

SoC / RTL 설계 직종 프로젝트 포트폴리오

설원혁

010-3311-6680

swh05284@gmail.com



Wormboiii

약력



Academic

- 2019 지산고등학교 졸업
- 2019 홍익대학교 세종캠퍼스 전자•전기융합공학과 학사 입학
- 2025 홍익대학교 세종캠퍼스 전자•전기융합공학과 학사 졸업

EXPERIENCE

- 2022.12~2023.12 홍익대학교 세종캠퍼스 정동아리 '오키드' 회장 재임
- 2023.12~2024.11 홍익대학교 세종캠퍼스 32대 총동아리연합 '春夏秋冬' 사무국장 재임

Education

- 2023.12~2024.01 코멘토 '눈으로 하는 반도체 설계(CHIP)' 과정 수료
- 2025.02 SEMI社 주최 'SEMICON KOREA' 컨퍼런스 참관
- 2025.09 Digikey社 주최 'Engineering labs' 웨비나 참여
- 2025.05~2025.10 대한상공회의소 경기인력개발원 '세미콘 설계/검증' 과정 이수(예정)

설원혁

- 2000.05.28
- 경기도 파주시 가온로 256
- 010-3311-6680
- swh05284@gmail.com

LICENSE

- 자동차운전면허증(1종 보통)
- YBM TOEIC 900

기술 스택



FPGA



임베디드



회로 설계•검증



사무



참여 프로젝트 목록			
	엘리베이터 프로젝트 (ARM)	<ul style="list-style-type: none"> 기간 핵심 목표 담당 역할 획득 역량 	2025.06.18~2025.06.24 STM32 보드 기반 엘리베이터 구현 전체적인 알고리즘 구현 블로킹 코드의 한계 체감 및 timing 관련 디버깅 능력 증진
	엘리베이터 프로젝트 (SoC)	<ul style="list-style-type: none"> 기간 핵심 목표 담당 역할 획득 역량 	2025.09.15~2025.09.30(예정, WIP) 기존 엘리베이터 시스템을 SoC로 재설계 Peripheral 모듈화, IP 패키징 AXI 인터페이스를 통한 Peripheral 제어
	지하탈출 넘버원 (RTL)	<ul style="list-style-type: none"> 기간 핵심 목표 담당 역할 획득 역량 	2025.08.26~2025.09.01 Basys3 기반 지하주차장 재난 대응 시스템 구현 센서 및 출력부 구현, 회로 제작 기본적인 RTL 설계 능력 및 Verilog 언어 능력 증진
	자율주행 RC카 (ARM)	<ul style="list-style-type: none"> 기간 핵심 목표 담당 역할 획득 역량 	2025.07.03~2025.07.13 초음파 센서를 통한 자율주행 및 블루투스 컨트롤러 제작 블루투스 모듈을 이용한 컨트롤러 제작 RTOS를 통한 non-block 모사 능력 습득

<div> <div>참여 프로젝트 목록</div> </div>	
	<div> <div>입석 길잡이</div> <div>(IoT)</div> </div> <div> <ul style="list-style-type: none"> 기간2024.04.01~2024.11.25 핵심 목표웹의 DB와 센서값을 대조해 빈 자리를 원격 확인 가능 담당 역할AWS EC2를 통한 APM서버 구축, 웹과 ESP보드 연동 획득 역량IoT에 대한 전반적 이해도 및 리눅스 활용 능력 증진 </div>
	<div> <div>선풍기 프로젝트</div> <div>(AVR)</div> </div> <div> <ul style="list-style-type: none"> 기간2025.06.02~2025.06.09 핵심 목표ATMEGA128a 보드 기반 미니 선풍기 구현 담당 역할기능별 모듈화 및 팀원간 업무 분배, 3D 프린트 모델링 획득 역량레지스터단 제어를 통한 MCU의 근본적 동작 원리 이해 </div>
	<div> <div>베이스 이펙터 설계</div> <div>(Analog)</div> </div> <div> <ul style="list-style-type: none"> 기간2025.07.28~진행중 핵심 목표아날로그 증폭 회로를 통한 베이스 기타 이펙터 설계 담당 역할회로 설계 및 시뮬레이션, 검증, 아트웍(개인 프로젝트) 획득 역량TR, OP-AMP, RC filter 등을 이용한 아날로그 역량 증진 </div>

[ARM] 엘리베이터 프로젝트



엘리베이터 프로젝트

[목 적]

STM32 보드 기반 엘리베이터 시스템 구현

[목 표]

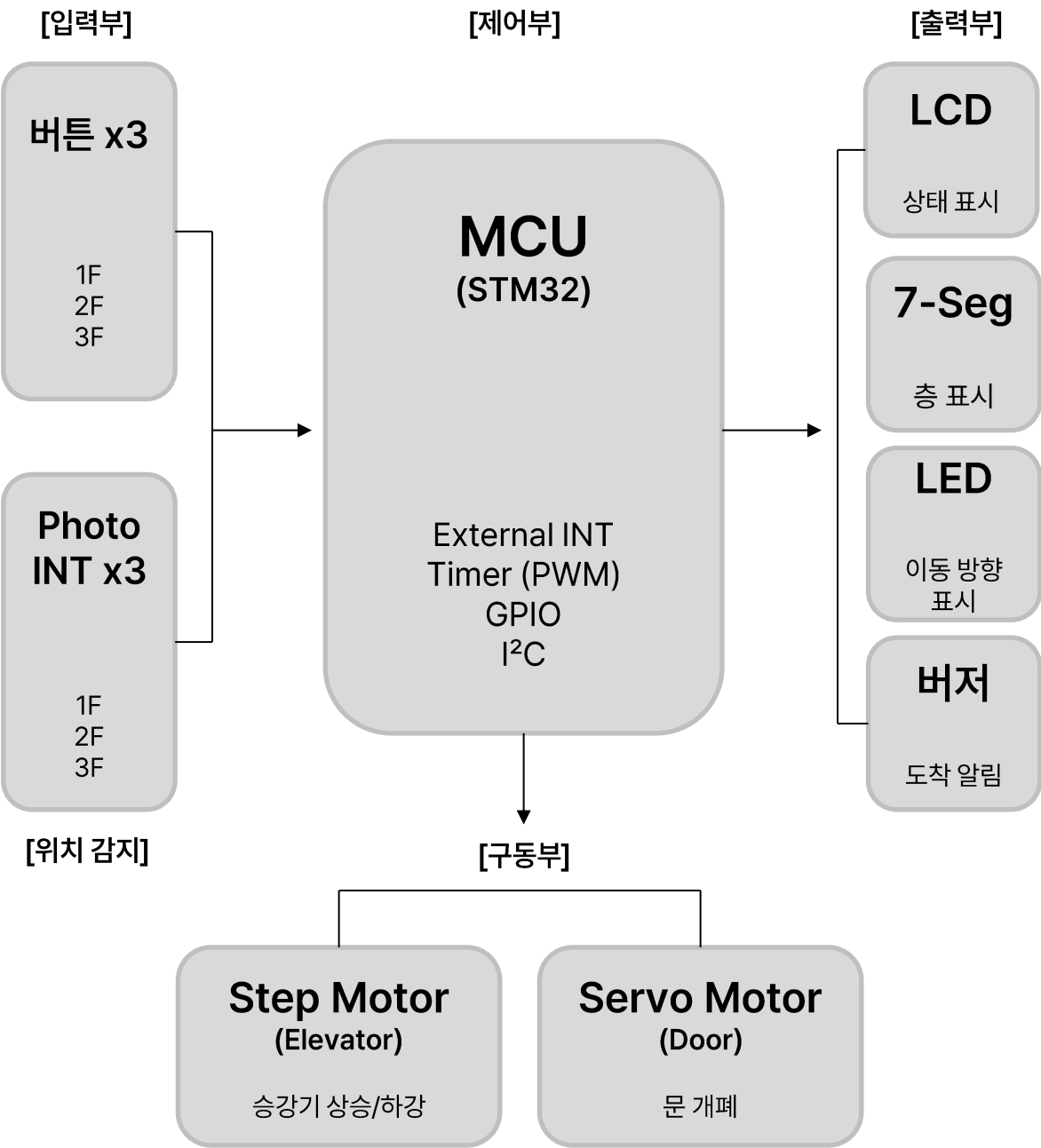
1. 버튼을 눌러 목표 층으로 이동 후, 도착 시 자동 문 개폐
2. 광인터럽터를 사용해 현재 층 인식
3. 각종 Peripheral을 통한 현재 상태를 알리는 UI 제공

- 개발 환경



[ARM] 엘리베이터 프로젝트 - 수행 내역

시스템 아키텍처 다이어그램



핵심 도전과제 & 해결

- Blocking 방식 코드의 한계

엘리베이터의 특성상 동시에 작동해야 하는 주변기기가 많지만 Blocking 방식 코드 구조를 사용해 특정 라인에 묶여 있어 전체적인 동작이 동시다발적으로 이루어지지 않는 문제 발생.

