

경험 기술서



김혁중 Hyuk Joong Kim

생년월일 2000.11.25 (만 24)
전화번호 82+ 10-9412-4572
이메일 x001125@gmail.com
깃허브 <https://github.com/Dongubak>

| 경험기술 | 프로그래밍 언어 |
|---------------------|---------------------------------------|
| 기술 | 설명 |
| JavaScript(ES6) | 자바스크립트 기반 프레임워크 사용 경험, 웹 어셈블리 사용 경험 |
| C / Embedded C | AVR 기반 MCU 펌웨어 제작 경험 |
| C++ | 자료구조 / 알고리즘 활용 |
| 프레임워크 / 라이브러리 | |
| 기술 | 설명 |
| React / Redux | SPA 구현, 상태관리 구조 설계 |
| Redux-saga | 리덕스 미들웨어 사용 및 Redux-DevTools 사용 경험 |
| Node.js / Koa | REST API, JWT 토큰 인증, Swagger 명세 작성 경험 |
| Emscripten | 웹 어셈블리 라이브러리 활용 경험 |
| 서버 / 데이터베이스 | |
| 기술 | 설명 |
| MySQL | DB 설계 및 Sequelize 연동 |
| Sequelize | ORM 기반 DB 접근 경험 |
| 임베디드 / 회로설계 | |
| 기술 | 설명 |
| ATmega328P | 회로 설계 및 펌웨어 제작 AVR |
| Autodesk Fusion 360 | 회로도 설계 |

| 프로젝트 요약 | 기간 | 제목 | 성과 | 역할 |
|------------|------------------------|-------------------|------------|----------------|
| | 2024. 04 ~ 2024. 12 | 군산대 시간표 기반 커뮤니티 앱 | 캡스톤 디자인 동상 | 팀장 / 개발 총괄 |
| | 2025. 02 ~ 2025. 04 | 동아리 공식 홈페이지 제작 | 개인 프로젝트 | 프론트, 백엔드, DB |
| | 2025. 03 | 습도 감지 제어기 | 개인 프로젝트 | 회로 설계 및 펌웨어 개발 |
| | 2025. 05 ~ 2025. 06 | 실회기 보호 장치 제어기 | 개인 프로젝트 | 회로 설계 및 펌웨어 개발 |
| | 2024. 12 ~ 2025. 04 | 자율주행 로봇 제어 시스템 | 캡스톤 디자인 입상 | 자율주행 로직개발 |
| | 2025. 05 ~ 2025. 11 | 사물 운송 로봇 | 캡스톤 디자인 입상 | 모빌리티 파트 담당 |

군산대 시간표 기반 커뮤니티 앱
3학년 1학기~2학기
2024.04~2024.12
캡스톤디자인(팀장)

군산대학교 학생들을 위한 시간표 기반 커뮤니티 애플리케이션을 설계하고 개발하였다. 백엔드는 Node.js의 Koa 프레임워크를 활용하고 JWT 기반 인증 시스템을 구축했으며, Sequelize ORM으로 MySQL 테이블을 설계하고 연동하였다. 시간표 데이터는 군산대 수강편람을 기반으로 사용자가 시간표를 저장 및 비교할 수 있도록 기능을 구현하였으며, 사용자 간 시간표 겹침 여부 확인 기능도 포함했다. 프론트엔드는 React 기반으로 구성되었고, 상태관리는 Redux 및 Redux-Saga를 통해 설계하였다.

경험기술
React, Redux, Redux-Saga, Node.js(v17), MySQL, Sequelize, JWT,

News API를 통해 최신 뉴스를 받아오는 기능과, 군산대 식단표 크롤링 서버를 구축하여 별도 API로 제공하는 기능도 개발하였다. 기획, 개발, 배포까지 전체 일정을 주도한 프로젝트로, Git Flow 기반 브랜치 전략을 통해 협업과 버전관리를 효율적으로 수행하였다. Git Flow 전략을 도입한 이유는 배포용 브랜치와 개발용 브랜치를 분리하여 안정적으로 배포 버전을 관리하고, 기능 개발 중 발생할 수 있는 변경사항과 충돌을 사전에 격리하여 형상 관리를 체계화하기 위함이다. 이를 통해 병렬 개발과 긴급 패치가 동시에 가능해졌으며, 팀원 간의 협업 효율성도 크게 향상되었다.

