

TEAM 오아시스

+ 김강민 김시우 최정완 방우영

기존 계획



블루투스를 이용하여
핸드폰 앱과 연동



앱을 통해 위치 입력하기



초음파 센서를 이용하여
움직임 감지



OpenWeathermap api를
이용해 강수 정보받기

변경 계획



와이파이 모듈로
인터넷과 연결



GPS모듈로 현재 위치 확인



PIR 센서를 이용하여
움직임 감지



한국 기상청 API 사용

```

1 import requests
2 import json
3 import serial
4 import time
5 from datetime import datetime
6
7 # 기상청 API 호출
8 def get_weather_data():
9     url = "http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtFcst"
10    now = datetime.now()
11    base_date = now.strftime("%Y%m%d")
12    base_time = now.strftime("%H%M")
13
14    params = {
15        "serviceKey": "vZ+F7/8M6mlpZdq0kmMhIKS10ubI8Fn00jArmMbiJRYLCwjMIqxWZRdeIKAvRFPEi0uuKWEL4C70l14fyT5fow==",
16        "numOfRows": "24",
17        "pageNo": "1",
18        "dataType": "JSON",
19        "base_date": base_date,
20        "base_time": base_time,
21        "nx": "55",
22        "ny": "127"
23    }
24    response = requests.get(url, params=params)
25    data = response.json()
26    return data
27
28 # 강수량 데이터 추출
29 def extract_precipitation_data(data):
30    precipitation_data = []
31    for item in data['response']['body']['items']['item']:
32        if item['category'] == 'RN1':
33            precipitation_data.append(float(item['fcstValue']))
34    return precipitation_data
35
36 # 아두이노로 데이터 전송
37 def send_data_to_arduino(data):
38    arduino = serial.Serial('/dev/ttyUSB0', 9600)
39    time.sleep(2) # 시리얼 통신 안정화 대기
40    for value in data:
41        if value >= 70:
42            arduino.write(b'B') # 파란색 LED
43        elif value >= 30:
44            arduino.write(b'Y') # 노란색 LED
45        else:
46            arduino.write(b'N') # LED 꺼짐
47        time.sleep(1)
48    arduino.close()
49
50 # 메인 함수
51 def main():
52    weather_data = get_weather_data()
53    precipitation_data = extract_precipitation_data(weather_data)
54    send_data_to_arduino(precipitation_data)
55
56 if __name__ == "__main__":
57     main()
58
59

```

진행사항

코드 작성 - 데이터 가져오기

===== RESTART: C:/Users

```

0시: 강수정정퍼오에
1시: 강수정정퍼오에
2시: 강수정정퍼오에
3시: 강수정정퍼오에
4시: 강수정정퍼오에
5시: 강수정정퍼오에
6시: 강수정정퍼오에
7시: 강수정정퍼오에
8시: 강수정정퍼오에
9시: 강수정정퍼오에
10시: 강수정정퍼오에
11시: 강수정정퍼오에
12시: 강수정정퍼오에
13시: 강수정정퍼오에
14시: 강수정정퍼오에
15시: 강수정정퍼오에
16시: 강수정정퍼오에
17시: 강수정정퍼오에
18시: 강수정정퍼오에
19시: 강수정정퍼오에
20시: 강수정정퍼오에
21시: 강수정정퍼오에
22시: 강수정정퍼오에
23시: 강수정정퍼오에

```

강수정보 없음 뜨는 시간대

PTY(강수량) 정보가 홈페이지에서 안 뜨고

낙뢰나 다른 정보가 떠서 데이터 추출이 안됨

개선 필요



```
1 #include <Adafruit_NeoPixel.h>
2
3 #define PIN          6
4 #define NUMPIXELS    24
5 #define DELAYVAL     100
6
7 // 시간별 강수 확률(1~12시)
8 int forc[12] = {0, 30, 60, 30, 60, 30, 0, 60, 30, 60, 30, 0};
9
10 // 네오픽셀 객체 생성
11 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
12
13 void setup() {
14     pixels.begin();
15     pixels.setBrightness(255);
16     pixels.show();
17 }
18
19 // 색상 정의
20 uint32_t colors[3];
21
22 void loop() {
23     // 색상 초기화
24     initializeColors();
25
26     // forc 배열을 순회하며 LED 색상 설정
27     for (int i = 0; i < 12; i++) {
28         setLEDColour(i);
29         delay(DELAYVAL);
30         pixels.show();
31     }
32 }
33
34 // 색상 초기화 함수
35 void initializeColors() {
36     colors[0] = pixels.Color(255, 255, 0); // yellow
37     colors[1] = pixels.Color(0, 255, 0); // green
38     colors[2] = pixels.Color(0, 0, 255); // blue
39 }
40
41 // LED 색상 설정 함수
42 void setLEDColour(int index) {
43     uint32_t color;
44     if (forc[index] < 30) {
45         color = colors[0];
46     } else if (forc[index] < 60) {
47         color = colors[1];
48     } else {
49         color = colors[2];
50     }
51     pixels.setPixelColor(((index * 2) + 18) % 24, color);
52     pixels.setPixelColor(((index * 2) + 18) % 24 + 1, color);
53 }
54
```