상태 Flag와 Interrupt를 적극적으로 사용, CPU가 특정 구문에 묶여있는 상황을 최소화해 완화.

위의 경험으로 Blocking 방식 코드 구조의 한계점을 체감할 수 있었고, 전체적인 코드 디버깅 능력을 증진할 수 있었음.

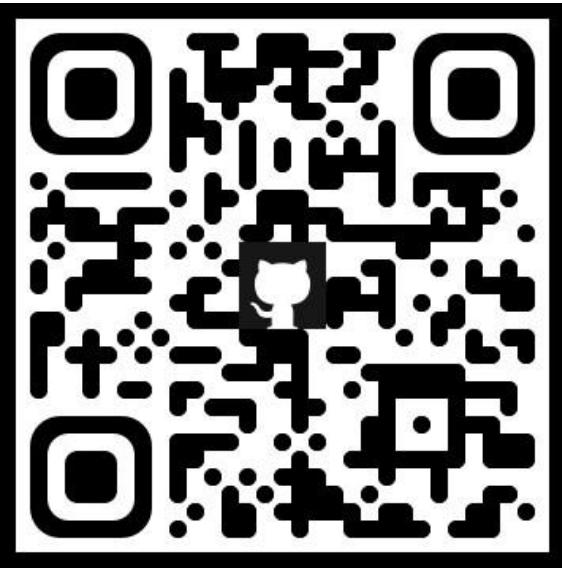
```
int photo_int_1_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 1층 광인터럽트.
int photo_int_2_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 2층 광인터럽트.
int photo_int_3_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 3층 광인터럽트.

int button_1_state; // 버튼의 상태를 나타내는 플래그 선언, 1층 버튼
int button_2_state; // 버튼의 상태를 나타내는 플래그 선언, 2층 버튼
int button_3_state; // 버튼의 상태를 나타내는 플래그 선언, 3층 버튼

int lcdState; // LCD의 상태를 나타내는 플래그 선언
int doorState; // 서보 모터의 상태를 나타내는 플래그 선언(문)
int buzzerState; // 버저의 상태를 나타내는 플래그 선언
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) { // 각각 3개의 광인터럽트와 버튼을 외부 인터럽트로 사용, 만약 신호
    if(GPIO_Pin == GPIO_PIN_8) { // c8에 연결되어있는 광인터럽트 1번(1층)으로 인터럽트 신호가 들어오면
        photo_int_1_state = 1; // 광인터럽트 1 플래그를 세우고
        photo_int_2_state = 0; // 다른 광인터럽트 플래그를 초기화.
        photo_int_3_state = 0;
        buzzerState = 1; // 버저 플래그를 세움.
    }
    else if(GPIO_Pin == GPIO_PIN_6) { // c6에 연결되어있는 광인터럽트 2번(2층)으로 인터럽트 신호가 들어오면
```

CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[SoC] 엘리베이터 프로젝트 (WIP)

엘리베이터 프로젝트

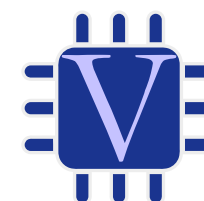
[목 적]

기존 엘리베이터 프로젝트를 SoC로 재설계해 개발 편의성 증진

[목 표]

1. 필요한 Peripheral들을 RTL 모듈 설계 후 IP로 패키징
2. 직접 설계한 IP와 기존 IP를 통합해 SoC 시스템 플랫폼 생성
3. Vitis를 통해 C언어 코드 개발 후 개발 과정 비교
4. IP Register, 개발한 Custom 함수 Library 등 데이터시트 작성

- 개발 환경



AMD
Vivado

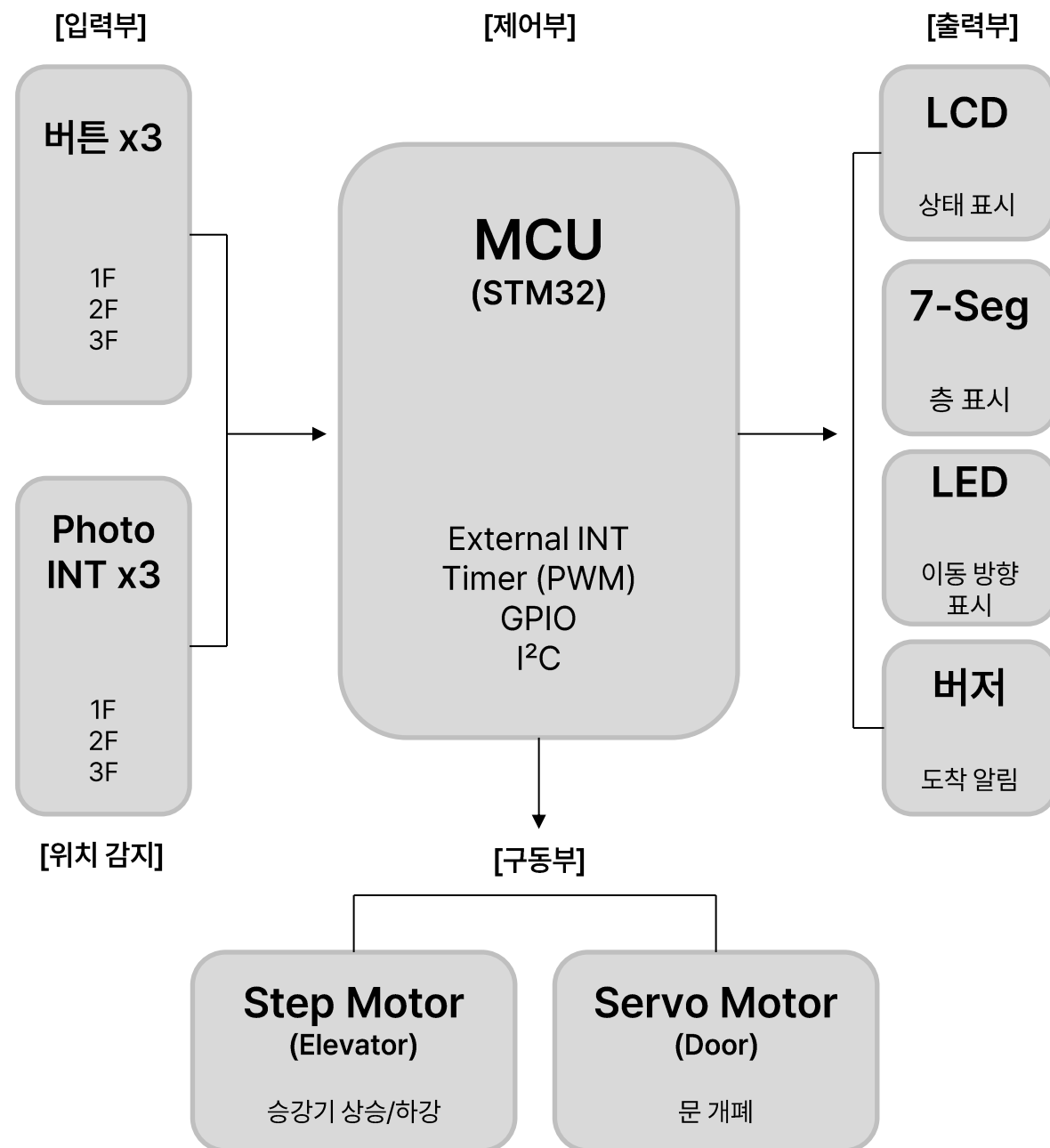
AMD
MicroBlaze

XILINX
VITIS™



[SoC] 엘리베이터 프로젝트 - 수행 내역

시스템 아키텍처 다이어그램



핵심 도전과제 & 해결

- Blocking 방식 코드의 한계

엘리베이터의 특성상 동시에 작동해야 하는 주변기기가 많지만 Blocking 방식 코드 구조를 사용해 특정 라인에 묶여 있어 전체적인 동작이 동시다발적으로 이루어지지 않는 문제 발생.

상태 Flag와 Interrupt를 적극적으로 사용, CPU가 특정 구문에 묶여있는 상황을 최소화해 완화.