- ✓ **프론트엔드 렌더링 성능 최적화 (React + Redux)**
 프로젝트에서 다량의 더미 데이터를 렌더링할 경우 약 130.0ms의 렌더링 지연이 발생했다. Profiler와 Redux-DevTools를 활용해 타입을 조회한 후, 상태 변경 시 불필요한 리렌더링이 발생하는 것을 확인하고, useMemo와 useCallback을 도입해 의존성 배열을 재정비하였다. 그 결과 렌더링 시간은 15.4ms로 약 88% 감소했으며, 사용자 체감 성능을 크게 향상시켰다.
- ✓ **Redux 미들웨어 흐름 검증 및 디버깅 체계 구축**
 Redux + Redux-Saga를 도입한 상태 관리 구조에서, Redux DevTools를 활용해 액션 흐름과 미들웨어 작동 여부를 실시간으로 확인하였다. Redux는 보일러플레이트 코드가 많다는 단점이 있음에도 불구하고, 액션을 명시적으로 정의함으로써

디버깅이 용이하고 상태 추적이 명확하다는 장점이 있어 도입하였다. 이 프로젝트에서는 각 액션의 발생 시점과 상태 변화 과정을 시각적으로 추적할 수 있었으며, 여러 처리 로직을 액션별로 세분화하여 사용자 피드백 및 실패 시 처리를 명확하게 구성함으로써 디버깅 시간과 유지보수 효율성을 향상시켰다.

동아리 공식 홈페이지 제작

4학년 1학기~1학기

2025. 02~2025. 04

개인 프로젝트

경험기술

React, Redux,
Redux-Saga,
Node.js(v17), MySQL,
Sequelize, JWT,
React-Quill, highlight.js

본 프로젝트는 프로그래밍 동아리 내부의 소통과 학습 효율을 높이기 위해 제작한 동아리 전용 홈페이지로, 단순한 커뮤니티 기능을 넘어 코드 중심의 Q&A 플랫폼으로 발전시킨 프로젝트이다.

기존에 제작했던 커뮤니티 애플리케이션에서는 코드를 보기 좋게 포매팅하여 업로드할 수 없다는 한계가 있었다.

이에 따라, 동아리원들이 코드 관련 질문을 보다 명확하게 공유하고, 언어별로 가독성 높은 코드 블록을 표시할 수 있도록 전용 에디터를 새로 설계·개발하였다.

이 과정에서 기존 커뮤니티 애플리케이션의 구조(회원 인증, 게시판, 상태관리, 서버 API 등)는 대부분 재사용 가능한 형태로 리팩터링되어 활용되었고, 신규 기능인 React-Quill 기반 커스텀 에디터가 핵심적으로 추가되었다.

에디터는 highlight.js를 통해 언어별 코드 구문 하이라이팅을 지원하며, 사용자는 코드 조각(Code Snippet)을 게시글에 삽입하거나, 마크다운 문법을 활용해 다양한 서식을 적용할 수 있다.

이로써 동아리원들은 단순 텍스트 기반 질문이 아닌, 실제 코드 예시를 포함한 실무형 Q&A 게시글을 작성할 수 있게 되었다.

이 기능은 동아리 내에서 큰 호응을 얻었으며, 실제로 회원들이 코드 실행 결과나 에러 로그를 포함해 질문을 작성하고, 다른 회원들이 이에 대한 답변을 코드와 함께 남기면서 협업 중심의 기술 토론 문화가 형성되었다.

