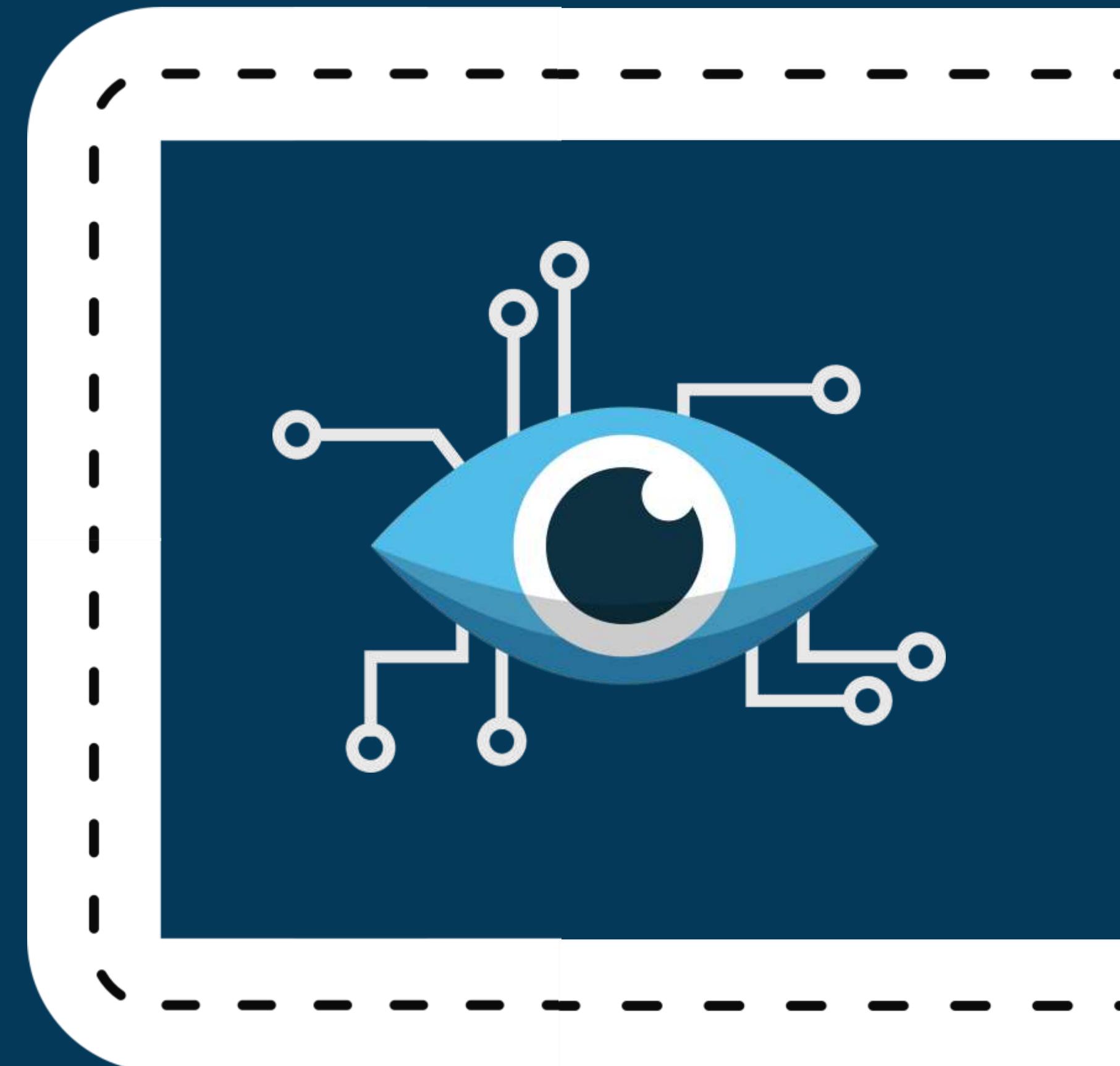


Real-Time Air Quality Monitoring System

Tech Stack: C# · JSON · TCP/IP Socket · Multi-threading

신선호



목차



1

프로젝트
개요

2

