

# 개발 완료 보고서

제출일 : 2025-02-21

참여인원	김도엽		
프로젝트 명	[LMS5 20/30] Project08_C#_WPF_Resource Monitoring Tool		
활동일시	2025.02.17 ~ 2025.02.21	장소	드론융합실
주요주제	C# WPF에 비동기 서버 클라이언트 작업, SQLite 데이터베이스 연동		
개발목적	컴퓨터 메모리, CPU 사용량 등 클라이언트에서 자원을 수집해서 서버로 보내 컴퓨터 리소스 정보를 시각화하기 위함		
개발환경	Windows 11, Microsoft Visual Studio C#, DBeaver, SQLite		
구현기능	<ul style="list-style-type: none"><li>- 컴퓨터 자원 수집하기</li><li>- 서버 / 클라이언트 비동기 통신</li><li>- DB 연동</li></ul>		
코드설명	<p>&lt;서버&gt;</p> <ul style="list-style-type: none"><li>- ReadNoticeDB : DB 파일 연결, 리스트 변수에 DB에서 랜덤으로 가져온 값 30개를 넣고 ListView에 값 전달</li><li>- Button_Click : 서버 구동 버튼, IP, Port 등으로 서버 오픈</li><li>- AcceptClient : 서버 소켓에 클라이언트 받아들이기 준비</li><li>- ClientAccepted : 클라이언트가 연결되면 준비된 소켓이랑 연결</li></ul>		

	<ul style="list-style-type: none"> <li>- ReceiveInfo : 클라이언트에서 데이터 받아옴</li> <li>- DataReceived : 받은 데이터를 역직렬화 해서 문자열로 바꾸고 클래스 파일에 반영, 수신 작업 다시 수행</li> <li>- RefreshDeviceInfo : 클래스 파일에 저장된 데이터를 가져와서 형식에 맞게 수정 후 UI에 출력 및 DB에 저장</li> <li>- AddLog : 서버 - 클라이언트 간에 주고 받는 데이터 형태를 로그로 남김</li> </ul> <p>&lt;클라이언트&gt;</p> <ul style="list-style-type: none"> <li>- memoryUpdate : 컴퓨터 자원들 수집해서 변수에 저장하고 차트에 반영</li> <li>- ConnectButton_Click : 서버 연결 버튼, 입력한 IP, 포트 번호를 통해 서버 연결</li> <li>- ServerConnecteed : 서버 연결 성공 여부 확인</li> <li>- ReceiveControl : 보낼 데이터 직렬화</li> <li>- ControlReceived : 클래스 파일에 저장된 데이터 제이슨 문자열로 변환</li> <li>- timer_Tick : 타이머 1번 발생할 때 넘어오는 함수, 데이터 전송 함수들 있음</li> <li>- SendDeviceInfo : 소켓, 서버 연결 확인하고 전송할 데이터 객체에 담아서 준비하고 비동기적으로 서버에 보냄</li> </ul>
--	---

	<p>- ClientWindow_Loaded : 클라이언트 창 로드될 때 발생, 타이머 이벤트 발생</p>
후기	<p>이번에도 WPF에 대한 자료가 많이 없고, 또 그 안에서 라이브차트는 명령어를 도대체 어떻게 써야하는 건지, 자료가 더더욱 전무했다.</p> <p>그 때문에 차트를 활용하는 법을 조사하는데 시간을 제일 많이 썼다. 작업관리자 처럼 초마다 값이 누적되는 그래프를 만드는 것이 이렇게 어려울 줄이야..막상 코드를 치고 보니 길진 않았다.</p> <p>서버 - 클라이언트 연결은 예제를 보면서 열심히 따라했는데, 비동기적으로 전송되는 부분 때문에 내가 원하는 데이터를 보낼 때 어느 부분에 집어넣느냐가 중요했다.</p> <p>데이터 저장을 아무데나 하면 서버로 제대로 가지 않거나 창이 강제 종료되곤 했다.</p> <p>실시간 데이터를 보내는 데에도 좀 벅찼고 결과물은 부실한 거에 비해 들인 노력은 다른 프로젝트에 비해 결코 적지 않았다.</p> <p>단, 초반에 구상했던 계획과는 다르게 좀 버린 기능들이 많아서 차트에 투자한 시간들이 조금 아깝게 느껴지기도 했다. 지금 내 프로그램은 한 클라이언트의 데이터만 받는다고 상정하고 있으니... 과거 데이터 확인도 글로 확인하고 있으니 좀 아쉽다.</p> <p>프로젝트06 서버 - 클라이언트 다중 채팅방 만들었을 때처럼 충분한 사전 조사를 거치고 했다면 어땠을까 싶다.</p>