위의 경험으로 Blocking 방식 코드 구조의 한계점을 체감할 수 있었고, 전체적인 코드 디버깅 능력을 증진할 수 있었음.

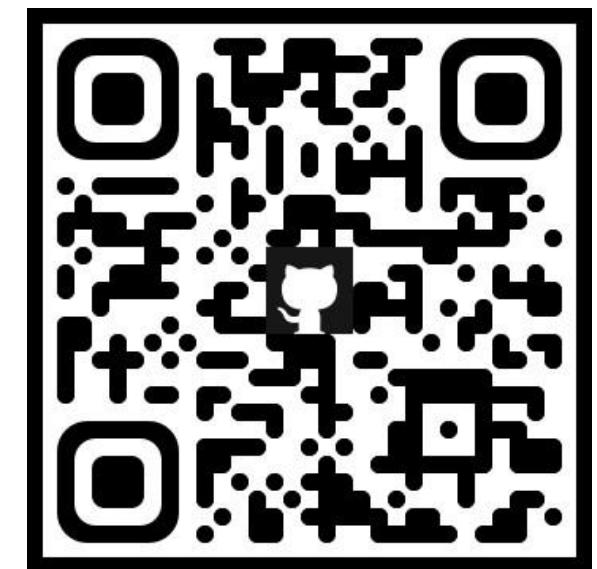
```
int photo_int_1_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 1층 광인터럽트.
int photo_int_2_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 2층 광인터럽트.
int photo_int_3_state; // 광인터럽트의 상태를 나타내는 플래그 선언, 3층 광인터럽트.

int button_1_state; // 버튼의 상태를 나타내는 플래그 선언, 1층 버튼
int button_2_state; // 버튼의 상태를 나타내는 플래그 선언, 2층 버튼
int button_3_state; // 버튼의 상태를 나타내는 플래그 선언, 3층 버튼

int lcdState; // LCD의 상태를 나타내는 플래그 선언
int doorState; // 서보 모터의 상태를 나타내는 플래그 선언(문)
int buzzerState; // 버저의 상태를 나타내는 플래그 선언
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) { // 각각 3개의 광인터럽트와 버튼을 외부 인터럽트로 사용, 만약 신호
    if(GPIO_Pin == GPIO_PIN_8) { // c8에 연결되어있는 광인터럽트 1번(1층)으로 인터럽트 신호가 들어오면
        photo_int_1_state = 1; // 광인터럽트 1 플래그를 세우고
        photo_int_2_state = 0; // 다른 광인터럽트 플래그를 초기화.
        photo_int_3_state = 0;
        buzzerState = 1; // 버저 플래그를 세움.
    }
    else if(GPIO_Pin == GPIO_PIN_6) { // c6에 연결되어있는 광인터럽트 2번(2층)으로 인터럽트 신호가 들어오면
```

CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[RTL] 지하탈출 넘버원



지하탈출 넘버원

[목 적]

Basys3 보드 기반 지하주차장 재난 대응 시스템 구현.

[목 표]

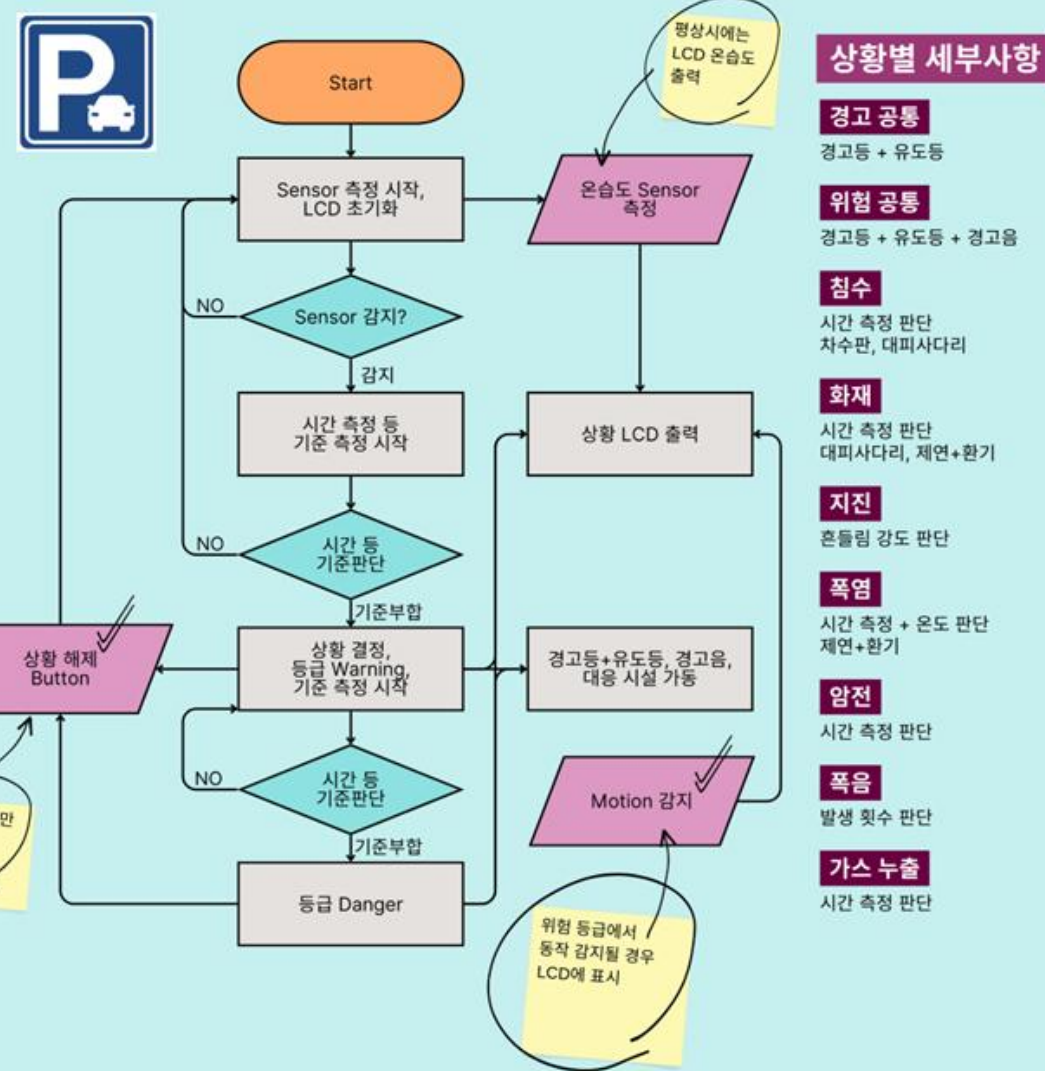
1. 각종 센서를 통해 침수, 화재, 가스누출, 지진 등의 상황을 감지
2. 감지된 상황을 기반으로 차수막, 자동 사다리 등 자동으로 필요한 조치를 취함
3. FSM 방식을 활용해 각종 상황을 효과적으로 제어

• 개발 환경



[RTL] 지하탈출 넘버원 - 수행 내역

플로우 차트



핵심 도전과제 & 해결

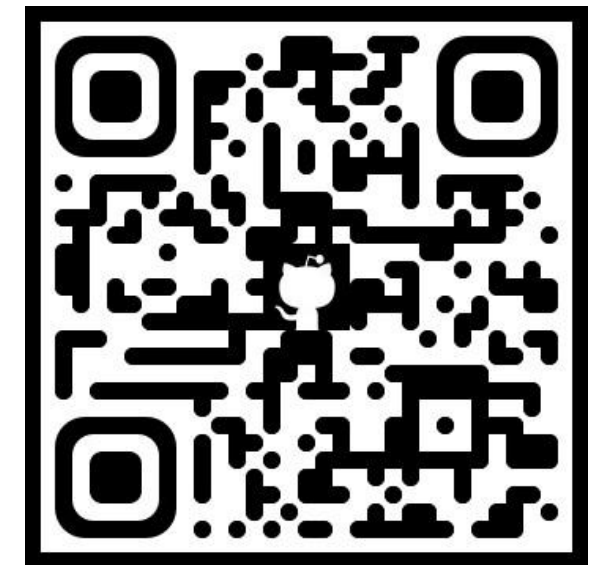
- 텍스트 LCD 문자열 전송

텍스트 LCD에 하나의 문자를 보내는 것은 비교적 간단하지만, 문자열을 화면 Refresh 한번에 띄우는 것에 어려움이 있었음.