진행사항

코드 작성 - 강수 확률을 기반으로 LED로 표시

```

1  #include <Adafruit_NeoPixel.h>
2
3  #define PIN          6
4  #define NUMPIXELS    24
5  #define DELAYVAL     100
6
7  // 시간별 강수 확률(1~12시)
8  int forc[12] = {0, 30, 60, 30, 60, 30, 0, 60, 30, 60, 30, 0};
9
10 // 네오픽셀 객체 생성
11 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
12
13 void setup() {
14     pixels.begin();
15     pixels.setBrightness(255);
16     pixels.show();
17 }
18
19 // 색상 정의
20 uint32_t colors[3];
21
22 void loop() {
23     // 색상 초기화
24     initializeColors();
25
26     // forc 배열을 순회하며 LED 색상 설정
27     for (int i = 0; i < 12; i++) {
28         setLEDColour(i);
29         delay(DELAYVAL);
30         pixels.show();
31     }
32 }
33
34 // 색상 초기화 함수
35 void initializeColors() {
36     colors[0] = pixels.Color(255, 255, 0); // yellow
37     colors[1] = pixels.Color(0, 255, 0);  // green
38     colors[2] = pixels.Color(0, 0, 255);   // blue
39 }
40
41 // LED 색상 설정 함수
42 void setLEDColour(int index) {
43     uint32_t color;
44     if (forc[index] < 30) {
45         color = colors[0];
46     } else if (forc[index] < 60) {
47         color = colors[1];
48     } else {
49         color = colors[2];
50     }
51     pixels.setPixelColor(((index * 2) + 18) % 24, color);
52     pixels.setPixelColor(((index * 2) + 18) % 24 + 1, color);
53 }
54

