

# 스마트 꼬북팜

2025.8.30

TEAM 3

정용재(팀장), 김윤우, 김지훈, 유지승, 이지원

# 목 차

## 1 팀 구성

역할

개발 환경

## 2 배경 및 개요

프로젝트 배경

프로젝트 개요

## 3 프로젝트 구성

H/W 구성도

S/W 구성도

시연 영상

## 4 프로젝트 기술서

라인 트레이서

터틀 봇

서버 및 UI

## 5 마치며

라인 트레이서

질의 및 응답

# 팀 구성

# 팀 구성

**서버팀** 정용재

Qt App 개발, 서버 구축, Wi-Fi 무선 네트워크 구축

**라인 팀** 유지승, 김지훈

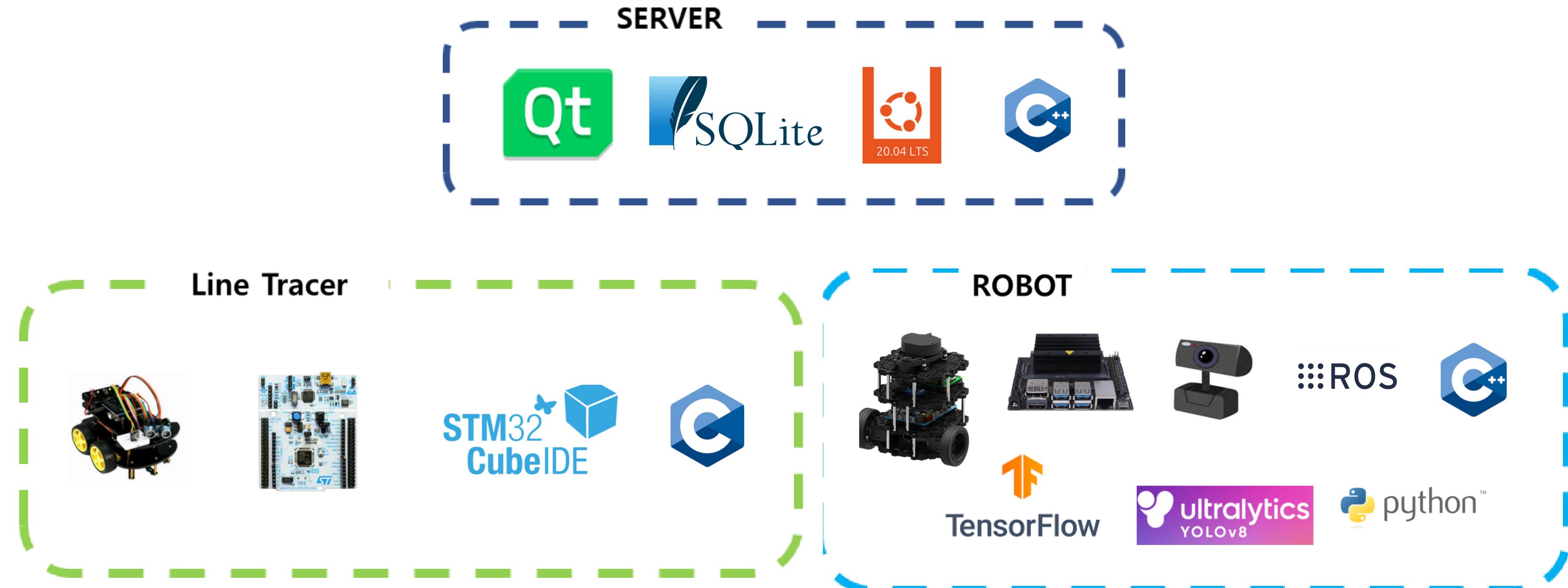
라인 트레이서, 센서 제어, 서보모터 제어, 모터 제어

**터틀 팀** 김윤우, 이지원

AI 모델링, Turtlebot 제어, ROS, Qt App 개발

# 프로젝트 개요 - 개발 환경

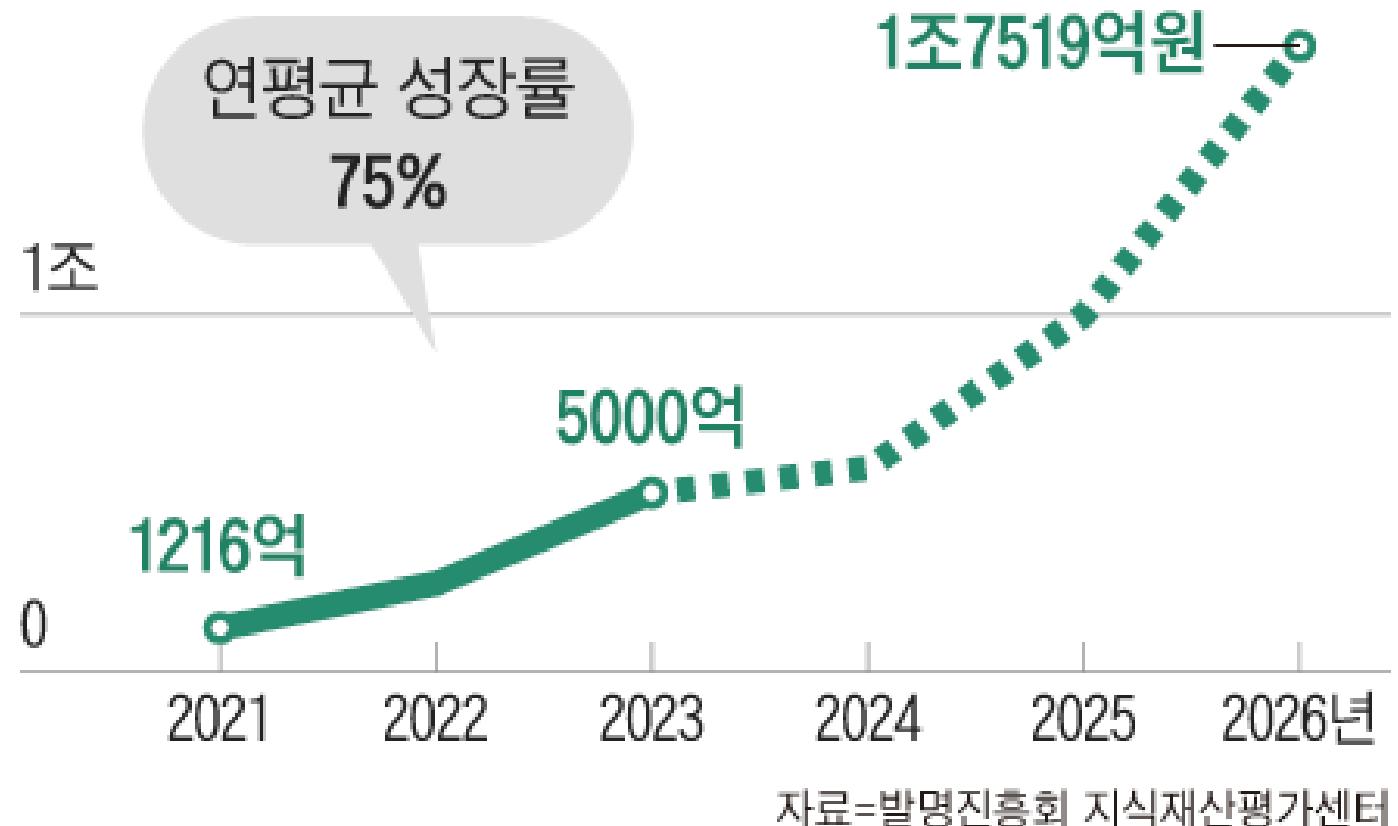
## ▶ 개발 환경



# **프로젝트 배경**

# 프로젝트 배경

## 국내 실내 농업 관련 시장 규모 및 전망



## 반려식물 상품 내놓은 기업

교원웰스 웰스팜·플로린

식물 재배기와 모종 정기구독 결합,  
다양한 채소·꽃 상품

그루우, 풀리프, 트리팜

AI 식물 진단, AR 식물 맞춤 제작,  
식물 관리 커머스 서비스

오늘의집, 컬리, 11번가