문자열을 16(LCD 한 줄)x8(ASCII 코드 크기)비트 배열로
구성한 후 1바이트씩 16번을 쪼개서 보내는 것으로 해결.
엄밀히 말해 한 클럭에 동시에 문자열이 들어가는 것은
아니지만, 인간이 인지할 수 없는 속도로 데이터가 전송되기
때문에 사실상 원하는 대로 구현을 할 수 있었음.

위의 경험으로 기본적인 RTL 설계와 베릴로그 언어 능력을 증진할 수 있었음.

CODE



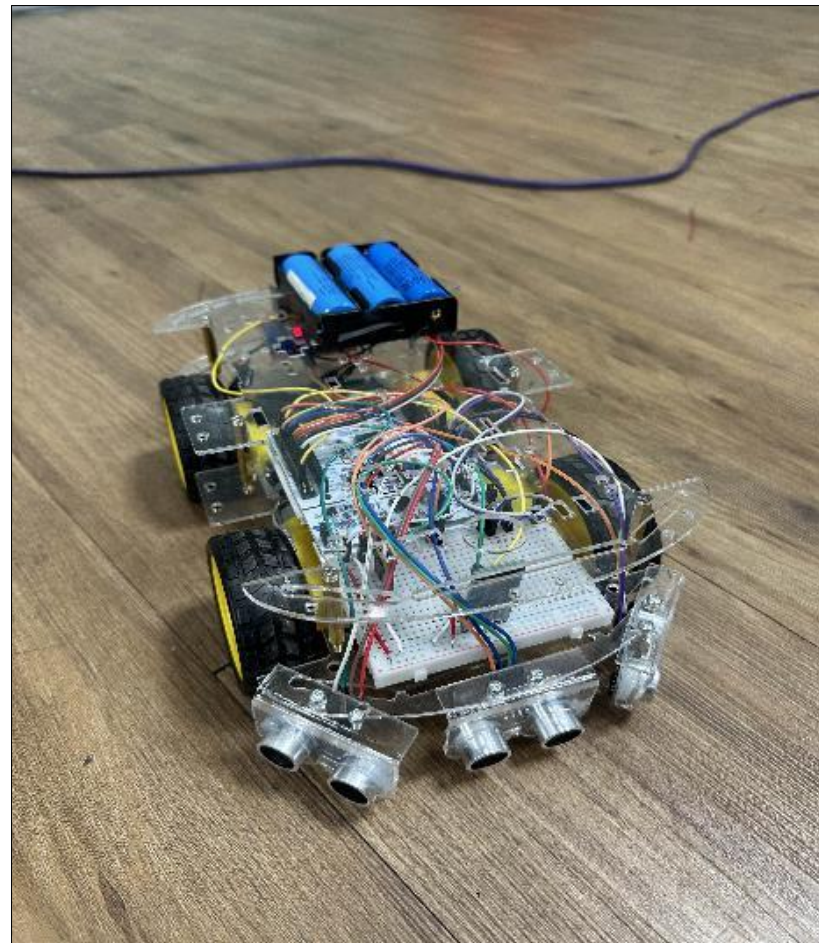
클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[ARM] 자율주행 RC카



자율주행 RC카

[목 적]

STM32 보드 기반 자율주행 RC카 및 무선 컨트롤러 구현

[목 표]

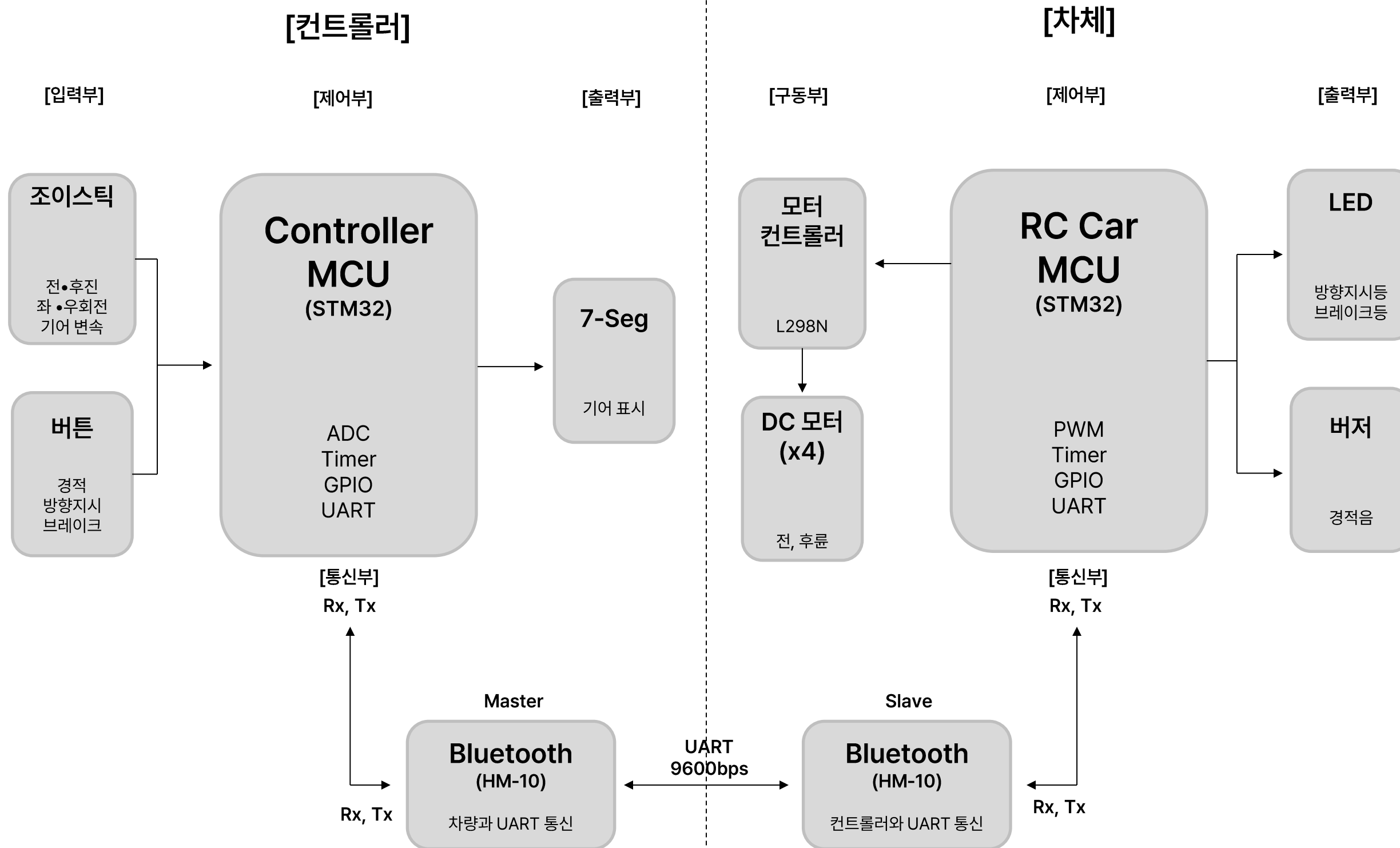
1. 초음파 센서를 통해 장애물을 감지하면 자동으로 길을 찾는 알고리즘 구현
2. 블루투스 모듈을 사용해 무선으로 RC카를 제어
3. 외부 전원을 이용한 완전 무선 시스템 구현

