

드론을 이용한 교내 Wi-Fi 음영 확인 및 해결 제안 시스템



분 과 : C
팀 명 : 니가쏘다찌
팀 원 : 201824427 김동혜
201824470 박동한
지도교수 : 유영환 교수님

목 차

1. 요구조건 및 제약 사항 분석	3
1.1 과제 목표	3
1.2 요구조건	4
1.3 제약사항 분석에 대한 수정사항	4
2. 설계 상세화 및 변경 내역	5
2.1 데이터 수집 장치	5
2.2 웹 서버	5
2.3 웹 application	5
3. 갱신된 과제 추진 계획	6
4. 구성원 별 진척도	6
5. 과제 수행 내용 및 중간 결과	7
5.1 드론 운용	7
5.2 서버 생성 및 데이터 수신	8
5.3 Wi-Fi 측정 및 데이터 전송	9

1. 요구조건 및 제약 사항 분석

1.1 과제 목표

드론과 임베디드 보드를 이용하여 공공 와이파이 서비스가 존재하는 지역의 Wi-Fi 신호 강도를 측정

● Wi-Fi 신호 강도 탐지

- Wi-Fi 모듈을 장착한 임베디드 보드와 드론을 이용하여 해당 지역의 신호 강도를 측정

● 신호 강도의 가시화를 통한 음영지역 확인

- 신호 강도에 따라 다른 색으로 표현하여 신호 강도 데이터에 따른 신호 세기 분포도인 Wi-Fi heatmap 을 생성
- 완성한 신호 세기 분포도, Wi-Fi heatmap 을 웹페이지에 출력하여 사용자가 음영 지역이 현재 어떤 곳에 나타나는지 확인

● 음영 지역 해결을 확인

- 사용자는 음영 지역 해결을 위한 AP 추가 위치를 세기 분포도를 통해 확인하고 이를 드론을 이용해 AP 를 추가하여 음영 지역을 해결

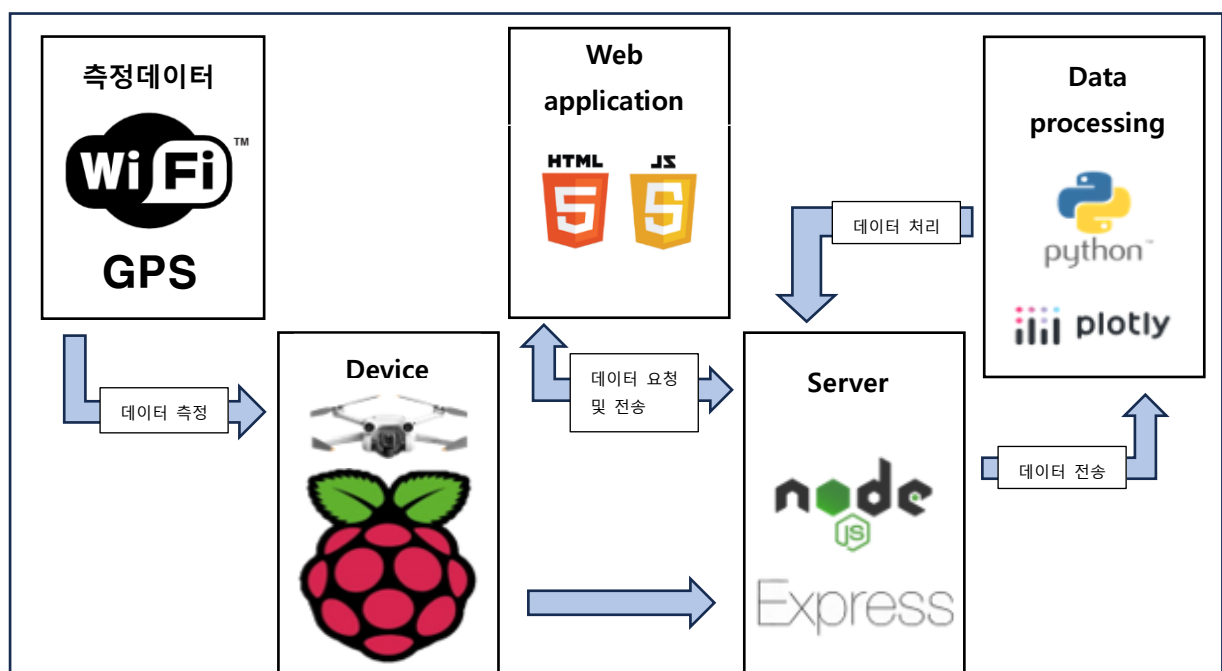


그림 1. 시스템 구성도

1.2 요구조건

Wi-Fi 신호 강도에 따른 세기 분포도를 생성하여 관리자가 음영 지역을 확인하고 드론을 이용한 AP추가를 통해 음영 지역을 제거하는 시스템을 개발