가상 정원·테라스에서 가상 농사  
지으면 실제 작물 배송

# 프로젝트 배경

## 기존 식물 재배기

### 주요 기능

- 인공광원 제공
- 온습도 관리
- 수분공급

### 단점

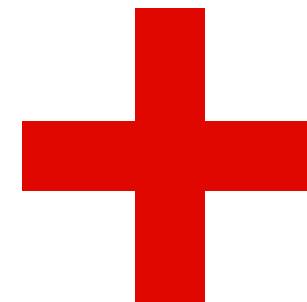
- 재배량 유연성 부족
- 작물 상태 직접 확인 필요(익음 정도, 질병 등)
- 작물 맞춤관리 불가능



# 프로젝트 개요 - 재배량 유연성 부족

- 기존 시스템: 고정된 화분에 물과 빛을 공급

- 물과 빛을 제공하는 인프라 고정
- 한정된 공간에 화분 배치
- 재배량 조정이 어려움



- 해결책: 움직이는 화분 시스템

- 화분이 스스로 물과 빛을 받으러 이동
  - 물과 빛의 공급 인프라 규모가 고정되지 않음
  - 공간 제약 해소

라인트레이서 + 화분



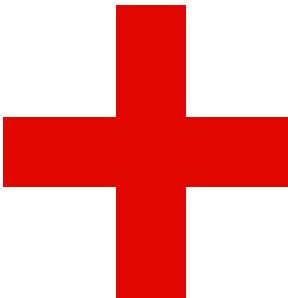
기능 및 효과

- 화분 자율 이동
- 인프라 조정 불필요
- 유연한 재배량
- 공간 최적화

# 프로젝트 개요 -작물 상태 직접 확인 필요

- 기존 시스템: 작물 상태 모니터링 기능 부재

- 익음 정도, 질병 등 확인 위해 노동력 소모



- 해결책: 실시간 화분 모니터링

- 실시간 모니터링 시스템

- 실시간모니터링으로 수동 확인 시간 절감.
    - 필요할 때만 알림을 받아 불필요한 작업 최소화
    - 수확 시기나 병충해 발생 시 빠른 대처로 작물 품질 향상