• 개발 환경

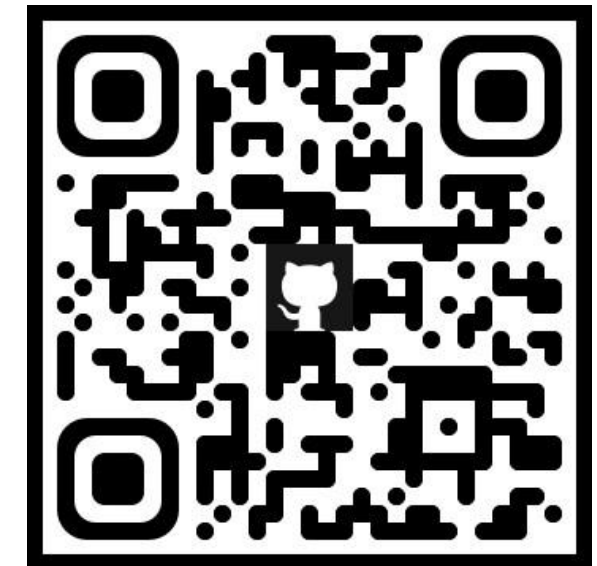


[ARM] 자율주행 RC카 - 수행 내역

시스템 아키텍처 다이어그램



CODE



클릭 시 링크 이동

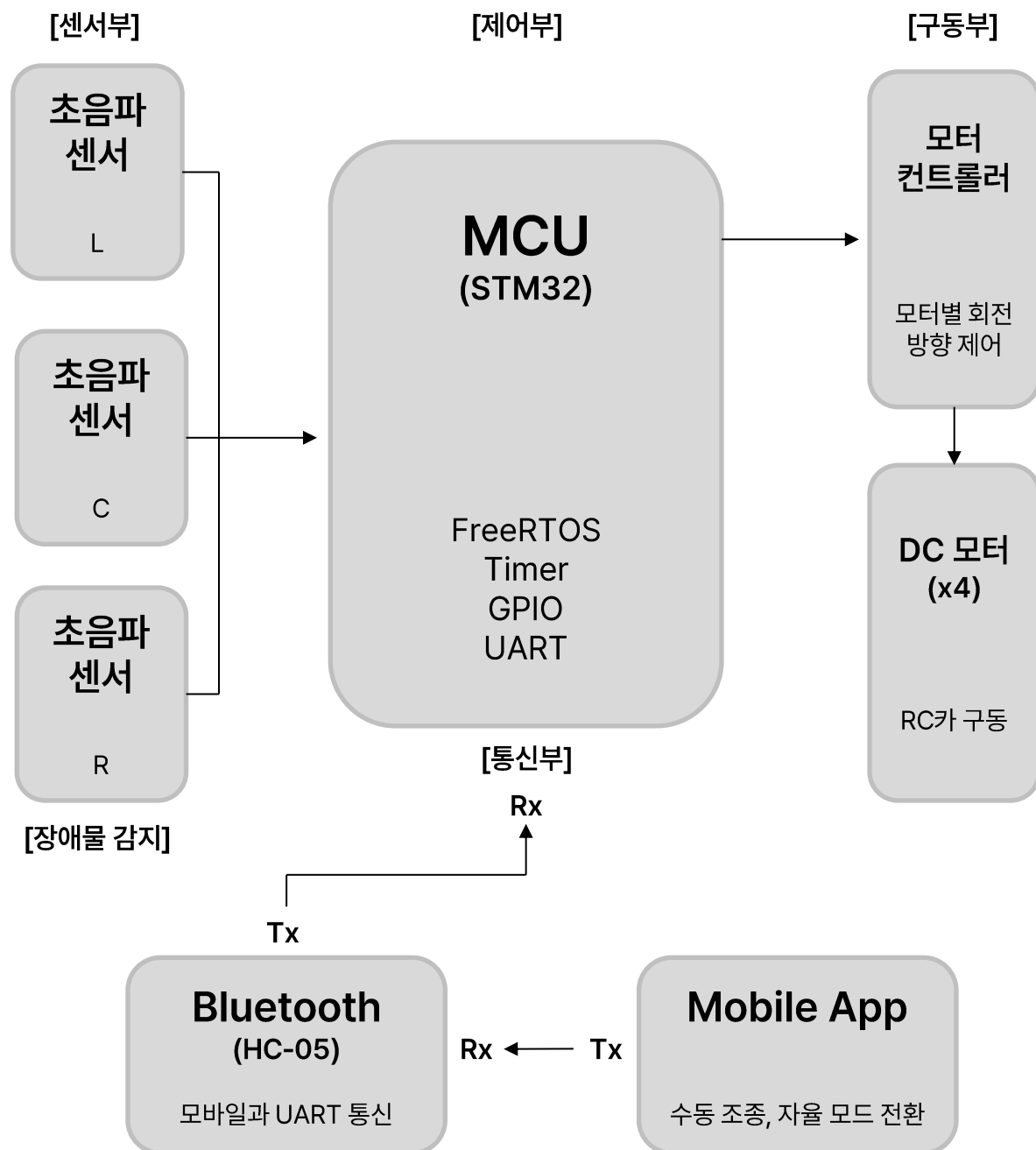
시연 영상



클릭 시 링크 이동

[ARM] 자율주행 RC카 - 수행 내역

시스템 아키텍처 다이어그램



핵심 도전과제 & 해결

- RTOS를 이용한 Non-Blocking 코드 모사

이전 엘리베이터 프로젝트 진행 시 동시에 제어해야 하는 Peripheral이 많고 Timing을 맞춰 동기해야 하는 경우에 일반적인 코드 구조로는 완벽한 시스템을 만들기 힘들다는 것을 체감.

마찬가지로 센서 값에 따른 모터 구동 변화를 동시다발적으로 제어해야 하는 이번 프로젝트에서는 FreeRTOS를 도입해 Non-Blocking 코드를 모사함.

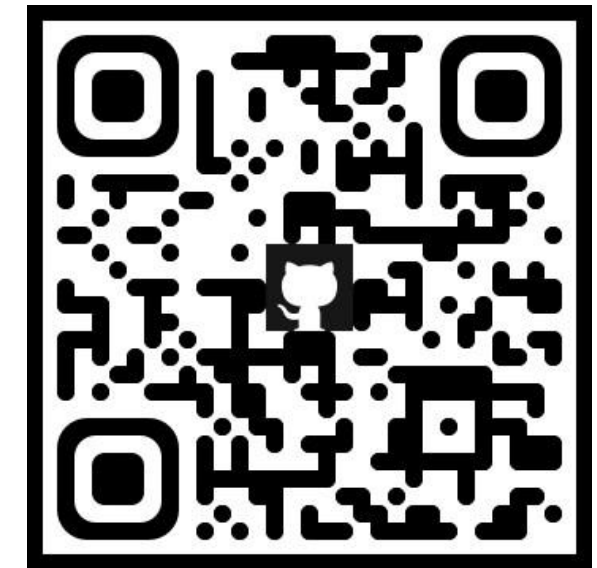
위의 경험으로 다중 태스크 환경에서 RTOS의 필요성 이해 및 RTOS 구현 능력을 습득함.

- 블루투스 4.0 양방향 통신

이 프로젝트에서 사용한 HM-10 모듈의 설정을 위한 AT 커맨드가 인터넷에 있는 정보와 많이 달라 모듈끼리 페어링 등 통신이 제대로 되지 않는 문제 발생.

데이터시트를 확인해 보면 모듈별 펌웨어에 따라 커맨드가 상이한 부분이 있어 인터넷의 정보는 배제하고 데이터시트를 정독하고 따라가 해결.

CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[IoT] 입석 길잡이

Train Route

Randomize Route

Current Station: Cheonan

Seat 1

Seat 2

Seat 3

Seat 4

Reservation and availability

Station	Seat 1	Seat 2	Seat 3	Seat 4
Cheonan	Seated: 0, Sensor: 0	Seated: 1, Sensor: 0	Seated: 0, Sensor: 0	Seated: 0, Sensor: 0
Cheonan	Seated: 0, Sensor: 0	Seated: 1, Sensor: 0	Seated: 0, Sensor: 0	Seated: 0, Sensor: 0