Flowchart

3

Protocol

4

UI 및
주요코드

5

실행결과

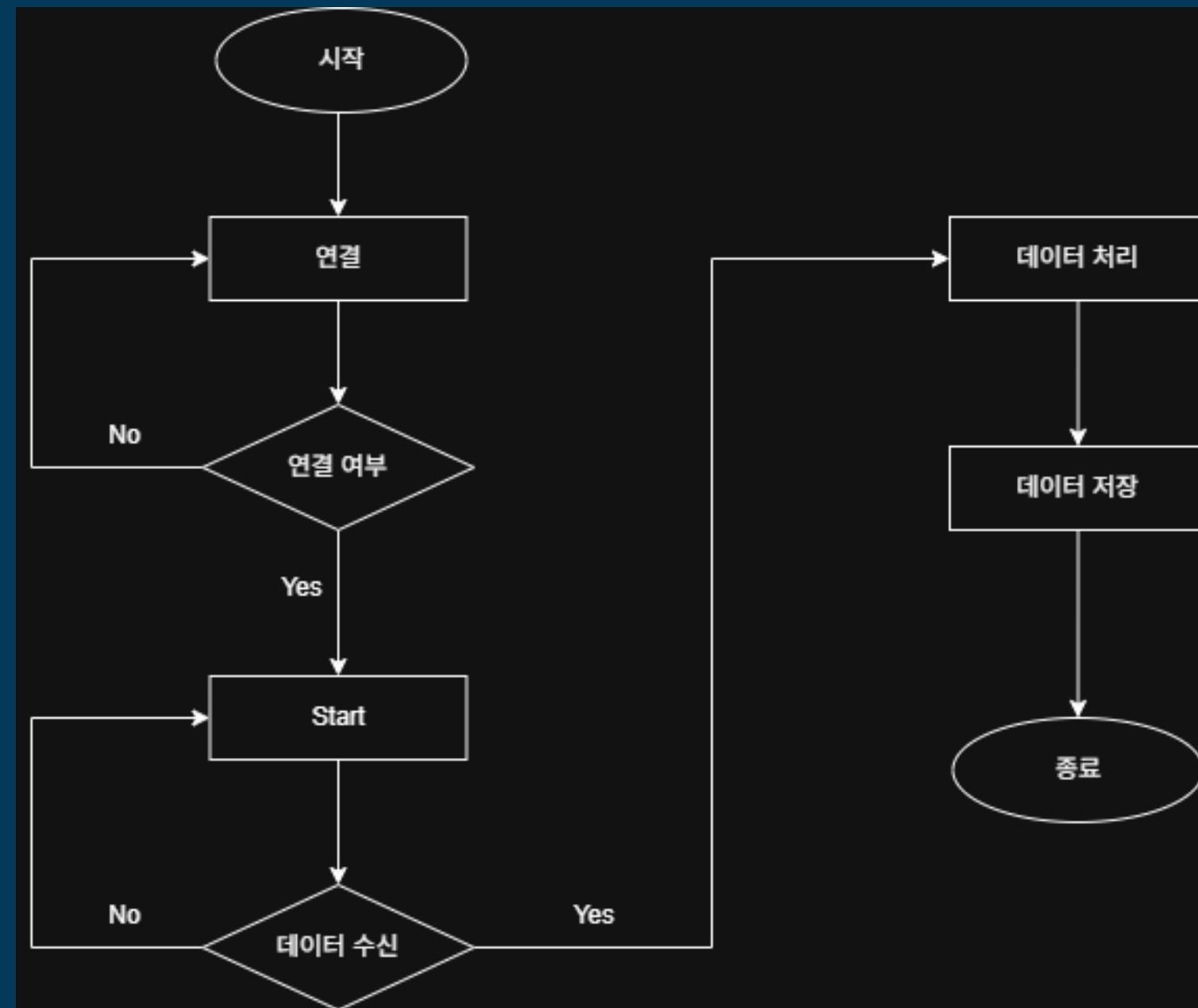
1. 프로젝트 개요

본 프로젝트는 **TCP/IP Socket** 기반으로
서버로부터 공기 질 데이터를 실시간
수신 및 모니터링하는 시스템입니다.

