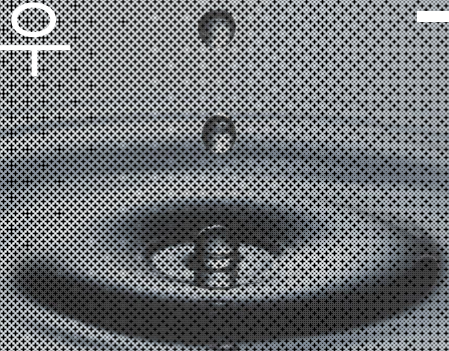


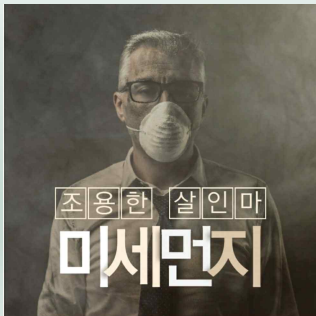
스마트 윈도우



1758027정민
1858015박장호

프로젝트 제안배경

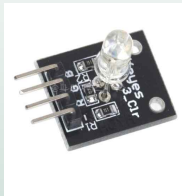
- 세계에서 공기오염이 심한 지역의 주민들은 최대 5.5년의 수명이 단축 된다고 함.
- 워드 코로나로 실내에 많은 인원이 밀집 되어있는 현재 환기가 수시로 필요로 함.



프로젝트 주요내용 및 기능

- 스마트 윈도우 공간 내의 사람들의 미세먼지로 인한 건강 악화 예방.
- 미세 먼지 농도를 알게 해주어 그날 그날의 복장 및 KF94 마스크 착용 유무 판단.
- 어플리케이션을 이용한 원격 제어로 집안 내 미세먼지 및 갑작스런 기상변화 사전 예방.

주요기술



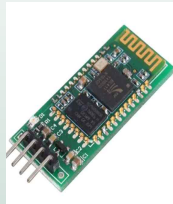
3색LED센서



미세먼지 감지 센서



Servo motors



hc06



LCD모듈

3색 LED센서로 미세먼지의 심함 정도를 색으로 식별.

미세먼지 감지 센서를 통해 미세먼지의 농도를 측정.

서보 모터로 미세먼지의 농도에 따라 창문을 여 닫기 가능.

와이파이 블루투스 모듈(hc06)을 통해 외출 시 어플리케이션을 이용한 원격제어 가능.

LCD판에 미세먼지 농도를 수치화 하여 대기 오염 정도 판별 가능.

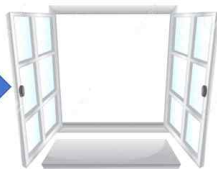
예상결과물



▲2019년 1월 19일, 송파구 선릉5동42호 주택에 설치된 센서 모습



pixtastock.com - 65095125



기대효과 및 활용방안

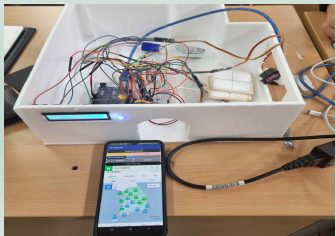
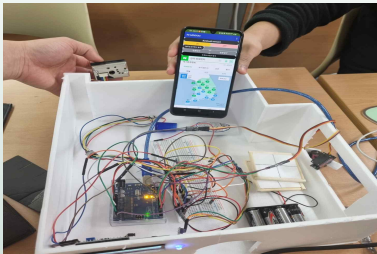
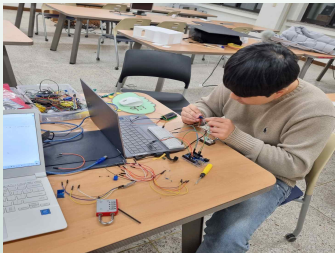
- 기대수명 감소 예방

- 미세먼지 유입 사전 예방으로 수명 감소를 방지.
- 미세먼지 농도 확인으로 인해 외출 시 KF94 마스크 착용으로 미세먼지 흡입 예방.