- Wi-Fi 속도와 위치 정보를 얻기 위한 라즈베리 파이용 데이터 수집 기능 구현
- 수집한 데이터를 전송하기 위한 서버 구현
- Wi-Fi 속도 데이터를 비교하여 측정 위치에 따라 다른 색으로 표현하기 위한 데이터 시각화 기능 구현
- 데이터 수집을 위한 드론 운용
- 최종 결과물을 사용자에게 보여주기 위한 웹페이지 구현

1.3 제약사항 분석에 대한 수정사항

1.3.1 기존 제약사항에 대한 수정사항

- AP의 재배치의 실현
 - Bluetooth 5 Beacon으로 Wi-Fi 신호를 대체하여 측정해 AP 재배치의 확인을 구현하는 것은 과제의 주제와 맞지 않는 방식이므로, Wi-Fi 신호를 측정한 후 음영 지역 발생 시 드론에 Wi-Fi 동글을 부착하여 AP를 추가하는 방식으로 변경

1.3.2 추가 제약사항 및 대책

- 드론에 부착하는 라즈베리 파이 보드가 너무 무거울 경우 드론이 높게 날지 못하는 것을 인지
 - 보드에 부착되는 부품(배터리, GPS 모듈 등)을 최대한 경량화 하여 드론이 제대로 날 수 있도록 할 예정
- 드론에 라즈베리 파이 보드를 부착할 방법을 고려
 - 라즈베리 파이 보드가 드론의 비행 시 방해되지 않도록 드론의 아랫면에 부착 예정
- 드론의 운용 가능 시간이 90분인 것을 확인
 - 운용 가능 시간을 고려해 측정 지역을 선택하며 측정 지역이 너무 넓을 경우 여러 번 나누어서 측정하거나 적절히 줄일 예정

2. 설계 상세화 및 변경 내역

2.1 데이터 수집 장치

- GPS와 배터리가 장착된 라즈베리 파이 보드 제작
- 측정한 GPS와 Wi-Fi 데이터를 JSON 형태로 저장
- 저장된 데이터를 웹 서버로 전송
- 라즈베리 파이 보드 경량화 예정
- 보드 경량화 후 드론과 물리적 연결 예정
- GPS와 배터리의 경우 학과 단체 구매 예산 편성 이후 구매 및 연결 예정

2.2 웹 서버

- Node.js로 구현
- 라즈베리 파이로부터 POST 방식으로 전달받은 데이터 저장 및 가공
- 웹 application에서 요청 시 데이터 전송

2.3 웹 application

- 웹 서버의 데이터를 plotly 라이브러리를 사용하여 Wi-Fi 속도에 따른 Wi-Fi heatmap 생성 및 출력
- Wi-Fi heatmap 내 각 지점은 GPS를 기반으로 하고 Wi-Fi 속도의 경우 색으로 출력

3. 갱신된 과제 추진 계획

5월			6월			7월					8월					9월					
3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주
착수보고서																					
	무선 신호 처리 및 임베디드 관련 기술 공부																				
							드론 운용														
							신호 탐지 기술 개발														
								서버 환경 구축													
								서버 개발													
								중간보고서 작성													
													부품 주문								
													GPS데이터 처리 기술 개발								
													신호 처리 및 가시화 기술 개발								
													웹페이지 개발								
															테스트 및 디버깅						
																		최종 발표/보고서 준비			

그림 2. 과제 추진 계획표

4. 구성원 별 진척도

이름	진척도
김동혜	<ul style="list-style-type: none">● 드론 운용 숙달● 서버 환경 구축 완료● 서버 개발 진행 중
박동한	<ul style="list-style-type: none">● 서버와 임베디드 보드 통신 완료● 신호 탐지 기술 개발 완료● 보드 경량화 구상 진행 중

5. 과제 수행 내용 및 중간 결과

5.1 드론 운용



그림 3. 드론 작동 전



그림 4. 드론 작동 후

- 드론 운용 숙달을 위한 시험 운행 수행
- 학교 내의 넉넉한 터에서 실제로 드론을 운용



그림 5. 드론에 보드 부착

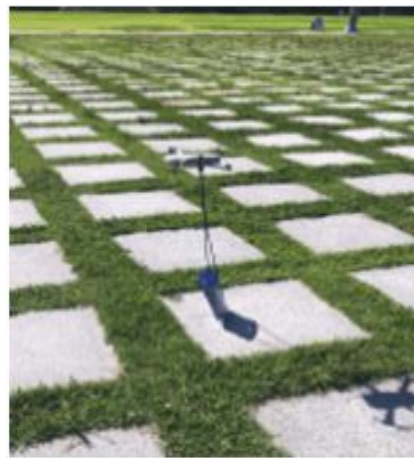


그림 6. 보드를 부착한 드론 운행

- 배터리를 연결한 보드를 드론에 매달아서 운용
- 드론에 보드를 부착할 방법을 아직 고려하지 못해 임시로 드론에 보드를 매달아서 확인
- 배터리가 너무 무거워 최대 출력임에도 드론이 잘 날지 못하는 것을 확인