터틀봇 + AI



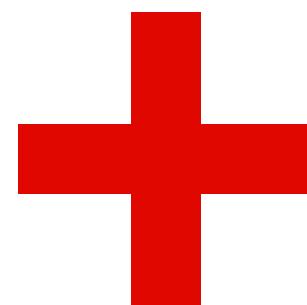
## 기능 및 효과

- 노동력 절감
- 효율적 관리
- 최적화된 작업

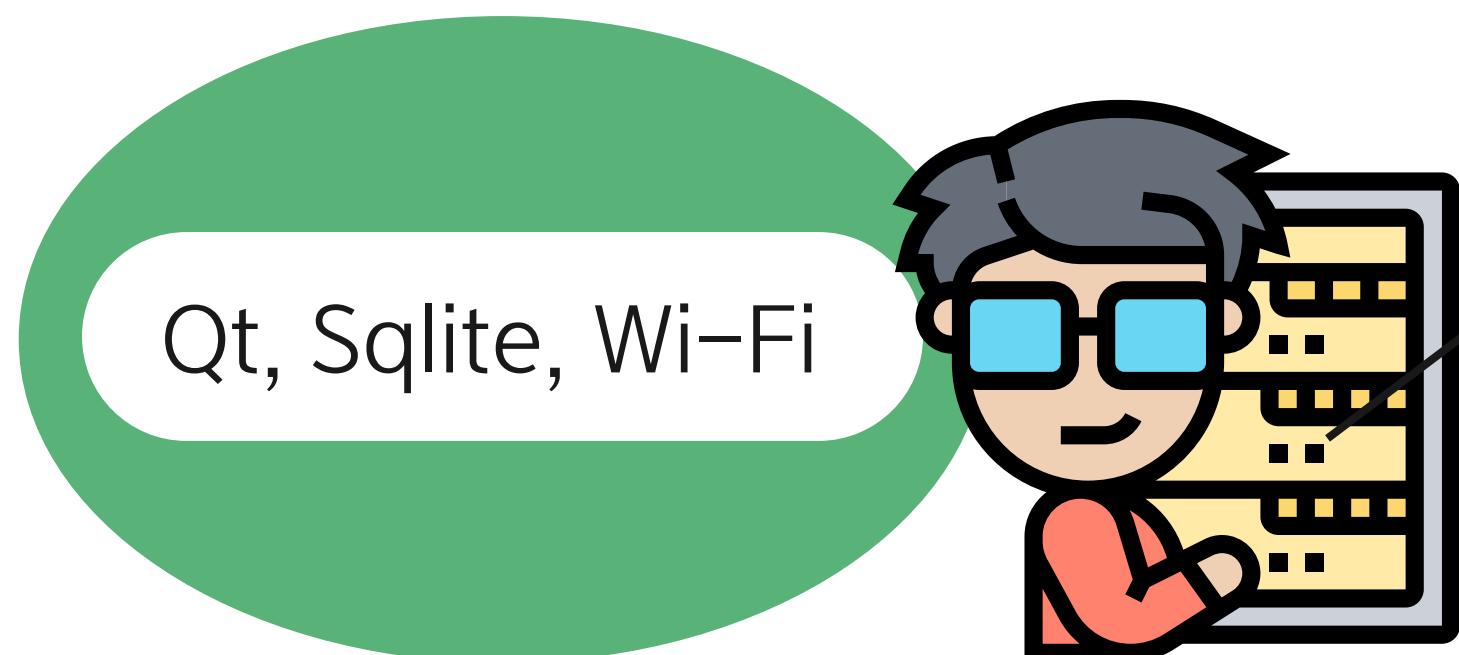
필요한 작업만 수행, 불 필요한 동선 최소화

# 프로젝트 개요 - 시스템 제어 및 모니터링

- 기존 시스템: 종합 제어, 데이터 수집 및 활용 기능 부족
  - 직접 식물을 확인 해야하는 불편함
  - 한 눈에 내 식물 상태의 통계량을 보기 힘듦



- 해결책: 종합 시스템 제어 및 모니터링
  - 터틀봇 전송: 지정된 좌표로 터틀봇을 이동시켜 원하는 화면을 볼 수 있다.
  - 작물의 통계 자료 수집: 작물별로 필요한 자원이 무엇인지 한 눈에 파악 할 수 있다.

