도를 측정 한 뒤 측정치를 그래프로 나타냄
- 과천 시내와 비교해 비교적 산지에 위치한 학교와, 과천 시내의 미세먼지 농도를 측정 한 뒤 비교해, 미세먼지의 지형적인 특이성을 알아봄

2. 경보 시스템 구축

○ 경보 장치 개발

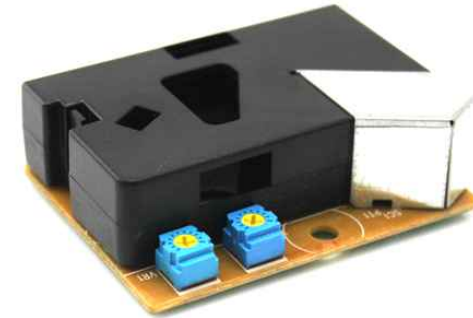
- 미세먼지 검출 센서와, 아두이노를 이용하여 미세먼지 경보 장치 개발
- 경보 장치의 서버와의 데이터 송수신을 위한 프로그램 작성



* 미세먼지 센서 PM1001

Measurement range	Diameter of dust: Above1um 8000 pcs / 283mL
Working voltage	DC5V±5%
Working current	90mA
Working temperature	-10~45℃
Storage temperature	-30~60℃
Working humidity	0- 95%RH non-condensing
Size	59*45*17.2(mm)
Weight	About 26g
Stable time	1 minute after power on
Output	Low level valid, Hi : 5V Lo : 0V

미세먼지 센서 PM1001 성능표



*미세먼지 센서 DSM501A



*미세먼지 센서 GP2Y10

3. 예보 시스템 구축

○ 예보 시스템 개발

- 실제 기상예보나, 기타 환경 관련 예보는 정교하게 짜여진 알고리즘에 따라 슈퍼컴퓨터의 연산을 거쳐 하게 됨
- 하지만 관측을 위한 고가의 장비, 다양한 지역의 기상정보, 예보 알고리즘에 대한 지식이 전무 함
- 따라서 처음에는 예보 시스템은 개발이 불가능 할 것이라 여김
- 하지만 인터넷으로 미세먼지 농도는 1시간 간격으로 기상청에서 제공되고, 학교가 산지에 위치해 인위적 환경보다 자연적인 환경의 영향을 더 받아 어느 정도 예보가 가능 할 것이라 여김
- 그리고 미세먼지는 주로 중국에서 발생하여 한반도로 바람을 타고 전파되는데, 이 바람인 편서풍의 속도를 이용하여 대략적인 예보를 할 수 있음
- 부가적인 하드웨어를 구축 할 필요 없이 예보 시스템을 개발할 수 있음