온도, 습도, 산소, 이산화탄소, 미세먼지,
초미세먼지 데이터를 수신하여
1분 단위 평균값을 계산하고 UI에 표시하며,
결과를 **CSV** 파일로 자동 저장합니다.

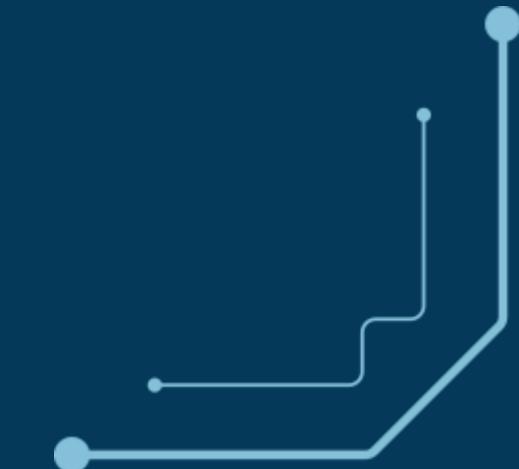
멀티스레딩을 적용하여 데이터
수신과 **UI 처리를 분리해**
실시간성과 안정성을 확보했습니다.

2. Flowchart

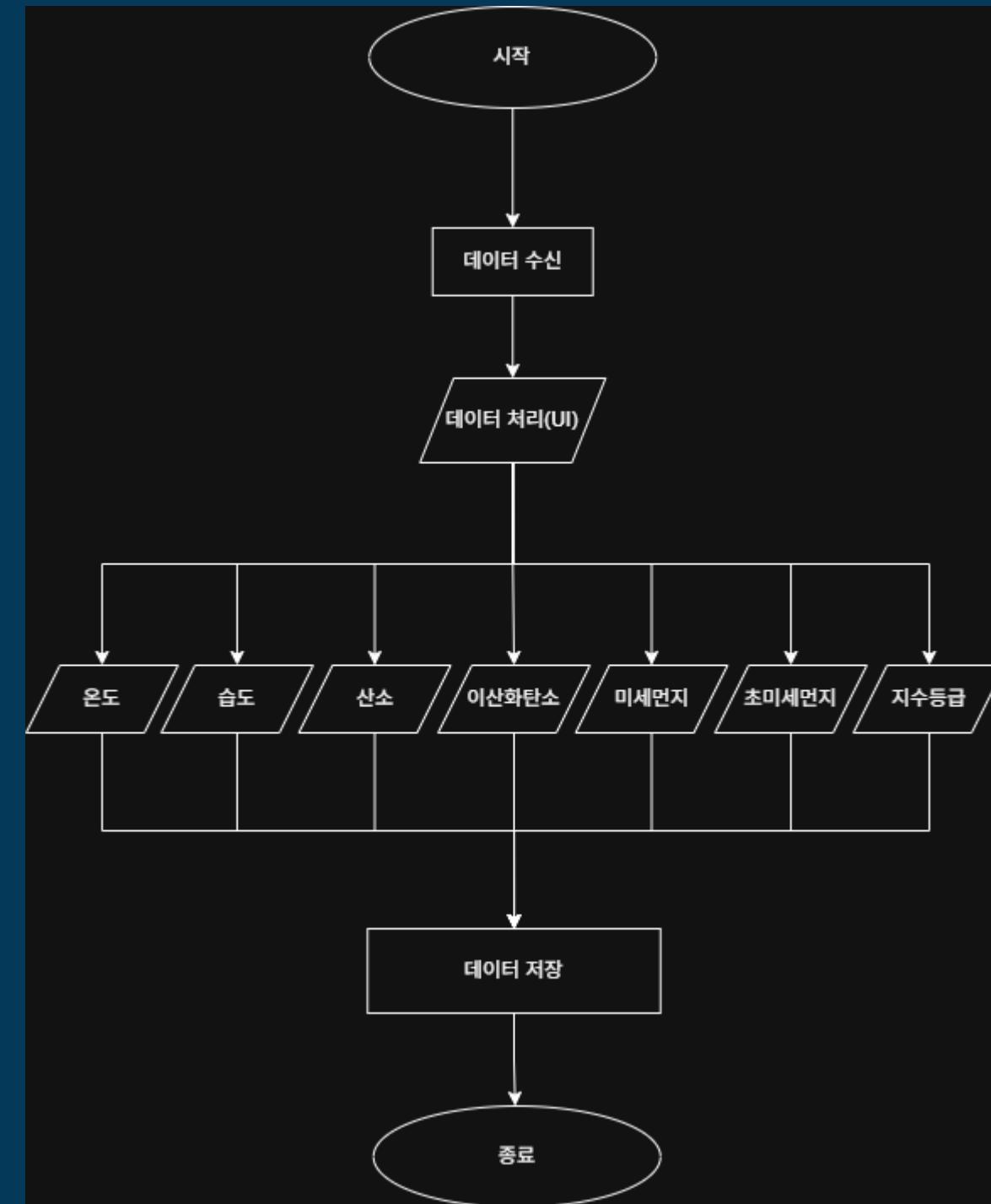
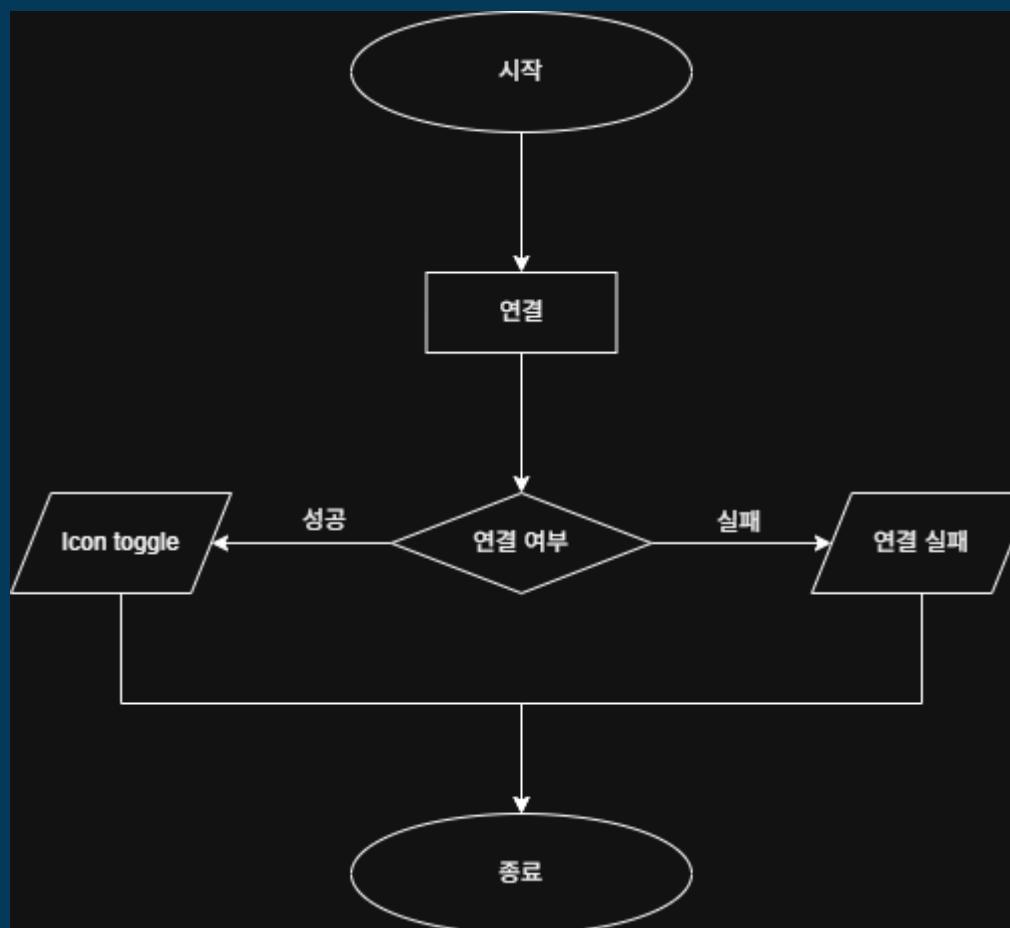
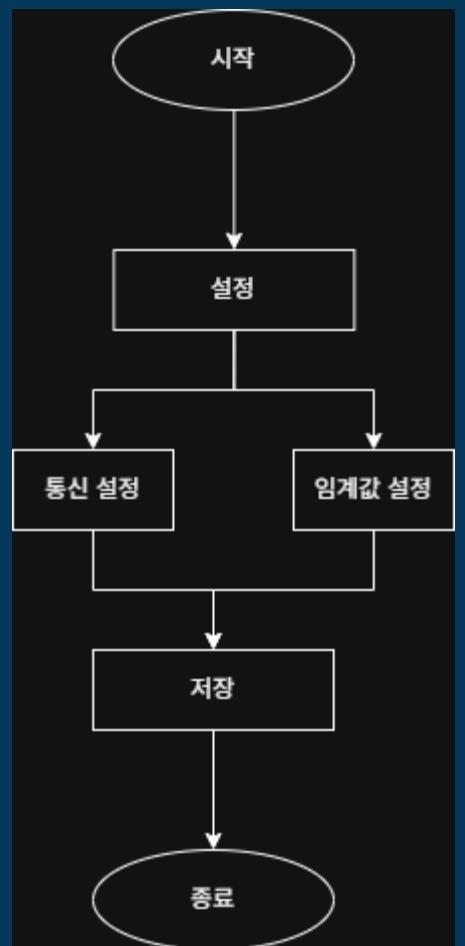