입석 길잡이

[목 적]
기차 입석 이용객을 위한 빈 자리를 찾아주는 IoT 프로젝트

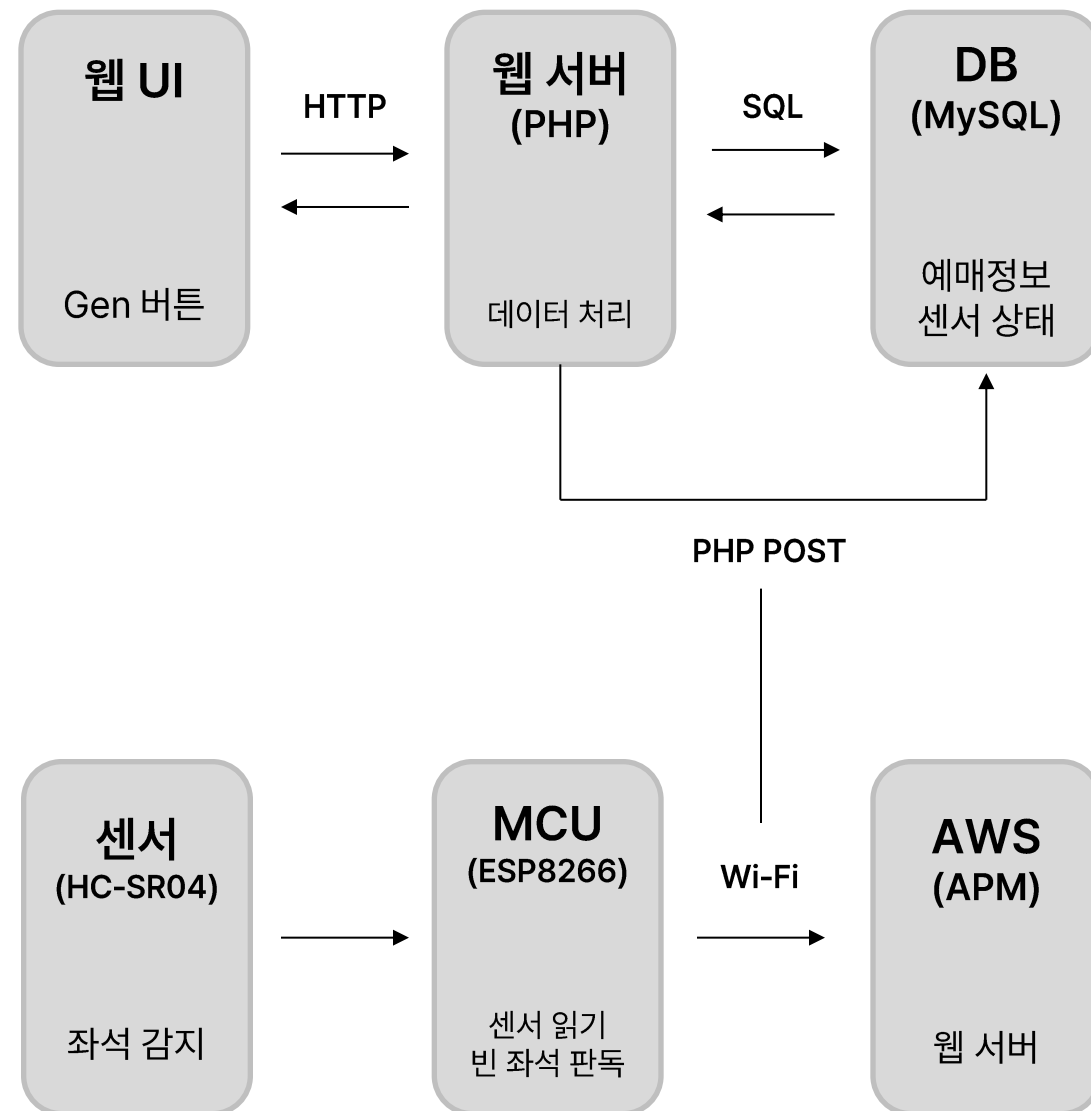
- [목 표]**
- 1. 아마존 AWS 서버를 대여해 EC2 서비스로 APM 서버를 직접 구축
 - 2. 센서를 통해 좌석이 물리적으로 점유되었는지 판단 후 서버 DB값과 대조
 - 3. 빈 좌석을 확인할 수 있는 웹 생성

● 개발 환경



[IoT] 입석 길잡이 - 수행 내역

시스템 아키텍처 다이어그램



핵심 도전과제 & 해결

- AmazonLinux2에 대한 정보 접근성 낮음

보통 서버를 구축 시 RedHat 혹은 CentOS 계열 리눅스를 많이 사용하는데, AL2는 독자적인 OS여서 RH이나 CentOS 등과 서버 구축 과정이 다른 부분이 많음.

해외 포럼 및 아마존 공식 문서들을 살펴보고, 에러 로그를 토대로 차근차근 디버깅해 성공적으로 구축할 수 있었음.

위의 경험으로 정보를 찾는 능력과 리눅스 사용 능력을 키울 수 있었음.

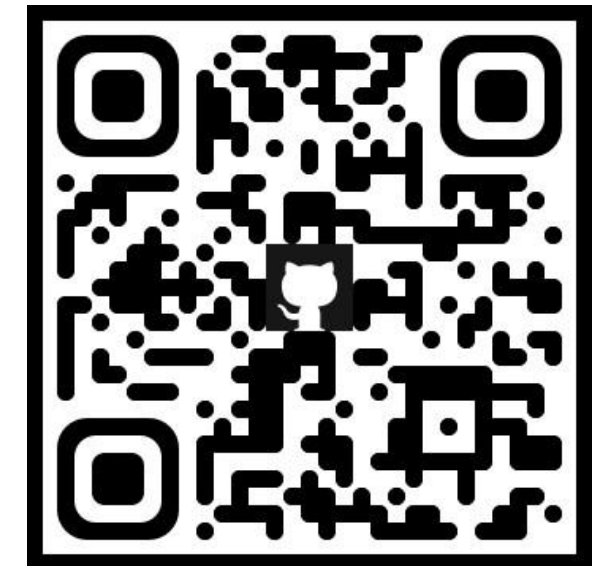
- 웹 새로고침 시 데이터 적용 오류

웹에서 새로고침을 하면 마지막으로 전송한 데이터를 다시 보내거나, 서버 DB와 로컬 캐시를 대조해 한 쪽으로 데이터를 맞춰 줘 개발자가 의도한 방식으로 작동하지 않게 된다.

해당 문제를 PRG(Post-Redirect-Get) 패턴을 적용해 새로고침시 정상적으로 DB값을 불러올 수 있게 됨.

위의 경험으로

CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[AVR] 선풍기 프로젝트



선풍기 프로젝트

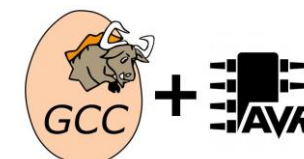
[목 적]

ATMEGA128a 보드 기반 미니 선풍기 제작

[목 표]

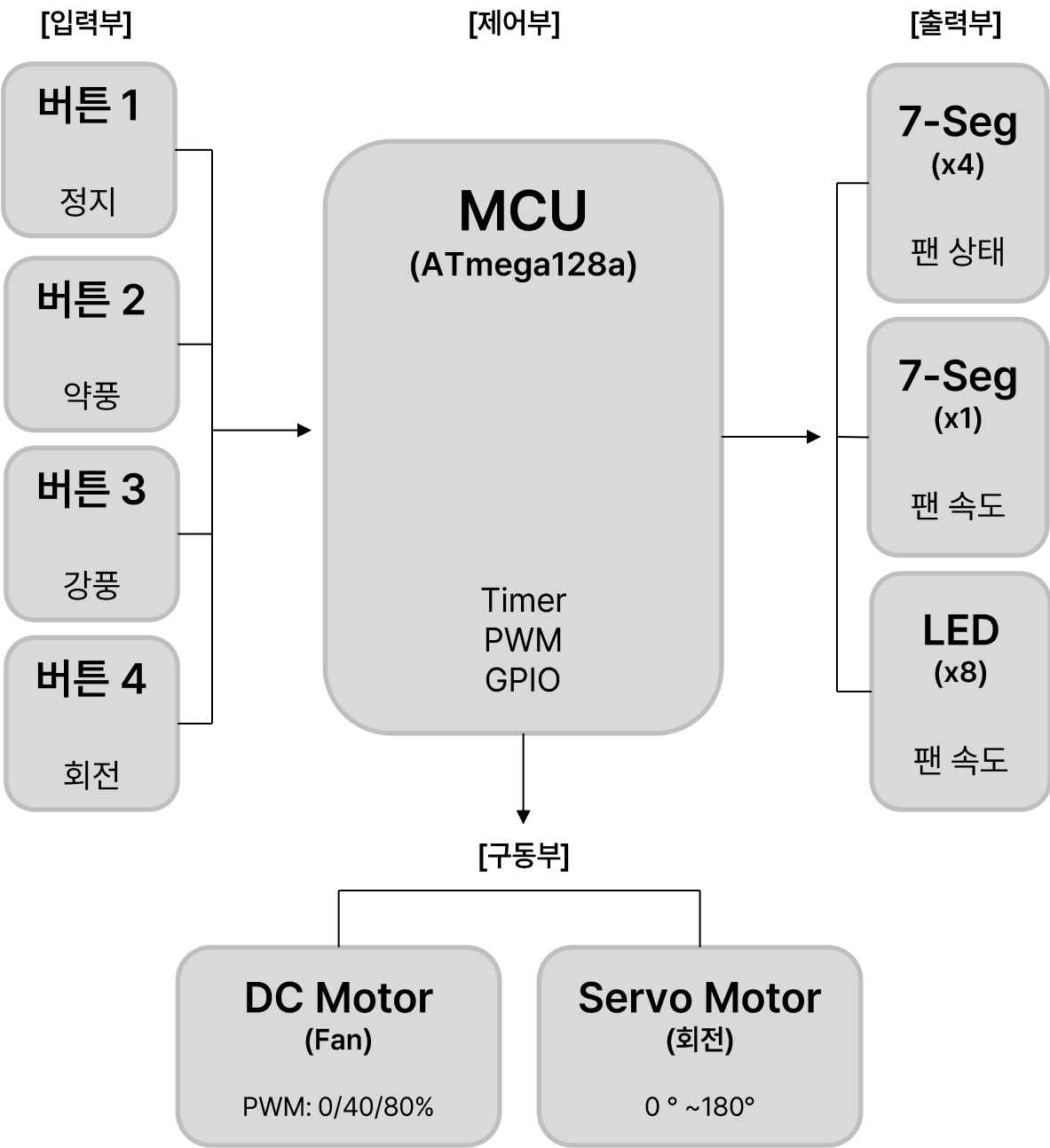
1. PWM을 통한 선풍기 바람 세기 조절 기능 구현
2. PWM으로 서보 모터를 제어해 회전 기능 구현
3. 각종 Peripheral을 통한 현재 상태를 알리는 UI 제공