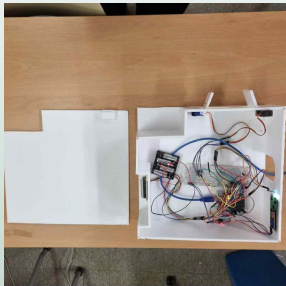
- 자동환기 시스템 및 우천 예방

- 외부에 있어도 미세먼지 집안 내 유입 사전 방지.
- 앱을 통한 실시간 미세먼지 농도 확인.
- 갑작스런 우천으로 인한 원격 제어 통제

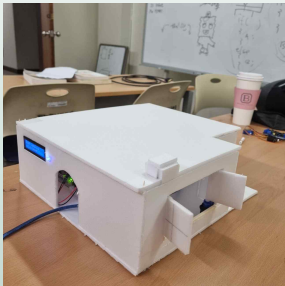
작품 만들기



작품 완성 사진



위에서 찍은 사진

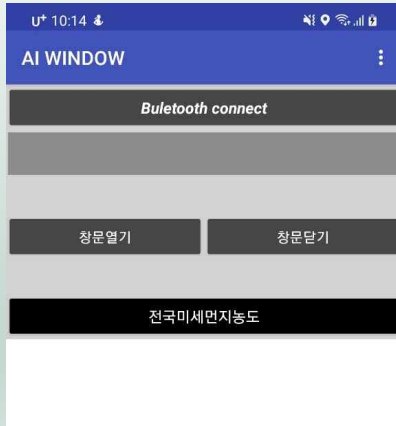
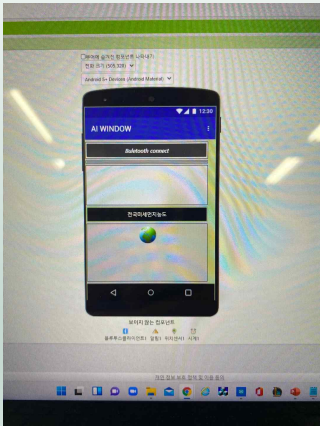


앞에서 찍은 사진

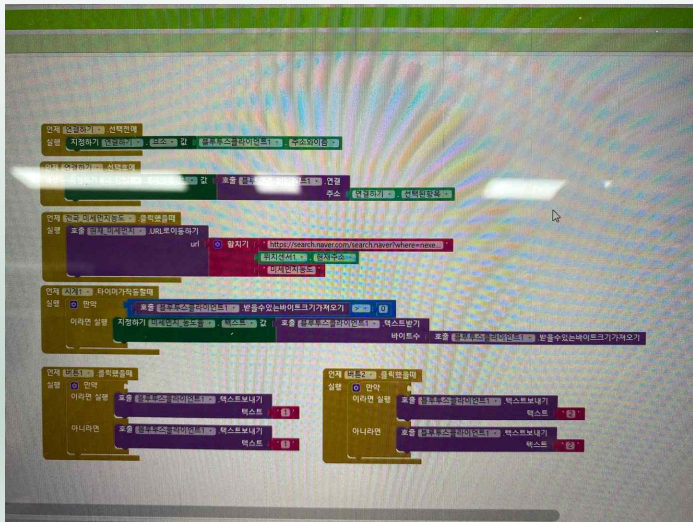


뒤에서 찍은 사진

앱 인벤터



앱 인벤터2



아두이노 코딩

```
#include <Servo.h>

#include <LiquidCrystal_I2C.h>

#include <SoftwareSerial.h>

#include <Wire.h>

SoftwareSerial mySerial(2, 3); //물류투스 모듈의 Rx -> 3번핀, Tx -> 2번핀 연결

Servo myservo; //서보

Servo myservol; //서보

int servoPin = 9; //서보핀

int servoPinl = 10; //서보핀

int dust_sensor = A0; // 미세먼지 핀 번호

int rgb_red = 5; // rgb 핀 빨간색 핀

int rgb_green = 6; // rgb핀 녹색 핀
```

```
int rgb_blue = 7; // rgb핀 파란색 핀
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C LCD 주소값 확인
```

```
float dust_value = 0; // 센서에서 입력 받은 미세먼지 값
```

```
float dustDensityug= int(0); // ug/m^3 값을 계산
```

```
int sensor_led = 12; // 미세먼지 센서 안에 있는 적외선 led 핀 번호
```

```
int sampling = 280; // 적외선 led를 키고, 센서 값을 읽어 들어 미세먼지를 측정하는 샘플링 시간
```

```
int waiting = 40;
```

```
float stop_time = 9680; // 센서를 구동하지 않는 시간
```