본 시스템은 연결 → 데이터 수신 → 처리 → 저장의 단계로
구성되어 있으며, 각 기능은 독립적인 흐름으로 설계되어
프로그램의 안정성을 높였습니다.

전체



2. Flowchart



설정부

연결부

데이터
수신부

3. Protocol

Protocol 규칙

- 기본적으로 요청/응답 방식으로 동작하며
Event가 발생한 경우에는 요청없이 즉시 응답한다.
- 통신 방식은 TCP 기반이다.
- Packet은 STX + DATA + ETX로 구성한다.
- STX는 0x02, ETX는 0x03이다.
- 클라이언트 요청(Request)에 대해
서버가 응답(Response)하는 방식으로 동작한다.

COMMAND

KICK	서버에 데이터 수신 시작을 요청
Get Data	실시간 센서 데이터 전송을 요청
STOP	서버의 데이터 송신 중지 및 세션 종료를 요청

데이터 수신시작

SENDER	FORMAT
HOST	KICK
Server	ACK

데이터 송신 중지

SENDER	FORMAT
HOST	STOP
Server	STOP_ACK

센서 데이터 수집

SENDER	FORMAT
HOST	Get Data
Server	DATA,Time,Temp,Hum,O2,CO2,PM10,PM25

4. UI 및 주요코드

4-1. UI



4. UI 및 주요코드

4-2.

주요코드

참조 4개

```
public void UpdateAverage(DataAverage avg)
{
    if (this.InvokeRequired)
    {
        this.Invoke(new Action(() => UpdateAverage(avg)));
        return;
    }

    SetValueVisible(true); // 최초 데이터 수신 시 표시

    lb_temp.Text = $"{avg.Temp.ToString("F1")} °C";
    lb_hum.Text = $"{avg.Hum.ToString("F1")} %";
    lb_oxy.Text = $"{avg.O2.ToString("F1")} %";
    lb_co2.Text = $"{avg.CO2.ToString("F0")} ppm";
    lb_pm10.Text = $"{avg.PM10.ToString("F0")} µg/m³ ";
    lb_pm25.Text = $"{avg.PM25.ToString("F0")} µg/m³ ";

    var settings = SettingsData.Load().Alerts;
    ApplyGrade(lb_tem_grade, pb_tem_grade, GetGrade(avg.Temp, settings.Temperature));
    ApplyGrade(lb_hum_grade, pb_hum_grade, GetGrade(avg.Hum, settings.Humidity));
    ApplyGrade(lb_oxy_grade, pb_oxy_grade, GetGrade(avg.O2, settings.Oxygen));
    ApplyGrade(lb_co2_grade, pb_co2_grade, GetGrade(avg.CO2, settings.CO2));
    ApplyGrade(lb_pm10_grade, pb_pm10_grade, GetGrade(avg.PM10, settings.PM10));
    ApplyGrade(lb_pm25_grade, pb_pm25_grade, GetGrade(avg.PM25, settings.PM25));
}
```