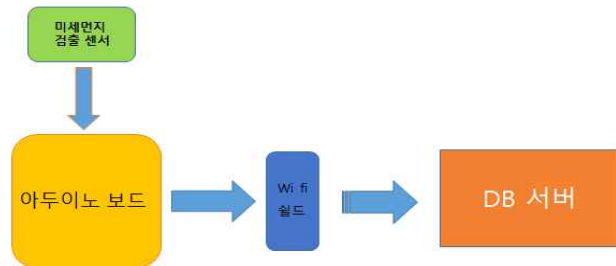
※ 문제 해결을 위한 실험, 기법, 전문가 자문 등에 대한 기술

□ 세부계획

1. 경보 시스템 구축

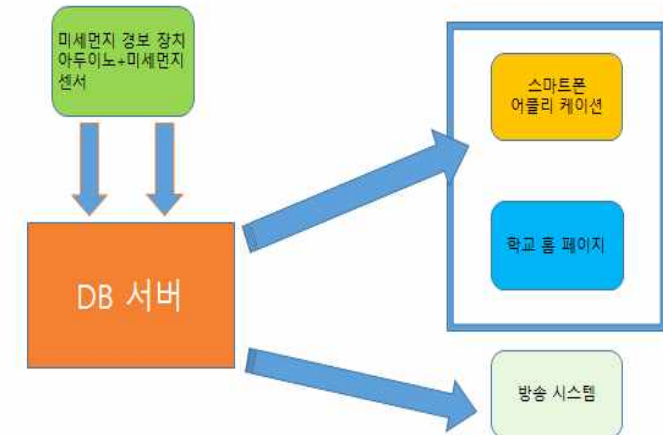
○ 경보 장치 개발

- 미세먼지 검출센서와 아두이노, Wifi 모듈을 연결하여 wifi를 통해 서버와 통신을 하며 미세먼지 측정값을 전송 할 수 있는 장치 개발
- 개발된 장치는 학교옥상에 배치



○ 경보 시스템 개발

- 데이터베이스 서버를 구축하여, 경보 장치에서 측정된 수치를 서버에 전송 할 수 있게 함
 - 구축된 데이터베이스 서버를 활용하여 학교 어플리케이션과 학교 홈 페이지에 측정값을 연동시킴
 - 학교 방송부와 연계하여 측정된 미세먼지 농도가 매우 높을때 방송을 통해 학교 전체에 경보를 발령함
 - 학교 홈페이지와 어플리케이션에 미세먼지 정보가 발령 되었을 때의 행동요령을 공지하여 피해를 최소화 시킬 수 있게함

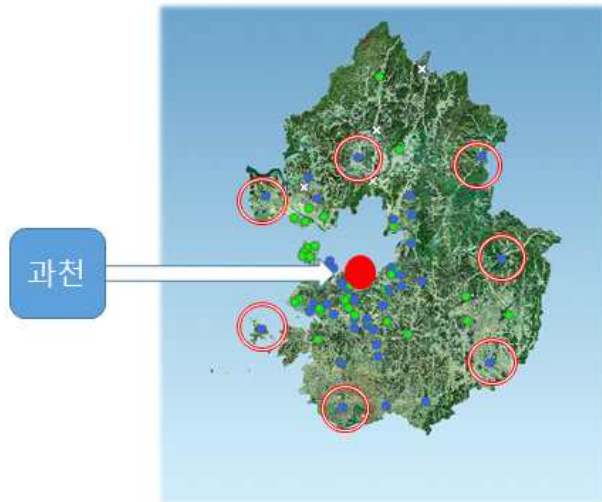


* 경보시스템 설계도

2. 예보 시스템 구축

○ 예보 시스템 DATA수집

- 과천은 경기도의 중심에 있어, 예보 시스템을 구축의 거점 지형으로 삼음
- 우리나라의 연 평균 풍속은 2.5m/s 인데, 이것을 3시간 단위로 바꾸면 27~30km 정도가 됨
- 예보에 필요한 정보들을 가진 거점 도시들은, 과천을 중심으로 반경 27~30km 에 위치한 도시들임

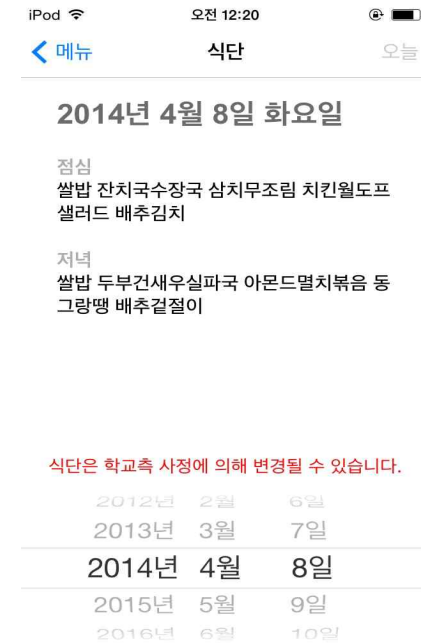


* 지정한 도시의 위치

- 기상청이나, airkorea.or.kr 사이트에서는 기상 정보를 1시간~3시간 주기로 제공함
- 예보에 필요한 지역별 풍속, 미세먼지 농도를 위의 사이트에서 데이터 파싱(Parsing)을 이용해 3시간 마다 구축한 데이터베이스 서버로 전송

- 파싱(Parsing) 이란?