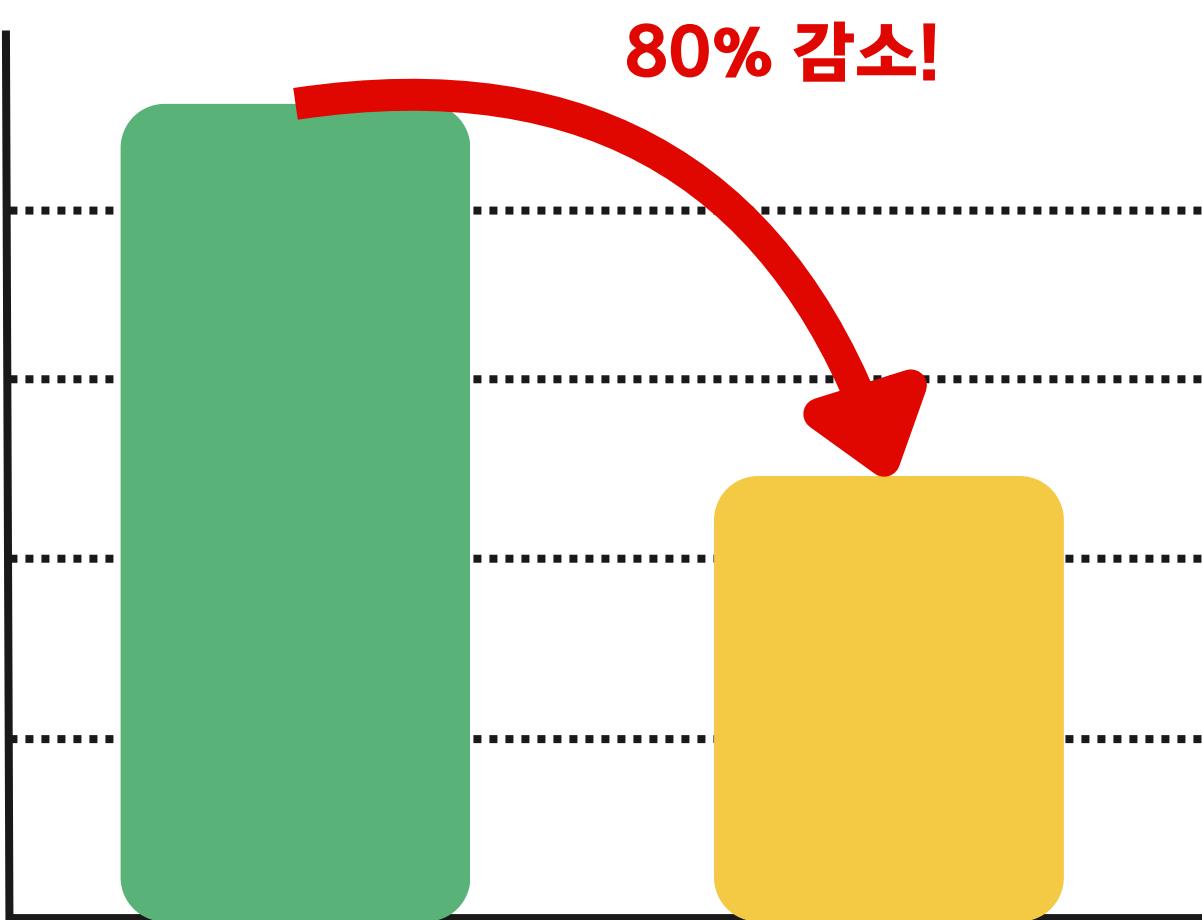


기능 및 효과

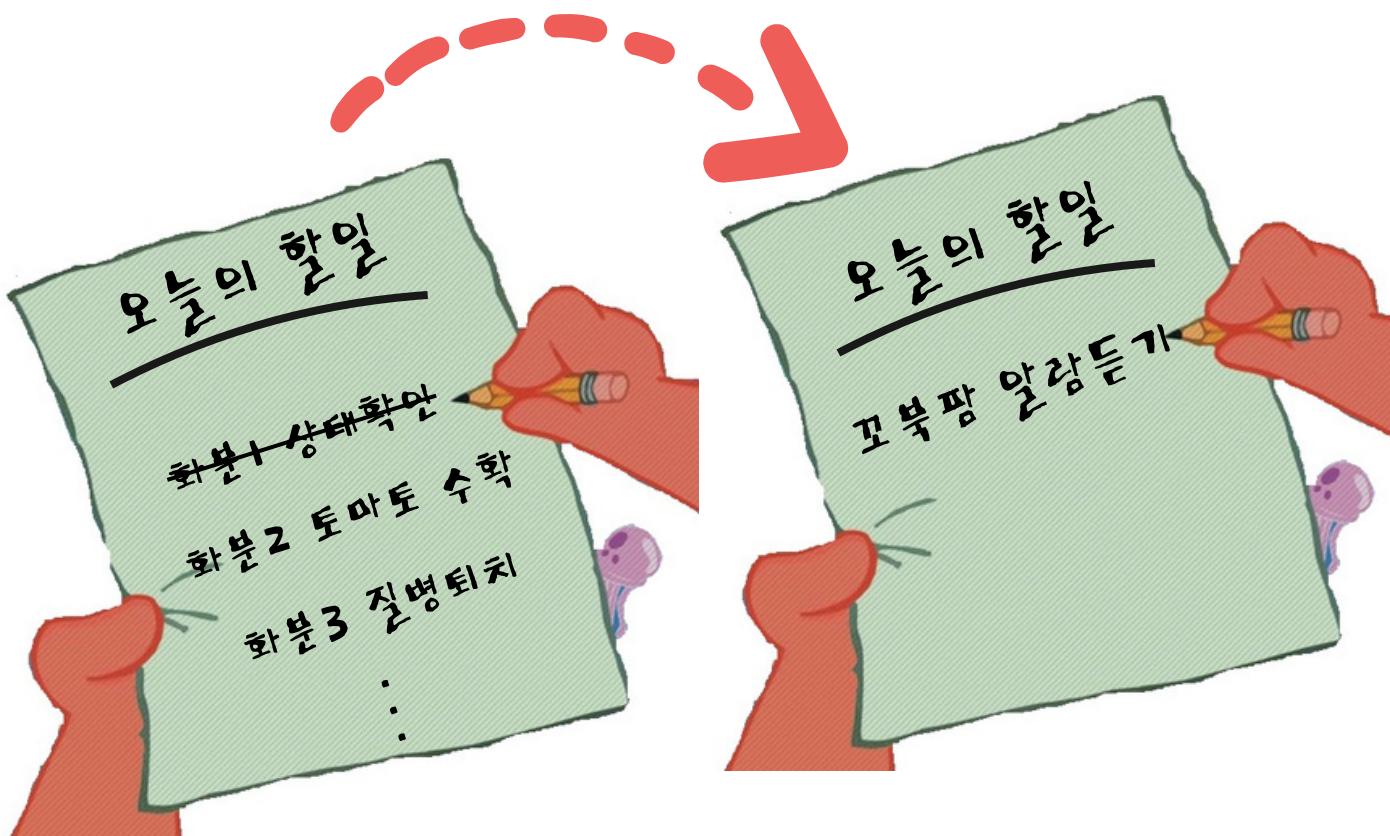
- 편리한 식물 관리
- 데이터 분석을 활용한 정밀 관리

# 프로젝트 개요 - 기대효과