```

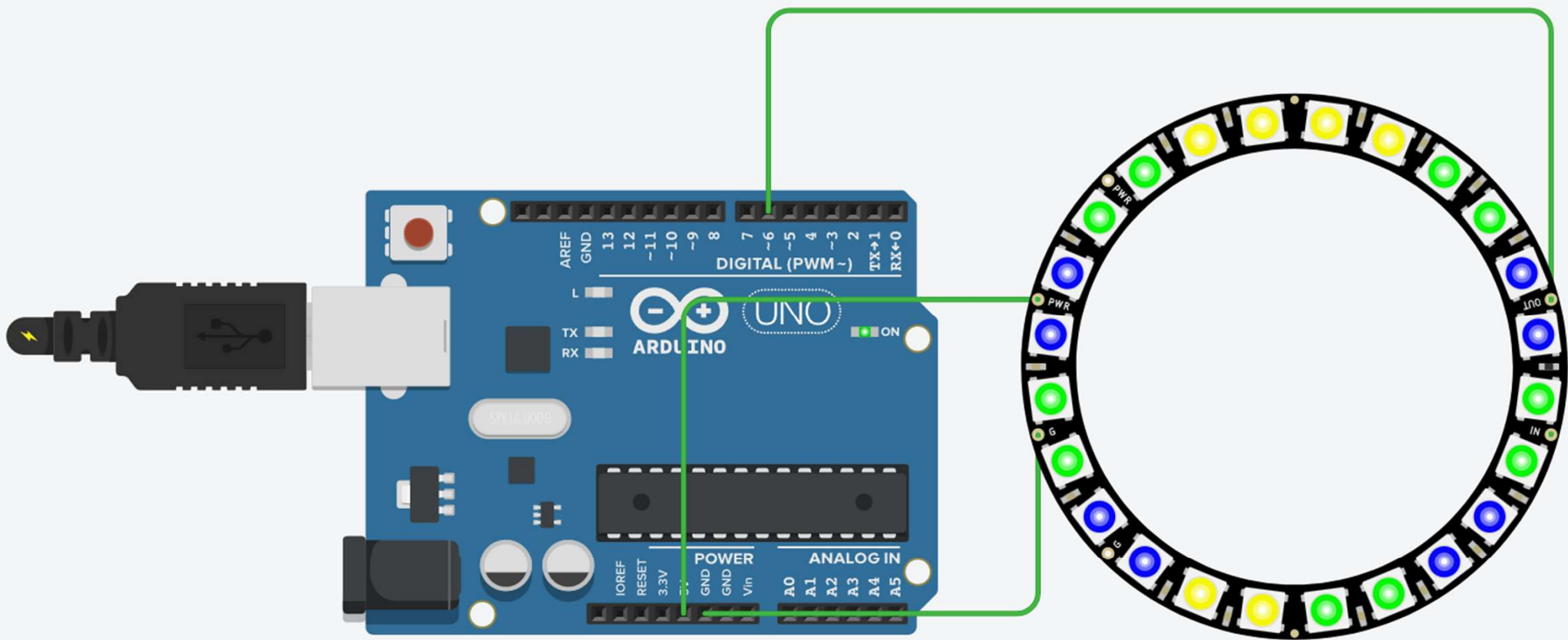
진행사항

코드 작성 - 강수 확률을 기반으로 LED로 표시

30% 미만 - 노랑

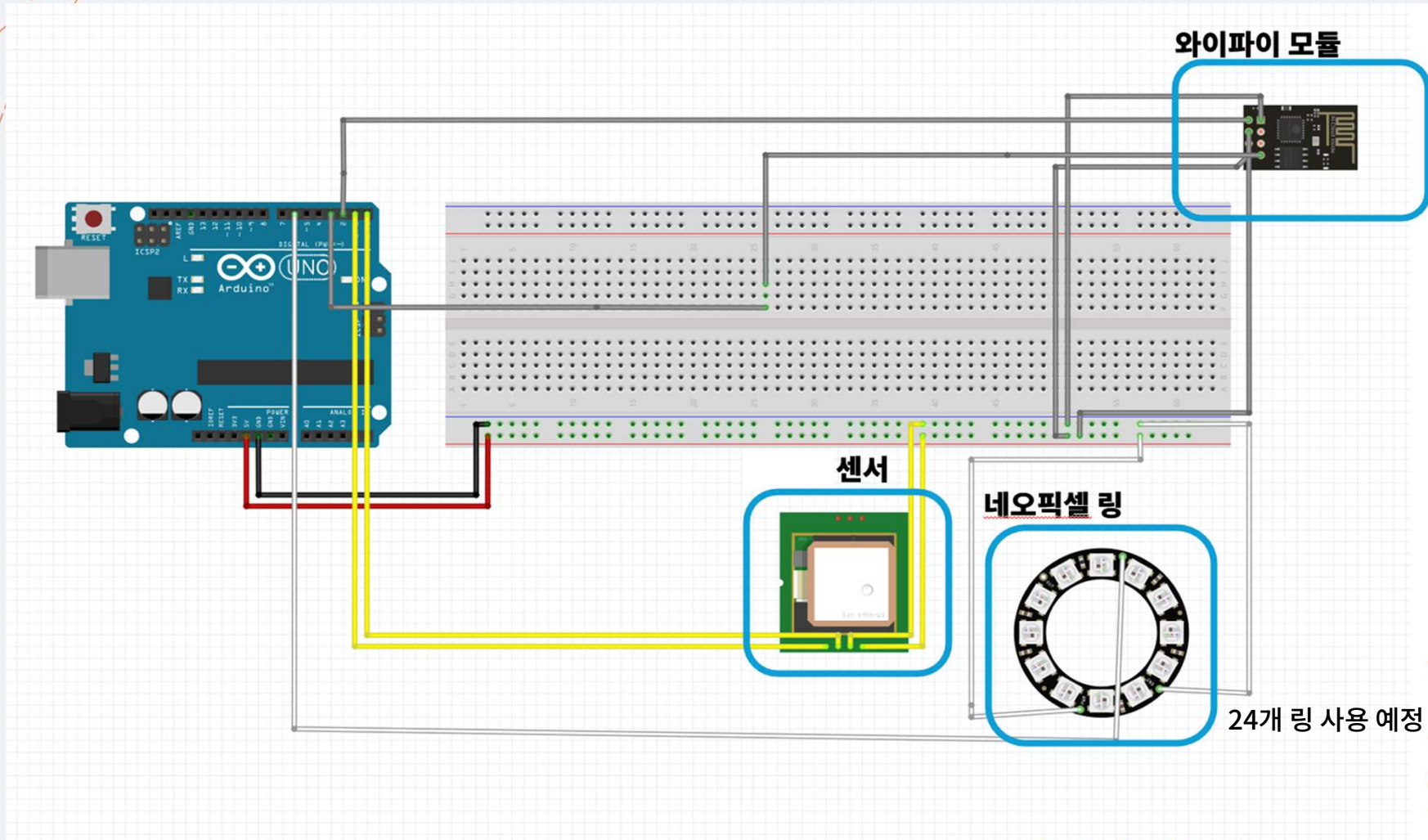
60% 미만 - 초록

그 외 - 파랑



진행사항

회로 구성



추후 계획

부품 구매 후 제작

강수 확률별 LED 색상 더 다양하게 변경

기상 데이터 코드 수정 필요

- 현재 방식은 외부 컴퓨터로 부터 시리얼 통신을 통해 기상 데이터를 가져오기 때문에, 와이파이 모듈을 통해 아두이노 보드 자체적으로 데이터를 가져오도록 한다

```

1 #include <ESP8266WiFi.h>
2 #include <ESP8266HTTPClient.h>
3 #include <ArduinoJson.h>
4
5 const char* ssid = "YOUR_SSID"; // Wi-Fi SSID
6 const char* password = "YOUR_PASSWORD"; // Wi-Fi 비밀번호
7
8 const char* apiKey = "YOUR_API_KEY"; // 기상청 API 키
9 const char* host = "apis.data.go.kr";
10 const int blueLED = D1; // 파란색 LED 핀 번호
11 const int yellowLED = D2; // 노란색 LED 핀 번호
12 const int redLED = D3; // 빨간색 LED 핀 번호
13
14 void setup() {
15   Serial.begin(115200);
16   WiFi.begin(ssid, password);
17
18   pinMode(blueLED, OUTPUT);
19   pinMode(yellowLED, OUTPUT);
20   pinMode(redLED, OUTPUT);
21
22   while (WiFi.status() != WL_CONNECTED) {
23     delay(1000);
24     Serial.println("Connecting to WiFi...");
25   }
26
27   Serial.println("Connected to WiFi");
28 }
29
30 void loop() {
31   if (WiFi.status() == WL_CONNECTED) {
32     HTTPClient http;
33     String url = String("http://") + host + "/1360000/VilageFcstInfoService_2.0/getUltraSrtFcst"
34       + "?serviceKey=" + apiKey
35       + "&numOfRows=24&pageNo=1&dataType=JSON"
36       + "&base_date=" + getDate() + "&base_time=" + getTime()
37       + "&nx=55&ny=127";
38     http.begin(url);
39     int httpCode = http.GET();
40
41     if (httpCode > 0) {
42       String payload = http.getString();