| | |
|---|--|
| 습도 감지 제어기 4학년 1학기 2025.03~2025.5 개인 프로젝트 경험기술 AVR C, ATmega328P, PWM, 릴레이 회로, Autodesk Fusion 360 | <p>습도 변화에 따라 환풍기를 자동으로 ON/OFF 하는 임베디드 시스템을 개발하였다. 이 프로젝트는 ATmega328P의 데이터시트를 기반으로 C 언어로 직접 펌웨어를 작성한 프로젝트이다. 마이크로컨트롤러의 레지스터 설정과 입출력 제어를 처음부터 구성하면서 펌웨어의 구조를 깊이 있게 이해할 수 있었고, 회로도 역시 Autodesk Fusion 360을 활용해 직접 설계함으로써 하드웨어와 소프트웨어 간의 연결 구조를 처음부터 끝까지 구축했다. ATmega328P를 사용하여 DHT11 센서로부터 습도값을 읽고, 설정 임계값 이상일 경우 릴레이를 작동시켜 환풍기를 제어하였다.</p> <p>회로는 Autodesk Fusion 360으로 설계하였고, 펌웨어는 C 언어 기반으로 직접 작성하였다. ADC를 통한 아날로그 데이터 수집, PORT 레지스터 조작, PWM 신호 발생 등 기본적인 제어 기술을 적용하였다.</p> <ul style="list-style-type: none"> ✓ 맥 환경에서 AVR 개발 환경 구축 <p>macOS에서는 통상적인 IDE를 설치하기 어렵기 때문에 CLI 기반으로 avr-gcc 툴체인을 구성하였다. 다양한 컴파일 옵션을 적용하여 ELF 및 HEX 바이너리를 생성하고, VSCode의 tasks.json을 통해 열려있는 파일 기준 자동 컴파일 및 업로드를 실행하도록 설정하였다. 반복적인 빌드 작업을 자동화함으로써 개발 속도와 작업 효율성을 크게 향상시켰다.</p> |
| 실외기 보호 장치 제어기 4학년 1학기 2025.06~2025.07 개인 프로젝트 경험기술 AVR C, ATmega328P, Autodesk Fusion 360, avr-gcc, avrdude, 외부 인터럽트, UART | <p>에어컨이 작동 중인 상태를 감지하여 실외기가 켜질 경우, 릴레이를 통해 외부 모터(가림막)를 자동 작동시키는 보호 시스템을 설계 및 구현하였다. 추가로 온도 센서를 통해 실외기 온도를 측정하고, 설정 온도 이상이 되면 외부 인터럽트를 통해 위험 신호(LED, 부저 등)를 발생시키는 안전 제어 로직을 구현하였다.</p> <p>회로는 Autodesk Fusion 360을 사용하여 전체 구조를 설계하였고, 센서 입력 및 릴레이 출력 제어는 모두 ATmega328P 마이크로컨트롤러를 기반으로 이루어졌다. ADC를 활용하여 아날로그 온도 센서 값을 디지털로 변환하였으며, UART 통신을 통해 디버깅 로그 및 상태 확인이 가능하도록 하였다.</p> <p>에어컨 작동 신호를 디지털 입력으로 감지하고, 이를 릴레이 출력과 연동하여 가림막 모터를 작동시켰다. 또한 UART 통신을 활용하여 실외기의 상태를 직렬 통신으로 실시간 보고받는 구조를 구현하였으며, 이를 통해 마이크로컨트롤러 간 통신 기능 직접 구성한 프로젝트이다. 센서 기반 제어뿐만 아니라 기기간 통신을 포함한 통합 제어 구조를 구성함으로써, 보다 정밀하고 안정적인 시스템을 구축할 수 있었다. 위험 온도 초과 시 인터럽트를 통해 즉각적인 알림이 작동되도록 설계함으로써, 시스템 안정성과 사용자 안전을 동시에 확보하였다.</p> <p>본 프로젝트를 통해 마이크로컨트롤러 수준에서의 인터럽트 설계 및 제어 흐름 구조를 경험하였으며, 실시간 대응과 아날로그-디지털 변환 기반의 센서 처리 능력을 향상시킬 수 있었다.</p> |

**자율주행 로봇
제어 시스템**

4학년 1학기

Jetson Xavier NX에서 RPLidar를 통해 장애물을 인식하고, 인식된 데이터를 바탕으로 회피 경로를 계산하여 Arduino에 명령을 전달하는 구조로 자율주행 로봇을 개발했다. ROS2를 기반으로 하여 LiDAR 데이터를 수신하고 /scan 토픽을 통해 장애물 거리 정보를 획득했으며, 이를 바탕으로 /direction 토픽으로 방향 명령을 publish하는 노드를 구현했다.

2024. 12 ~ 2025. 04

캡스톤디자인

경험기술

ROS2, RPLidar,

Arduino, PWM, Serial

통신, Jetson Xavier NX

Arduino는 Jetson과シリ얼 통신을 통해 데이터를 받아서 스텝모터를 구동하였다. PWM 제어로 정밀한 방향 제어가 가능하게 하였으며, 로봇이 실시간으로 장애물을 회피하며 주행할 수 있도록 시스템을 구성하였다. Jetson 측에서는 Python 기반 ROS2 노드를 직접 작성했으며, DDS 통신구조를 학습하고 실험하여 안정적인 데이터 전송 구조를 설계하였다.