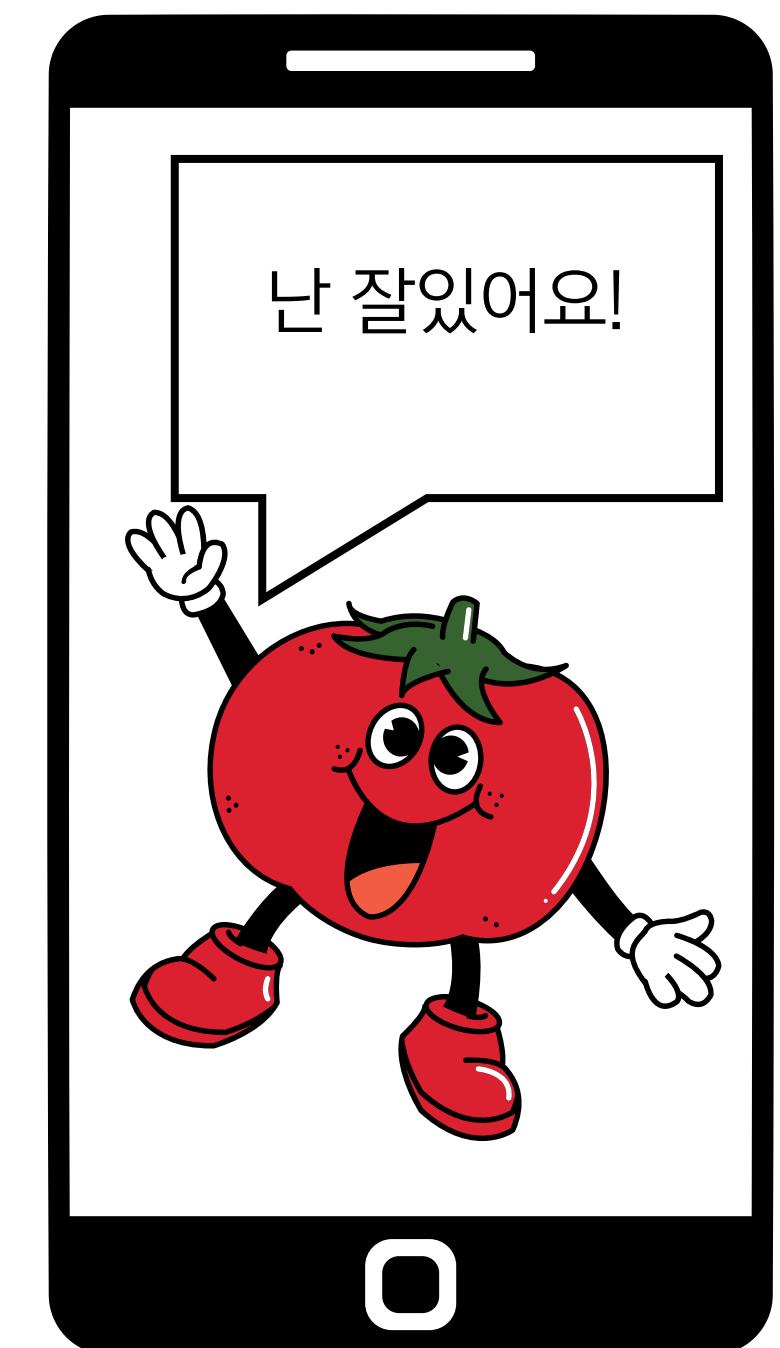
- 비용 절감



- 노동력 감소



- 반려식물 시장 개척



# **프로젝트 구성**

구성도

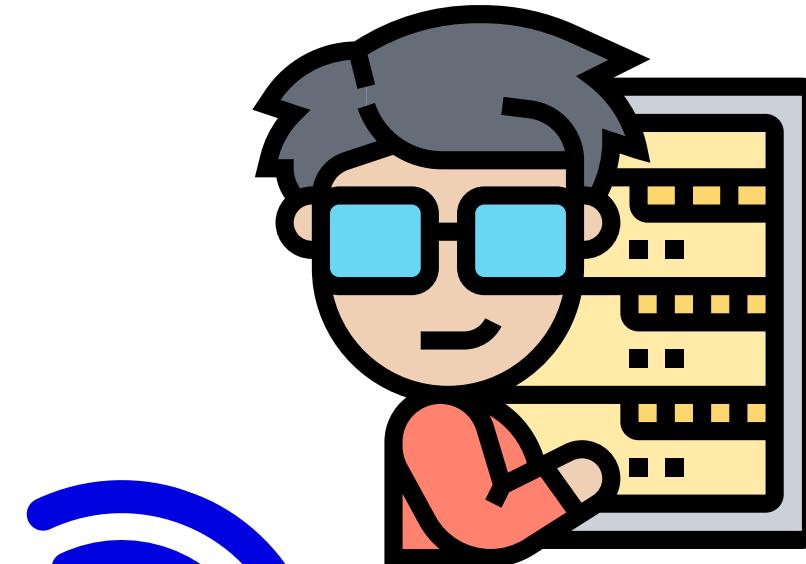
Server

HW구성도

작물 상태 송신