43       Serial.println(payload);
44       handleWeatherData(payload);
45     }
46
47     http.end();
48   }
49
50   delay(60000); // 1분마다 데이터 갱신
51 }

```

```

1 String getDate() {
2   // 현재 날짜를 "YYYYMMDD" 형식으로 반환하는 함수
3   time_t now = time(nullptr);
4   struct tm* timeinfo = localtime(&now);
5   char buffer[9];
6   strftime(buffer, 9, "%Y%m%d", timeinfo);
7   return String(buffer);
8 }
9
10 String getTime() {
11   // 현재 시간을 "HHMM" 형식으로 반환하는 함수
12   time_t now = time(nullptr);
13   struct tm* timeinfo = localtime(&now);
14   char buffer[5];
15   strftime(buffer, 5, "%H%M", timeinfo);
16   return String(buffer);
17 }
18
19 void handleWeatherData(String data) {
20   // JSON 데이터를 파싱하여 강수량 데이터를 추출하고 LED를 제어
21   StaticJsonDocument<2000> doc;
22   deserializeJson(doc, data);
23
24   float maxPrecipitation = 0;
25   for (JsonObject item : doc["response"]["body"]["items"]["item"].as<JsonArray>()) {
26     if (String(item["category"].as<const char*>()) == "Rn1") {
27       float precipitation = item["fcstValue"].as<float>();
28       if (precipitation > maxPrecipitation) {
29         maxPrecipitation = precipitation;
30       }
31     }
32   }
33
34   if (maxPrecipitation >= 70) {
35     digitalWrite(blueLED, HIGH);
36     digitalWrite(yellowLED, LOW);
37     digitalWrite(redLED, LOW);
38   } else if (maxPrecipitation >= 30) {
39     digitalWrite(blueLED, LOW);
40     digitalWrite(yellowLED, HIGH);
41     digitalWrite(redLED, LOW);
42   } else {
43     digitalWrite(blueLED, LOW);
44     digitalWrite(yellowLED, LOW);
45     digitalWrite(redLED, LOW);
46   }
47 }
48