• 개발 환경



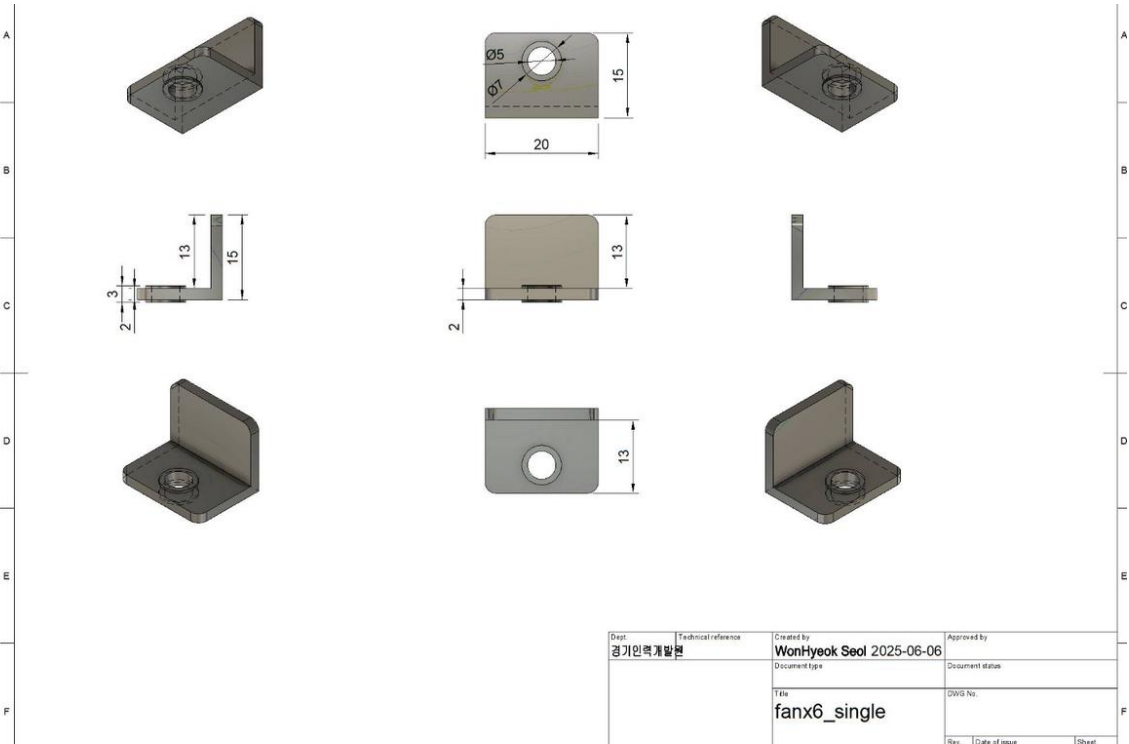
[AVR] 선풍기 프로젝트 – 수행 내역

시스템 아키텍처 다이어그램



핵심 도전과제 & 해결

- 서보모터와 DC모터 체결
- 선풍기의 회전 기능을 구현하기 위해 서보 모터와 DC모터를 체결해야 했는데, 서보 모터는 수평 평면에서 움직이고 DC 모터는 수직으로 세워야 해서 튼튼하게 체결할 면적이 부족.
- Fusion360 툴을 이용해 L자 브라켓을 모델링해 3D프린터로 출력 후 서보모터와 DC모터를 체결.



CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

[Analog] 베이스 이펙터 설계(WIP)

베이스 이펙터 설계

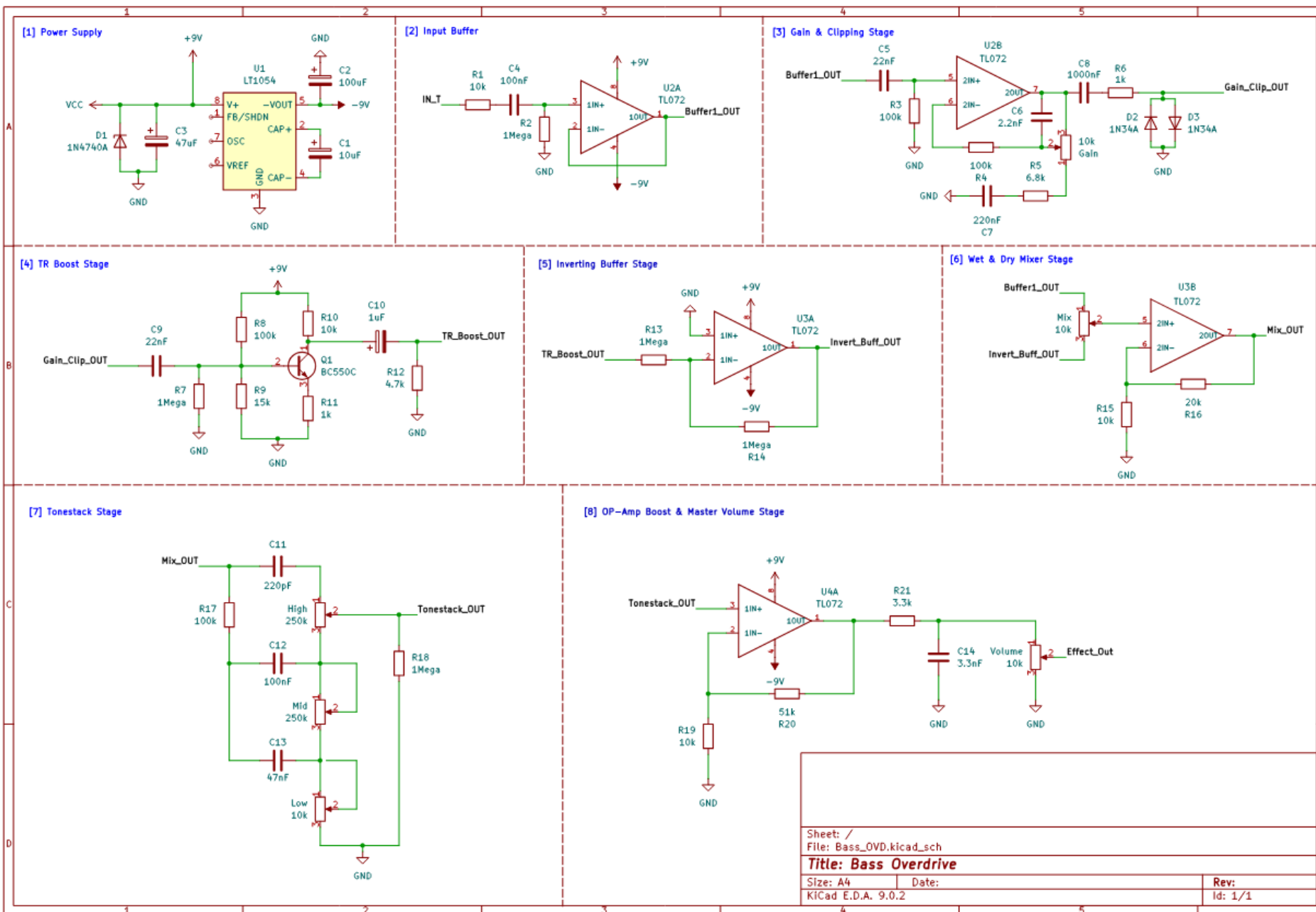
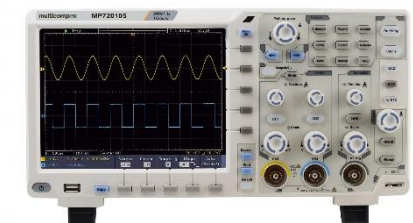
[목 적]

베이스 기타를 위한 Overdrive 이펙터 설계.