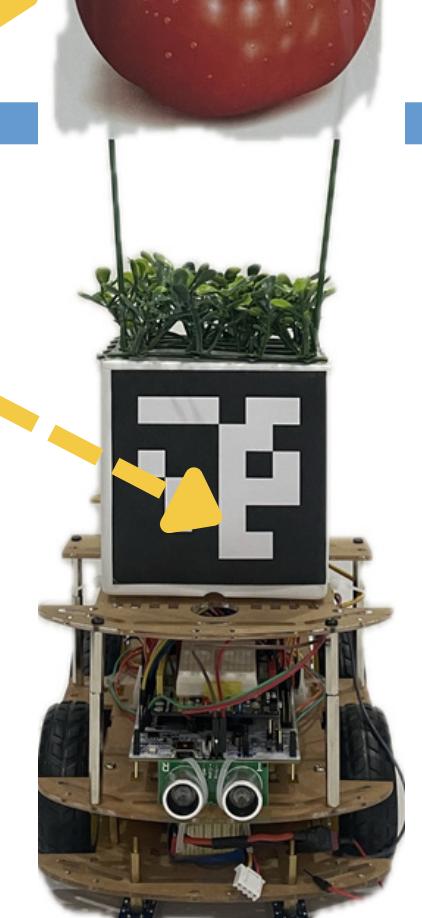


작물 상태 확인

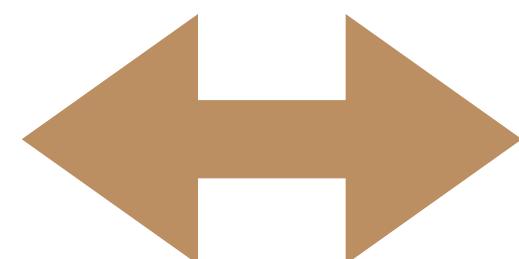


이동명령 수신

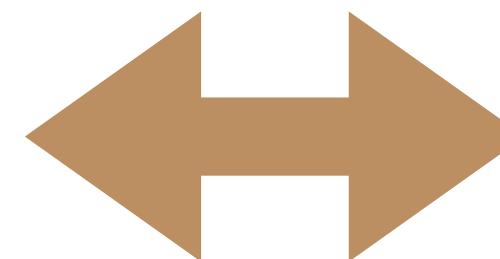
화분인식