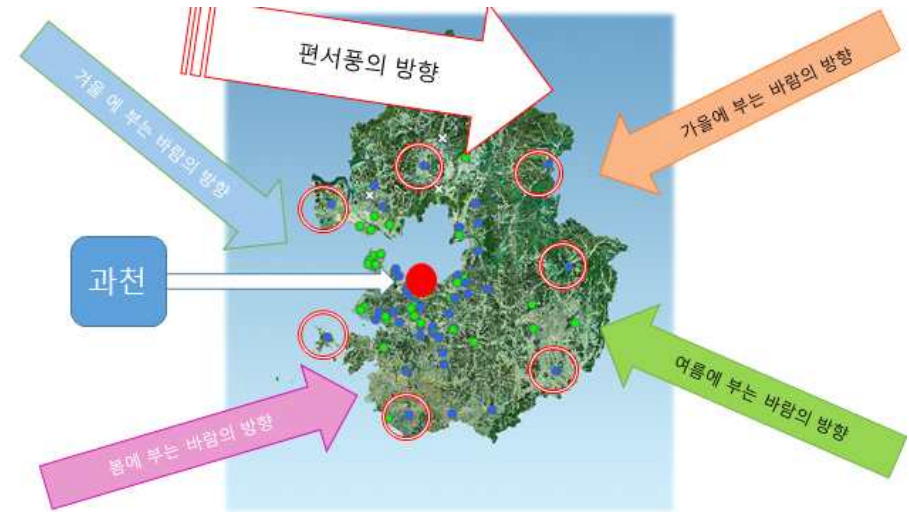
문장의 분류 루틴에 의하여 문장의 음절로부터 문장의 종류를 판별해 각 문자의 처리 프로그램의 제어를 넘기는 파싱 동작을 표현하는 알고리즘.



* 파싱의 예 (학교 급식 정보를 학교 데이터베이스에서 가져옴)

○ 예보 시스템 구축

- 고등학생의 능력과 제한된 자본으로 인해 직접 측정소를 도시마다 세워 직접 예보 시스템을 구축하는 것은 불가능한 일임
- 미세먼지의 특성을 조사 하던 중 미세먼지는 다른 기상현상과는 다르게 산발적으로 급격히 일어나지 않고, 주로 편서풍을 타고 중국으로부터 날아오는 것을 알게 됨
- 그래서 중국에서 불어오는 편서풍의 정보와, 그에 따른 미세먼지 검출 여부를 종합해 분 단위의 정확한 예보는 아니지만, 시간단위로 예보를 할 수 있을 거라 예상됨
- 마침 과천은 경기도의 중심에 위치해, 미세먼지가 바람을 타고 한국으로 이동하게 되면 경기도의 외곽에 위치한 도시들부터 미세먼지가 검출될 것 이고, 그 도시들을 거쳐 과천까지 도달할 것임
- 예보시스템 구축 단계에서 수집한 데이터들을 토대로, 거점 도시에서 미세먼지가 검출 된 후, 과천에서 미세먼지가 검출 되기까지 시간을 그 시점의 풍향을 반영해 측정함
- 측정된 시간을 통해 통계 프로그램인 SPSS를 이용해 평균적으로 미세먼지가 과천에서 측정되는데 까지 걸린 시간을 구함
- 약 2주 ~ 3주 동안 수집된 데이터와 위에서 구한 평균적인 시간을 토대로 미세먼지 예보 시스템을 구축함
- 미세먼지 예보시스템을 구축하는데 고려할 시스템 변수로는 계절에 따른 바람의 방향의 변동이 있다
- 이것을 해결하기위해 지정한 거점 도시에서 받아오는 데이터를 계절별로 변경함