데이터 수신부

참조 1개

```
private void HandleServerMessage(string msg)
{
    MessageDisplay($"[수신] {msg}");

    if (msg == "ACK")
    {
        client.SendMessage("Get_Data");
        MessageDisplay("[송신] Get_Data");
    }
    else if (msg == "STOP_ACK")
    {
        MessageDisplay("서버 데이터 송신 중지 확인");
    }
    else if (msg.StartsWith("DATA") && isRunning)
    {
        ProcessSensorData(msg);
    }
}
```

Packet 처리

5. 실행결과



UI



Log

2026-02-04 20:52:47.743	Program Open
2026-02-04 21:03:13.978	서버(127.0.0.1:6000) 연결됨
2026-02-04 21:03:15.594	[송신] KICK
2026-02-04 21:03:15.604	[수신] ACK
2026-02-04 21:03:15.605	[송신] Get_Data
2026-02-04 21:03:15.624	[수신] DATA, 21:03:15, 0, 43, 88, 478, 43, 6
2026-02-04 21:03:16.636	[수신] DATA, 21:03:16, 1, 4, 52, 1187, 183, 85
2026-02-04 21:03:17.648	[수신] DATA, 21:03:17, 8, 98, 73, 1008, 116, 2
2026-02-04 21:03:18.664	[수신] DATA, 21:03:18, 29, 99, 15, 1044, 58, 79
2026-02-04 21:03:19.671	[수신] DATA, 21:03:19, 15, 57, 26, 1984, 73, 14
2026-02-04 21:03:20.686	[수신] DATA, 21:03:20, 9, 43, 87, 1689, 10, 54
2026-02-04 21:03:21.701	[수신] DATA, 21:03:21, 14, 97, 45, 234, 150, 16
2026-02-04 21:03:22.707	[수신] DATA, 21:03:22, 35, 88, 46, 189, 164, 60
2026-02-04 21:03:23.710	[수신] DATA, 21:03:23, 33, 49, 68, 240, 66, 77
2026-02-04 21:03:24.717	[수신] DATA, 21:03:24, 17, 47, 14, 187, 169, 64
2026-02-04 21:03:25.726	[수신] DATA, 21:03:25, 32, 3, 68, 756, 72, 47
2026-02-04 21:03:26.728	[수신] DATA, 21:03:26, 39, 37, 93, 379, 29, 61
2026-02-04 21:03:27.751	[수신] DATA, 21:03:27, 22, 61, 24, 1989, 126, 40
2026-02-04 21:03:28.761	[수신] DATA, 21:03:28, 13, 29, 75, 1316, 32, 87
2026-02-04 21:03:29.771	[수신] DATA, 21:03:29, 18, 11, 10, 204, 81, 57
2026-02-04 21:03:30.781	[수신] DATA, 21:03:30, 15, 26, 12, 1611, 62, 72
2026-02-04 21:03:31.793	[수신] DATA, 21:03:31, 19, 51, 52, 1966, 22, 44
2026-02-04 21:03:32.846	[수신] DATA, 21:03:32, 26, 63, 34, 148, 109, 71
2026-02-04 21:03:33.860	[수신] DATA, 21:03:33, 21, 44, 84, 1495, 152, 55
2026-02-04 21:03:34.874	[수신] DATA, 21:03:34, 35, 14, 86, 1627, 183, 24
2026-02-04 21:03:35.890	[수신] DATA, 21:03:35, 14, 87, 97, 717, 158, 88

CSV

A	B	C	D	E	F	G
Time	Temp	Hum	O2	CO2	PM10	PM2.5
2026-02-04 21:04	21.3	51.3	52.6	811	105	51
2026-02-04 21:05	20	44.3	50.7	985	97	54
2026-02-04 21:06	19.6	47.4	49.7	1046	103	52

Thank You