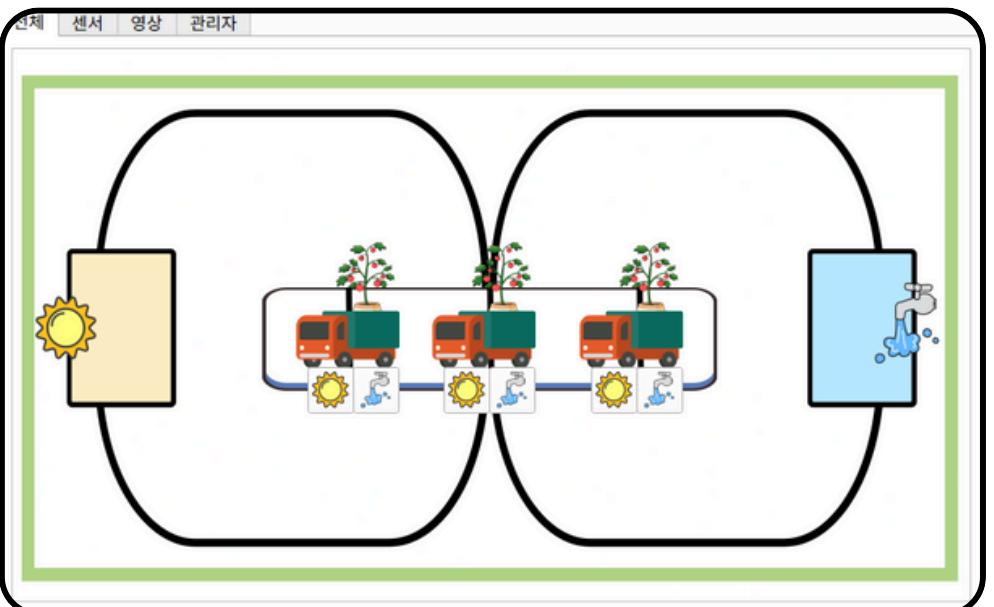
급수대



일조대



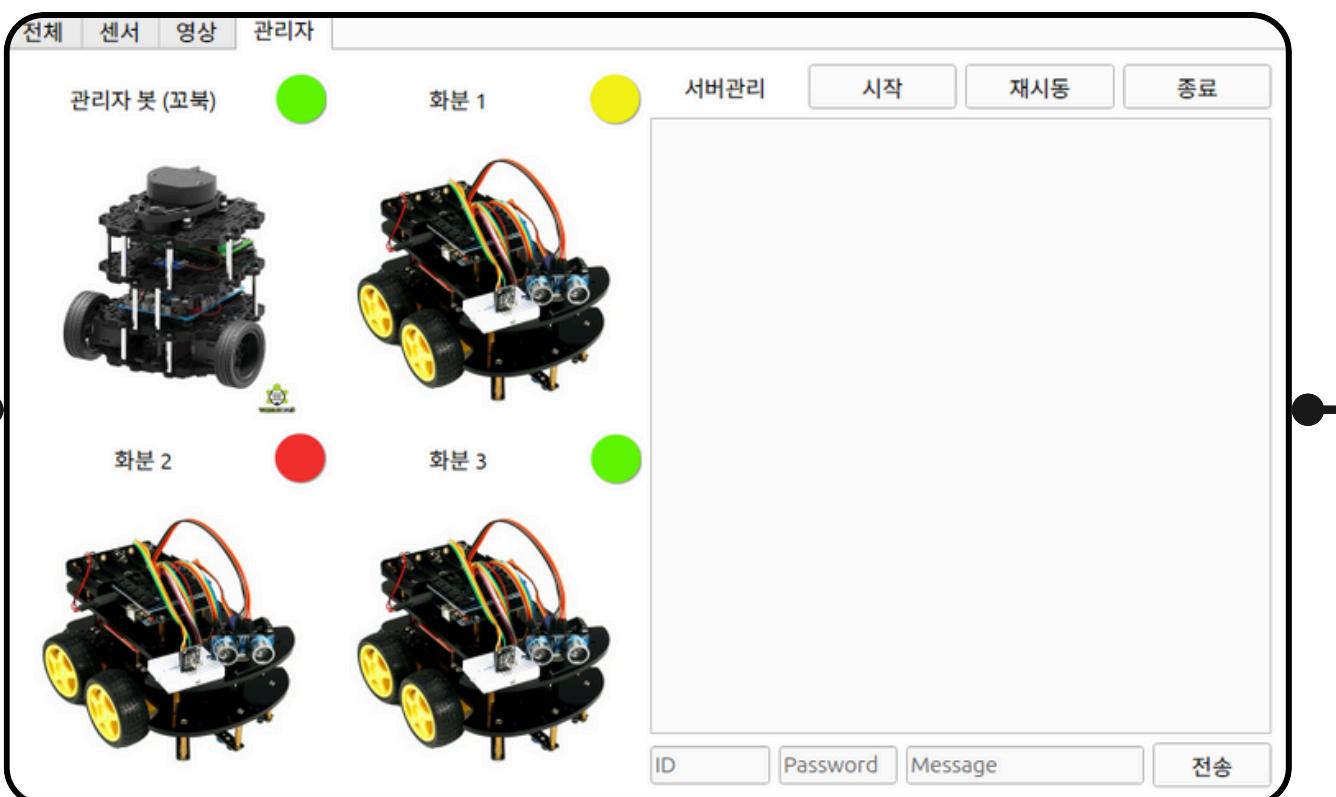
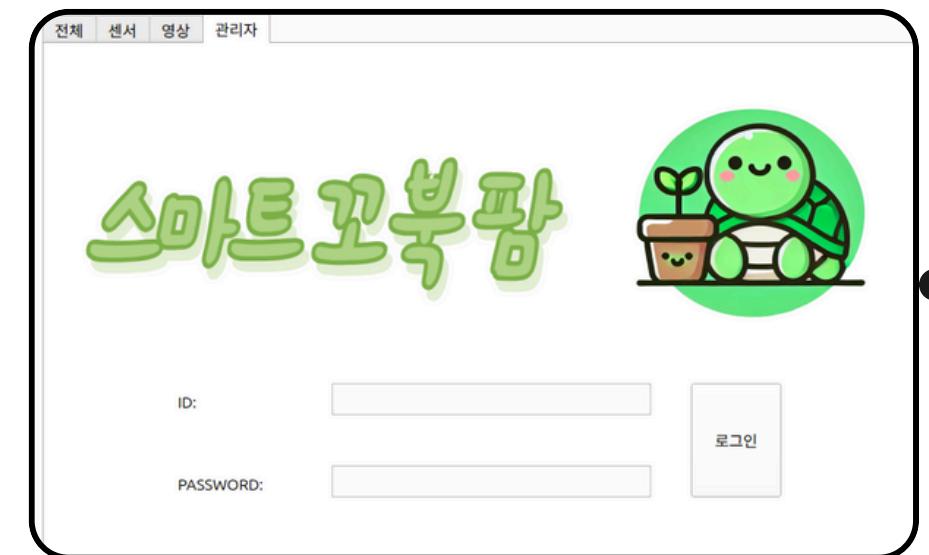
## • 화분위치 모니터링