* 대한민국의 계절별 바람의 방향

(계절이 봄이라면 과천의 남서쪽에 있는 도시의 데이터를, 가을이라면 북동쪽에 있는 도시의 데이터를 가지고 예보함)

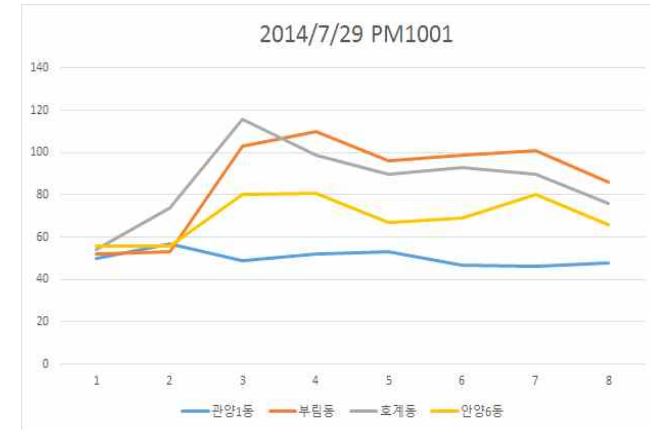
- 그리고 미세먼지와 더불어 여러 기상 상황을 알려주는 사이트인 <http://airkorea.or.kr>에서 제공되는 다른 지역의 대기 상황과 시간에 따른 미세먼지 농도 증감의 추이를 분석해 미세먼지 예보의 정확성을 향상 시킬 것임

○ 예보 시스템 테스트

- 구축된 미세먼지 시스템을 가지고 실제 예보가 가능한지 직접 테스트를 해봄
- 통계를 이용해 구한 풍향에 따른 미세먼지 도착 예정시간을 이용하여 실제 그 시간에 미세먼지가 학교에서 검출되는지 학교에 설치된 미세먼지 검출 센서를 이용하여 측정함
- 지형과, 온도와 같은 변수들을 고려하여 미세먼지 예보의 시간 오차는 ± 20 분으로 함
- 만약 지속적으로 오차범위를 크게 벗어나 예보라 할 수 없을 때 예는 예보시스템의 알고리즘의 재점검 및 거점 도시의수 (표본)를 늘리고, 환경적인 변수를 고려해 시스템을 재설계함
- 오차범위를 크게 벗어나지 않고 예보의 정확도가 70~80% 이상이 되면 학교 홈페이지와, 어플리케이션에 미세먼지 예보 시스템을 탑재함
- 그리고 학교 서버를 이용하여 학생들을 통솔하는 선생님들에게 학교 내부 메신저를 이용하여 1~2시간 간격으로 미세먼지 예보를 전송함