```

추가

GPT를 통해 파이썬 -> 아두이노로 코드 수정

```

1 #include <ESP8266WiFi.h>
2 #include <ESP8266HTTPClient.h>
3 #include <ArduinoJson.h>
4
5 const char* ssid = "YOUR_SSID"; // Wi-Fi SSID
6 const char* password = "YOUR_PASSWORD"; // Wi-Fi 비밀번호
7
8 const char* apiKey = "YOUR_API_KEY"; // 기상청 API 키
9 const char* host = "apis.data.go.kr";
10 const int blueLED = D1; // 파란색 LED 핀 번호
11 const int yellowLED = D2; // 노란색 LED 핀 번호
12 const int redLED = D3; // 빨간색 LED 핀 번호
13
14 void setup() {
15   Serial.begin(115200);
16   WiFi.begin(ssid, password);
17
18   pinMode(blueLED, OUTPUT);
19   pinMode(yellowLED, OUTPUT);
20   pinMode(redLED, OUTPUT);
21
22   while (WiFi.status() != WL_CONNECTED) {
23     delay(1000);
24     Serial.println("Connecting to WiFi...");
25   }
26
27   Serial.println("Connected to WiFi");
28 }
29
30 void loop() {
31   if (WiFi.status() == WL_CONNECTED) {
32     HTTPClient http;
33     String url = String("http://") + host + "/1360000/VilageFcstInfoService_2.0/getUltraSrtFcst"
34       + "?serviceKey=" + apiKey
35       + "&numOfRows=24&pageNo=1&dataType=JSON"
36       + "&base_date=" + getDate() + "&base_time=" + getTime()
37       + "&nx=55&ny=127";
38     http.begin(url);
39     int httpCode = http.GET();
40
41     if (httpCode > 0) {
42       String payload = http.getString();
43       Serial.println(payload);
44       handleWeatherData(payload);
45     }
46
47     http.end();
48   }
49
50   delay(60000); // 1분마다 데이터 갱신
51 }

```

```

1 String getDate() {
2   // 현재 날짜를 "YYYYMMDD" 형식으로 반환하는 함수
3   time_t now = time(nullptr);
4   struct tm* timeinfo = localtime(&now);
5   char buffer[9];
6   strftime(buffer, 9, "%Y%m%d", timeinfo);
7   return String(buffer);
8 }
9
10 String getTime() {
11   // 현재 시간을 "HHMM" 형식으로 반환하는 함수
12   time_t now = time(nullptr);
13   struct tm* timeinfo = localtime(&now);
14   char buffer[5];
15   strftime(buffer, 5, "%H%M", timeinfo);
16   return String(buffer);
17 }
18
19 void handleWeatherData(String data) {
20   // JSON 데이터를 파싱하여 강수량 데이터를 추출하고 LED를 제어
21   StaticJsonDocument<2000> doc;
22   deserializeJson(doc, data);
23
24   float maxPrecipitation = 0;
25   for (JsonObject item : doc["response"]["body"]["items"]["item"].as<JsonArray>()) {
26     if (String(item["category"].as<const char*>()) == "RM1") {
27       float precipitation = item["fcstValue"].as<float>();
28       if (precipitation > maxPrecipitation) {
29         maxPrecipitation = precipitation;
30       }
31     }
32   }
33
34   if (maxPrecipitation >= 70) {
35     digitalWrite(blueLED, HIGH);
36     digitalWrite(yellowLED, LOW);
37     digitalWrite(redLED, LOW);
38   } else if (maxPrecipitation >= 30) {
39     digitalWrite(blueLED, LOW);
40     digitalWrite(yellowLED, HIGH);
41     digitalWrite(redLED, LOW);
42   } else {
43     digitalWrite(blueLED, LOW);
44     digitalWrite(yellowLED, LOW);
45     digitalWrite(redLED, LOW);
46   }
47 }
48

```

추가

GPT를 통해 파이썬 -> 아두이노로 코드 수정

(테스트 안함) (가능한지 확인만 해봄)

The background features a light gray field with stylized, wavy, dashed lines in a light brown or tan color. These lines flow from the left side towards the right. In the top-left corner, there is a white circle. In the bottom-right corner, there is another white circle, partially obscured by a solid gold-colored line that curves upwards and to the right.

감사합니다

완성본을 기대해주세요! ⁺