5.2 서버 생성 및 데이터 수신

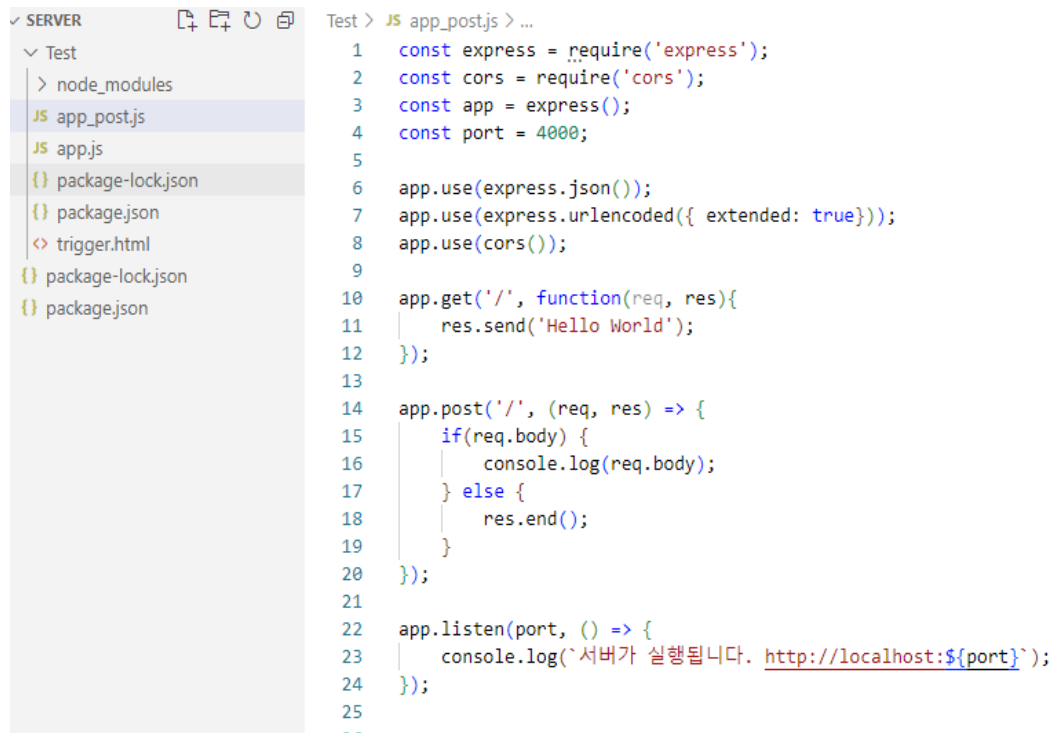


그림 7. 웹서버 구동 코드

- Node.js 의 express.js 프레임워크를 이용하여 서버를 생성
- node app_post.js 를 터미널에 입력 시 서버 구동 확인
- POST 방식을 이용해 라즈베리파이 보드로부터 수집한 JSON 데이터를 수신

```
PS C:\Server\test> node app_post.js
서버가 실행됩니다. http://localhost:4000
{ upload_speed: '8.05', download_speed: '25.705' }
```

그림 8. 전송된 데이터 확인

- 서버로부터 전송된 JSON 데이터를 콘솔에서 확인 가능

5.3 Wi-Fi 측정 및 데이터 전송

```
import time
import sys
import inspect
import json
import math

sys.path.append('/home/pi/.local/lib/python3.9/site-packages')

import speedtest

print(inspect.getfile(speedtest))
f = open('./data.txt', 'w')

wifi = speedtest.Speedtest(secure=True)
wifi.get_best_server()

total_download = 0
total_upload = 0

loop_num = 2

for i in range(loop_num):
    print("start")
    start = time.time()

    download_speed = round(wifi.download()/1000000, 2)
    f.write(str(download_speed)+'Mbps\n')

    upload_speed = round(wifi.upload()/1000000, 2)
    f.write(str(upload_speed)+'Mbps\n')

    end = time.time()

    total_download += download_speed
    total_upload += upload_speed

data = {
    'upload_speed': total_upload/loop_num,
    'download_speed': total_download/loop_num
}

file_path = './speed_test_result.json'
with open(file_path, 'w') as file:
    json.dump(data, file)

f.close()
```

그림 9. Wi-Fi 속도 측정 및 저장 코드

- speedtest 라이브러리를 활용하여 Wi-Fi 속도 측정
- 다운로드 속도와 업로드 속도를 측정하여 JSON 형태로 저장

```
{
  "upload_speed": 79.155,
  "download_speed": 57.8
}
```

그림 10. 저장된 데이터 형태

```
import requests
import json

file_path2 = './speed_test_result.json'

server_url = 'http://172.21.239.103:4000/'

with open(file_path2, 'r') as file:
    json_data = json.load(file)
    print(json_data)

try:
    response = requests.post(server_url, json_data)
    response.raise_for_status()
    print('data is successfully uploaded')
except requests.exceptions.RequestException as e:
    print('error occured in upload', str(e))
```

그림 11. 데이터 전송 코드

- 구동중인 서버로 데이터 전송
- 지정된 파일 경로에 있는 JSON 데이터를 읽어서 서버로 전송