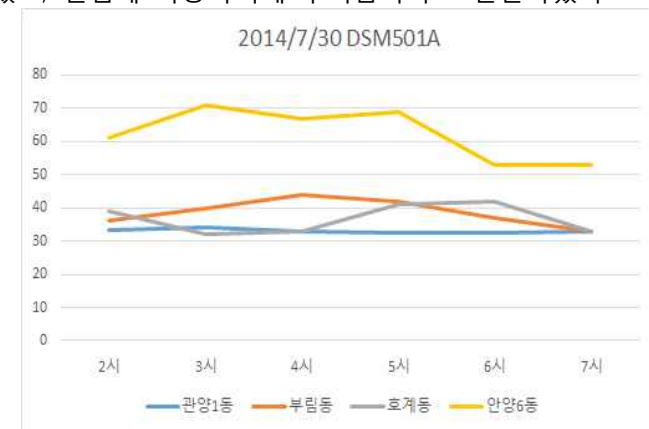
4. 연구 결과

□ 사전 실험을 통한 센서 선정

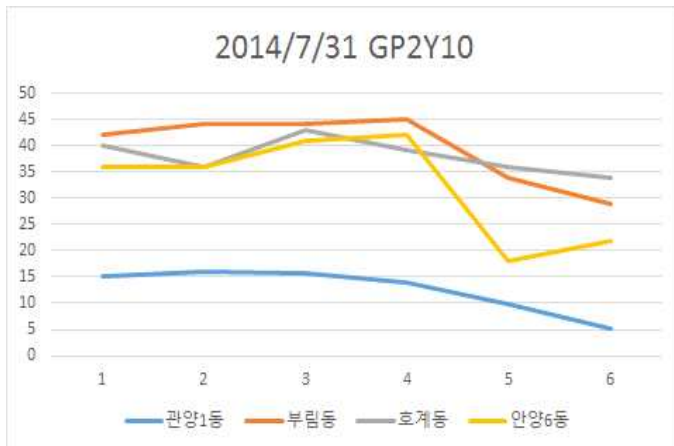


PM1001센서를 이용하여 실험을 진행하였다. 가로축은 시간, 세로축은 미세먼지 농도이며 관양1동이 센서로 측정한 값, 나머지 선들은 기상청에서 제공하는 미세먼지 농도이다.

하루간의 실험결과 PM1001센서는 미세먼지 농도의 변화를 제대로 반영하지 못하고 있고, 실험에 사용하기에 부적합하다고 판단하였다.



다음은 DSM501A 센서이다. Pm1001센서와 마찬가지로 제대로 반영을 하지 못하고 있으며, 역시 실험에 사용하기에 부적합하다고 판단하였다.



마지막으로 GP2Y10센서이다. 측정지점에서 가장 가까운 관측소인 부림동의 측정결과와 추세가 일치다. 그래서 가장 신뢰성 있는 결과라고 판단하였고 앞으로 실험에 사용할 센서로 선정하였다.

□ 미세먼지 측정센서 코드작성: 1분에 한번 씩 미세먼지 측정값을 LCD에 표시하도록 개발함. 다음은 작성된 프로그램의 일부임

```
#include "SoftwareSerial.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup(){
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
}
void loop(){
  digitalWrite(ledPower,LOW); // 센서 내부 LED 점등
  delayMicroseconds(samplingTime);
  voMeasured = analogRead(measurePin); // 먼지 센서 측정값 읽기
  delayMicroseconds(deltaTime);
  digitalWrite(ledPower,HIGH); // 센서 내부 LED 소등
  delayMicroseconds(sleepTime);
  calcVoltage = voMeasured * (3.3 / 1024); // 데이터 시트 기반 측정값
  dustDensity = 0.17 * calcVoltage - 0.1; // 변환공식
  lcd.setCursor(0, 1); // LCD커서를 첫줄로 설정
  lcd.print("Dust(ug/m^3)="); //LCD에 "~" 부분 표시
  lcd.print(dustDensity*100); //LCD에 미세먼지 농도 표시
}
```