자율주행 알고리즘 개발을 위해 학습 데이터를 별도로 수집하는 아키텍처를 설계하였다. 수집 단계에서는 RPLidar와 PS2 컨트롤러를 Arduino Mega에 연결하고, 이를 통해 측정된 거리 정보와 조작 방향을 실시간으로 Mac에シリ얼 통신으로 전송하여 CSV 파일 형태로 저장하였다. 이 구조는 ROS2 없이 구성된 경량 아키텍처로, 수동 조작 기반의 지도 학습용 데이터를 확보하는 데 초점을 두었다.

학습 단계에서는 Jetson Xavier NX 상에서 PyTorch를 사용하여 간단한 다층 퍼셉트론 모델을 구성하고, 수집된 CSV 데이터를 기반으로 방향 예측 모델을 학습하였다. 학습 환경은 ROS2와 분리되어 있으며, 모델 학습 후에는 ROS2 기반 추론 노드와 연동하여 /scan 입력에 따른 /direction 출력이 가능하도록 아키텍처를 분리·연계하였다.

하드웨어 플랫폼은 오픈소스 커뮤니티의 로봇 도면을 참고하여 직접 3D 프린터로 출력하고 조립하였다. 부품 선정, 회로 설계, 모터 드라이버 배선, 전원 구성 등 하드웨어 구축 전반을 스스로 담당했으며, 저렴한 가격으로 실현 가능한 자율주행 플랫폼을 목표로 센서, 연산 장치, 제어 시스템을 통합하였다.

사물 운송 로봇

4학년 2학기

2025. 05 ~ 2025. 11

캡스톤디자인

경험기술

ROS2, RPLidar,
Arduino, micro-ros,
Serial comm, docker,
Jetson nano, SLAM,
Odometry, AS5600

Jetson nano 환경에서 micro-ros로 실행되는 좌·우 훨의 오도메트리 데이터를 융합하고, RPLidar를 이용하여 실시간 지도 작성 및 위치 추정을 수행하는 차륜형 자율주행 로봇 시스템을 개발하였다.

로봇의 좌우 훨에는 AS5600 자기 엔코더를 장착하여 바퀴 회전수를 측정하였으며, Python 기반 ROS2 노드에서 각 훨 속도를 수신하여 /odom 프레임을 계산하였다.

해당 노드는 좌·우 바퀴 속도를 평균 및 차로 변환하여 선속도(v)와 각속도(ω)를 계산하고, 이를 통해 로봇의 (x, y, yaw) 상태를 적분하여 /odom 토픽과 $odom \rightarrow base_link$ 변환을 퍼블리시하도록 구성하였다.

오도메트리 융합으로 추정된 로봇 위치 정보는 slam_toolbox 패키지와 연동되어 LiDAR 데이터(/scan)를 이용한 map \rightarrow odom 정합(SLAM) 과정으로 통합되었다. 이를 통해 오도메트리 누적 오차를 보정하고, 로봇의 위치와 환경 지도를 동시에 추정할 수 있도록 하였다.

시스템 구조는 ROS2의 Pub/Sub 통신 구조 및 TF 프레임워크를 중심으로 설계되었으며, Jetson 측에서 LiDAR 데이터를 수신하고 오도메트리 융합을 수행하는 Python 노드를 실행한다.

로봇은 LiDAR 기반 지도 정합 결과를 바탕으로 실시간으로 이동 경로를 추적하고, 장애물 환경에서도 안정적인 주행이 가능하도록 구현되었다.

하드웨어는 오픈소스 터틀봇 구조를 기반으로 자체 3D 프린팅 및 조립을 수행하였으며, 엔코더, 모터 드라이버, 배터리 분배 모듈 등 핵심 부품의 배선 및 전원 설계를 직접 진행하였다.

특히 ROS2 프레임 구조($map \rightarrow odom \rightarrow base_link \rightarrow laser$)를 명확히 구성하여 각 센서의 좌표계 관계를 정립하고, URDF 및 TF 시각화를 통해 로봇 구조의 공간적 정합성을 확보하였다.

본 프로젝트는 단순한 센서 데이터 처리 수준을 넘어, 하드웨어 통합-센서 융합-SLAM 정합-TF 구조 설계까지 포괄하는 실시간 자율주행 로봇 플랫폼 구축을 목표로 수행되었다.