아두이노 코딩

```
void setup(){

  myservo.attach(servoPin);

  myservo.attach(servoPin1);

  Serial.begin(9600);

  mySerial.begin(9600);

  Serial.println("Servo Value(angle) = ?");

  myservo.write(90);

  |

  lcd.init(); // LCD

  Serial.begin(9600); // 시리얼 모니터 시작, 속도는 9600

  pinMode(sensor_led,OUTPUT); // 미세먼지 적외선 led를 출력으로 설정

  pinMode(5, OUTPUT); // 3색 LED 모듈 출력으로 설정, 붉은색

  pinMode(6, OUTPUT); // 녹색
```

```
pinMode(7, OUTPUT); // 파란색

lcd.backlight();

lcd.setCursor(0,0); //

lcd.print("Dust Density ");

}

void loop(){

  digitalWrite(sensor_led, LOW); // LED 켜기

  delayMicroseconds(sampling); // 샘플링해주는 시간.

  dust_value = analogRead(dust_sensor); // 센서 값 읽어오기

  delayMicroseconds(waiting); // 너무 많은 데이터 입력을 피해주기 위해 잠시 멈춰주는 시간.
```

아두이노 코딩

```
digitalWrite(sensor_led, HIGH); // LED 끄기

delayMicroseconds(stop_time); // LED 끄고 대기

dustDensityug = (0.17 * (dust_value * (5.0 / 1024)) - 0.1) * 1000; // 미세먼지 값 계산

Serial.print("Dust Density [ug/m3]: "); // 시리얼 모니터에 미세먼지 값 출력

Serial.print(int(dustDensityug));

lcd.setCursor(0,1); //lcd 위치

lcd.print(dustDensityug);

if(dustDensityug <= 30.0){

    analogWrite(rgb_red, 0);

    analogWrite(rgb_green, 0);

    analogWrite(rgb_blue, 255); // 대기 중 미세먼지가 좋을 일 때 파란색 출력
```

```
    analogWrite(rgb_blue, 255); // 대기 중 미세먼지가 좋을 일 때 파란색 출력

    Serial.print(" ");

    Serial.println(" blue");

    lcd.setCursor(5,1);

    lcd.print(" good ");

    mySerial.println("미세먼지농도 ");

    mySerial.println(dustDensityug);

    myservo.write(120); //서보각도값

    delay(1000);

} else if(30.0 < dustDensityug && dustDensityug <= 80.0){

    analogWrite(rgb_red, 0);
```

아두이노 코딩

```
analogWrite(rgb_green, 255); // 대기 중 미세먼지가 보통 일 때 녹색 출력

analogWrite(rgb_blue, 0);

Serial.print("  ");

Serial.println("green");

lcd.setCursor(5,1);

lcd.print(" SOSO          ");

mySerial.println("미세먼지농도 ");

mySerial.println(dustDensityug);

myservo.write(80); //서보각도값

delay(1000);

}

else if (80.0 < dustDensityug && dustDensityug <= 150.0){
```

```
analogWrite(rgb_red, 255); // 대기 중 미세먼지가 나쁨 일 때 노란색 출력

analogWrite(rgb_green, 155); // 대기 중 미세먼지가 나쁨 일 때 노란색 출력

analogWrite(rgb_blue, 0);

Serial.print("  ");

Serial.println("yellow");

lcd.setCursor(5,1);

lcd.print(" Bad          ");

mySerial.println("미세먼지농도 ");

mySerial.println(dustDensityug);

myservo.write(40); //서보각도값

delay(1000);

}

else if(150.0 < dustDensityug){
```

아두이노 코딩

```
analogWrite(rgb_red, 255); // 대기 중 미세먼지가 매우 높을 때 빨간색 출력
analogWrite(rgb_green, 0);
analogWrite(rgb_blue, 0);

Serial.print(" ");
Serial.println("red");

lcd.setCursor(5,1); //lcd위치
lcd.print(" very Bad ");

mySerial.println("미세먼지농도 ");
mySerial.println(dustDensityug);

myservo.write(5); //서보각도값

delay(1000);

}

delay(1000);
```

```
delay(1000);

{

  if(mySerial.available()){ //작동 확인 위한 시리얼통신

    int servoValue = mySerial.parseInt();

    myservo.write(servoValue);

    Serial.println(servoValue);

    delay(300);

  }

}

}
```

작품 영상



Q&A

질의 응답