□ RTC 사용 코드 작성: 분당 미세먼지 측정 농도를 기록하기 위해
RTC를 사용함. 다음은 작성된 프로그램의 일부임

*RTC(real time clock)의 약자로 시간정보를 알려주는 모듈이다

```
#include <string.h>
#include <Wire.h>
#define RTC_ADDRESS 0x68 // RTC 모듈의 I2C 주소
byte bin2bcd(int n){
    byte val = 0;
    int ten = n / 10; // 십자리
    int one = n % 10; // 일자리
    val = (ten << 4) | one; // BCD 형식으로 변환
    return val;
}
void timeSetting(int _year, int _month, int _day, int _hour, int _min, int _sec){
    Wire.beginTransmission(RTC_ADDRESS); // I2C 주소 설정 및 전송 시작
    Wire.write(0); // RTC 모듈의 레지스터 주소 설정
    Wire.write(bin2bcd(_sec)); // 초 설정
    Wire.write(bin2bcd(_min)); // 분 설정
    Wire.write(bin2bcd(_hour)); // 시 설정
    Wire.write(0); // 요일 설정
    Wire.write(bin2bcd(_day)); // 일 설정
    Wire.write(bin2bcd(_month)); // 월 설정
    Wire.write(bin2bcd(_year - 2000)); // 년 설정
    Wire.write(0); // 정현파 출력 제어 설정
    Wire.endTransmission(); // 전송 종료 }
byte bcd2bin(byte n){
    byte val = 0;
    int ten = n >> 4; // 십자리 (5비트에서 7비트까지 3비트)
    int one = n & 0x0F; // 일자리 (하위 4비트)
    val = ten * 10 + one; // 이진 형식으로 변환
    return val;
}
```

```
void setup(){
    Serial.begin(9600);
    Wire.begin(); // Wire 라이브러리 초기화
    timeSetting(2014, 8, 8, 23, 47, 0); // 시작시간 설정
}
void loop(){
    Wire.beginTransmission(RTC_ADDRESS);
    Wire.write(0); // RTC 모듈의 레지스터 주소 설정
    Wire.endTransmission();
    Wire.requestFrom(RTC_ADDRESS, 7); // 날짜 시간 정보 요구
    uint8_t _second = bcd2bin(Wire.read() & 0x7F); // 초
    uint8_t _minute = bcd2bin(Wire.read()); // 분
    uint8_t _hour = bcd2bin(Wire.read()); // 시
    Wire.read(); // 요일
    uint8_t _day = bcd2bin(Wire.read()); // 일
    uint8_t _month = bcd2bin(Wire.read()); // 월
    uint16_t _year = bcd2bin(Wire.read()) + 2000; // 년
    // 연월일 출력
    lcd.setCursor(0, 0);
    lcd.print(String(_year) + "/");
    lcd.print(String(_month) + "/");
    lcd.print(String(_day) + " ");
    // 시분초 출력
    lcd.print(String(_hour) + ":");
    lcd.print(String(_minute));
    lcd.print(String(_second));
    *RTC 소스는 오픈소스를 이용했음
```

□ MicroSD 카드 저장 코드 작성: 미세먼지 센서에서 받은 값과, RTC의 시간 정보를 이용하여 1분에 한번씩 SD카드에 저장하는 코드이다.
다음은 작성된 프로그램의 일부임

```
#include "SoftwareSerial.h"
#include <Wire.h>
#include <SD.h>
const int chipSelect = 4;
void setup(){
  Serial.begin(9600);
  Serial.print("Initializing SD card...");    // SD카드 인식 부분
  pinMode(10, OUTPUT);                       // SD카드를 인식하지 못하면
  if (!SD.begin(chipSelect)) {               // 시리얼 출력을 통해 알려줌
    Serial.println("Card failed, or not present");
    return;
  }
  Serial.println("card initialized.");
}
String dataString = "";                      //SD카드에 저장할 데이터지정
int sensor = dustDensity*100;               // 저장할 데이터 = 측정값

dataString += String(sensor);                /* 저장할 데이터의 양식
dataString += " ug/m^3";                    측정값, 연,월,일,분,초 순으로 기록
dataString += "_";                          ex: 48ug/m^3 2014/8/23/17:02/01
dataString += String(_year);                */
dataString += "/";
dataString += String(_month);
dataString += "/";
dataString += String(_day);
dataString += " ";
dataString += String(_hour);
dataString += ":";
dataString += String(_minute);
dataString += ":";
dataString += String(_second);
delay(60000);                               // 1분을 주기로 저장
```

```
File dataFile = SD.open("datalog.txt", FILE_WRITE); // 파일 생성
if (dataFile) {                                  // datalog.txt라는 이름을 가짐
  dataFile.println(dataString);
  dataFile.close();
  Serial.println(dataString);
}
else {
  Serial.println("error opening datalog.txt");
}
```