[목 표]

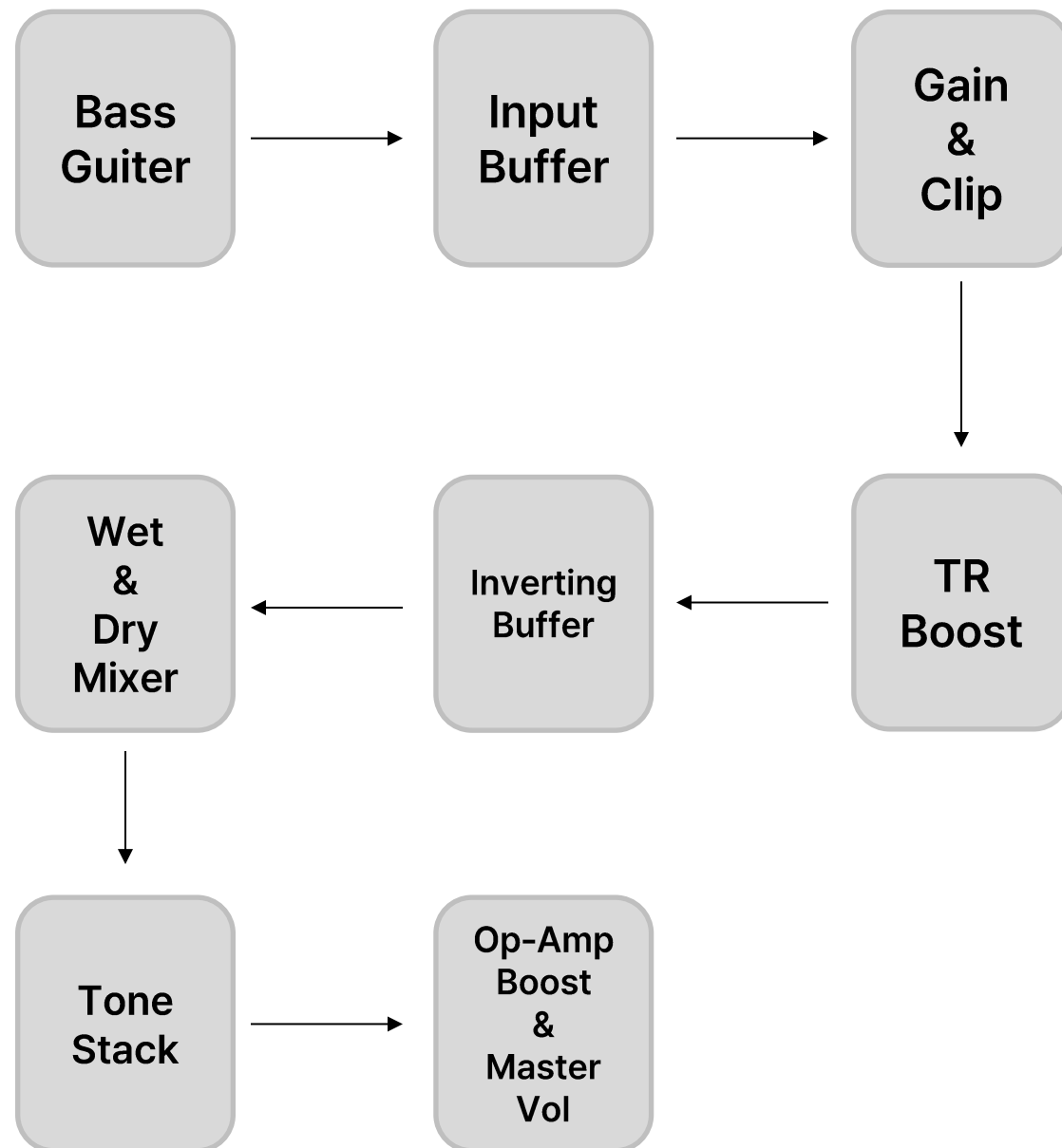
1. TR, OP-Amp, RC Filter 등을 사용한 아날로그 증폭 회로 구현
2. 베이스 기타의 주 음역대인 저역대 손실 최소화
3. 최대 Gain이 +40db정도가 되도록 설계

● 개발 환경



[Analog] 베이스 이펙터 설계(WIP) – 수행 내역

Signal Flow

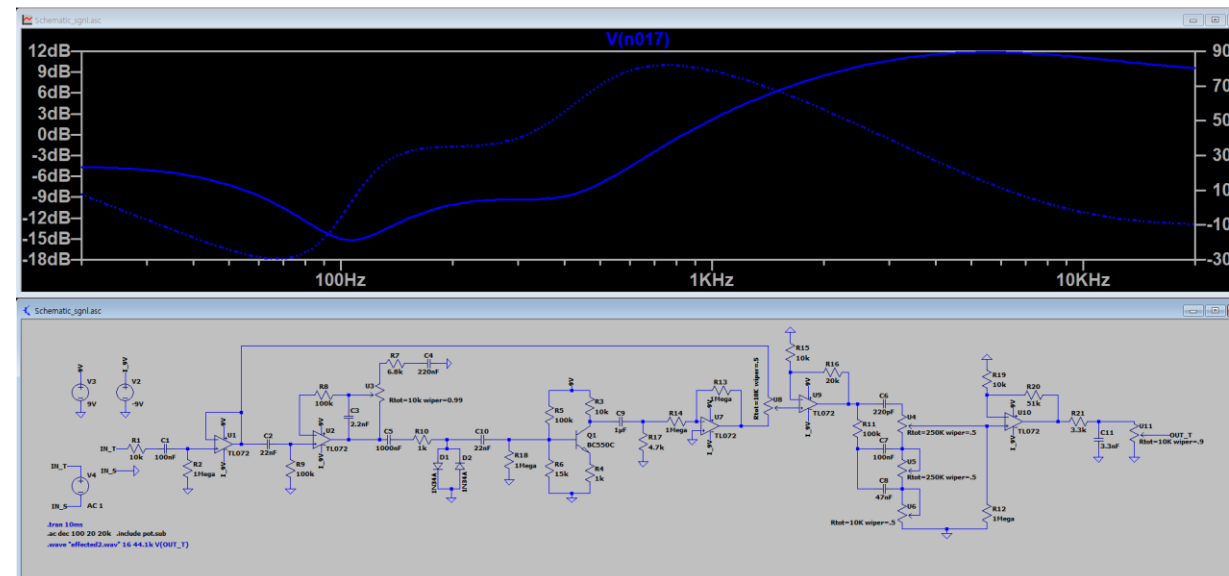


개선해야 할 문제점

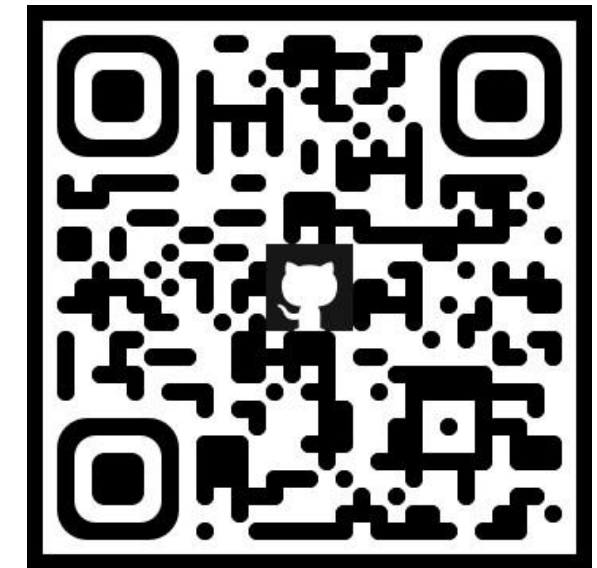
- Loading Effect 관련 문제

RC 필터를 이용해 주파수 대역을 조정할 때, 각 단을 분리해서 검증해 보면 원하는 대로 cutoff 혹은 bandwidth 주파수가 나오지만 신호단을 모두 합치고 검증해 보면 필터 대역이 이상하게 나오는 문제 발생.

신호 전, 후단의 저항 성분 때문에 Voltage divider 처럼 작용해 원하는 대역폭이 나오지 않는 Loading Effect 때문인 것으로 확인했지만, 이를 해결하기 위해선 각 단의 입/출력에 buffer를 달아 주어야 함. 이 프로젝트의 예상 전력 소모량보다 훨씬 큰 전류를 필요로 할 것 같아서 보류.



CODE



클릭 시 링크 이동

시연 영상



클릭 시 링크 이동

감사합니다.

설원혁

010-3311-6680

swh05284@gmail.com



Wormboiii