□ 제작 방법

1 SPLduino 보드 준비 (Lcd 모듈이 호환되는 신형 Spl보드 사용)

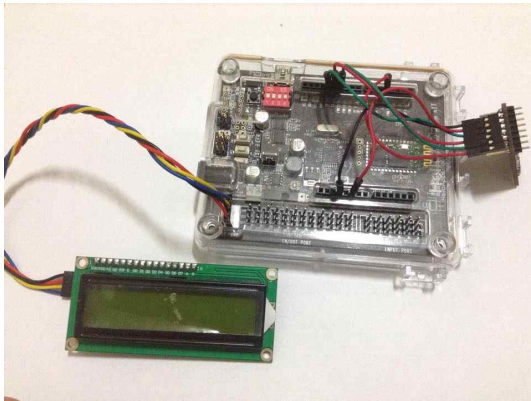


*신형 보드는 I2C포트의 배열이 SCL, SDA, GND, VCC 순임

2 좌측 하단의 I2C 포트에 LCD 모듈 연결

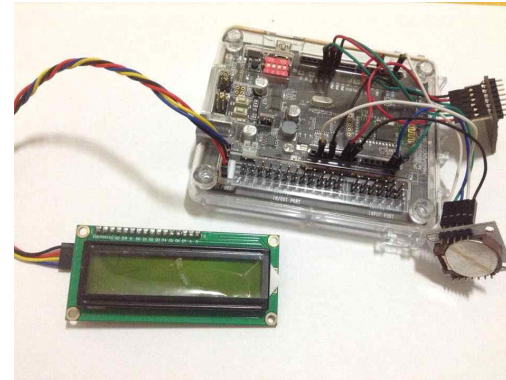


3 Micro SD 리더기 부착



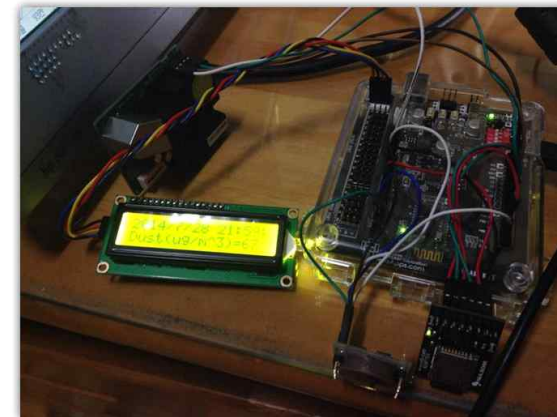
이때 리더기의
3V3단자는 보드의 3.3V에,
GND to GND
SCK to Digital 13 Pin
Do to Digital 12 Pin
DI to Digital 11 pin
CS to Digital 4 Pin
에 연결한다.

4 RTC 모듈 부착



이때 RTC 모듈의
VCC단자는 보드의 5V핀에,
GND to GND
SDA to Analog 4 Pin
SCL to Analog 5 Pin
DS to Digital 1 pin
에 연결한다.

5 미세먼지 센서 부착



이때 미세먼지 센서의
5V단자는 보드의 5V핀에,
GND to GND
TX to Analog 0 pin
RX to Analog 1 pin
P1 to Analog 13pin
에 연결한다.

*최종적으로 완성된 미세먼지 측정 장치

5. 연구 결과 및 향후 계획

□ 연구 결과

- 미세먼지 센서를 이용하여 미세먼지 농도를 측정 할 수 있게 됨
- 여러 센서들을 테스트하여 실제 사용할 적합한 센서를 찾아냄
- 같은 지역이라도 거리에 따른 편차가 꽤 큼

□ 향후계획

- 측정 장치를 모두 개발 완료 했으므로, 겨울방학 동안 연구를 진행시켜 우선 경보 시스템을 완성 할 계획
- 경보시스템이 성공적으로 작동하면 어플리케이션에 탑재하여 학생들의 접근성을 높일 것

□ 참고문헌

홍선학 외 저, (2012), " 아두이노 따라하기 "

채진욱 외 저, (2011), " 아두이노 인터랙티브 뮤직 "

윤성우 외 저, (2012), " 열혈 c 프로그래밍 "