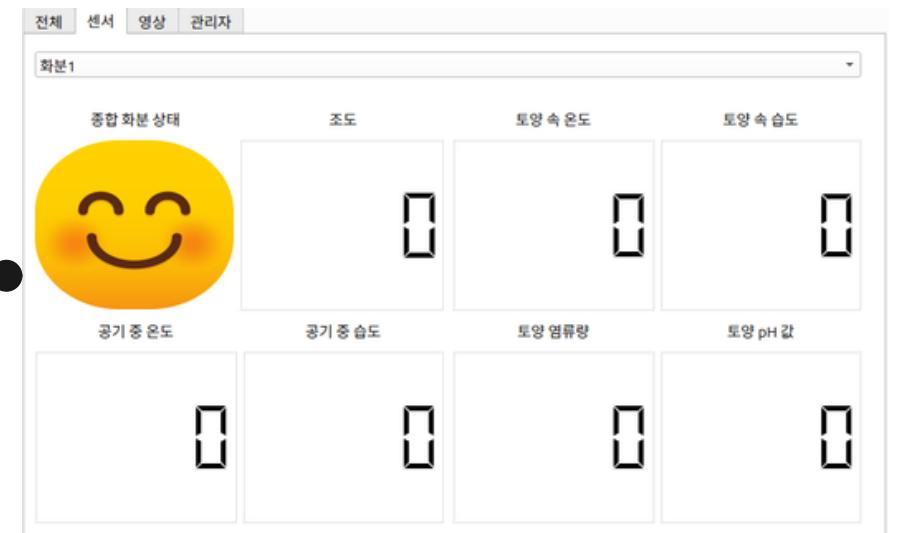
## SW구성도

- 메인(화분상태, 누적 데이터 확인)

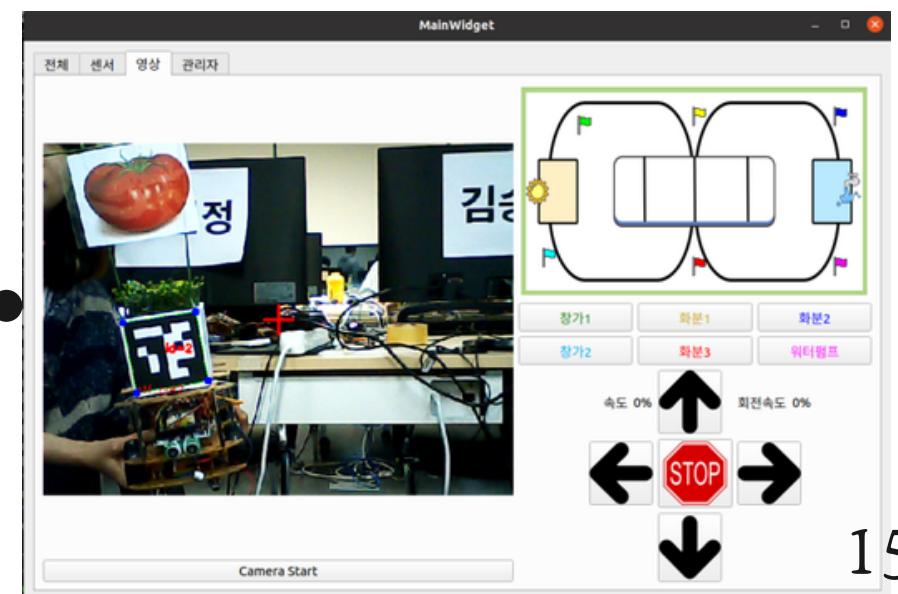
- 로그인 화면



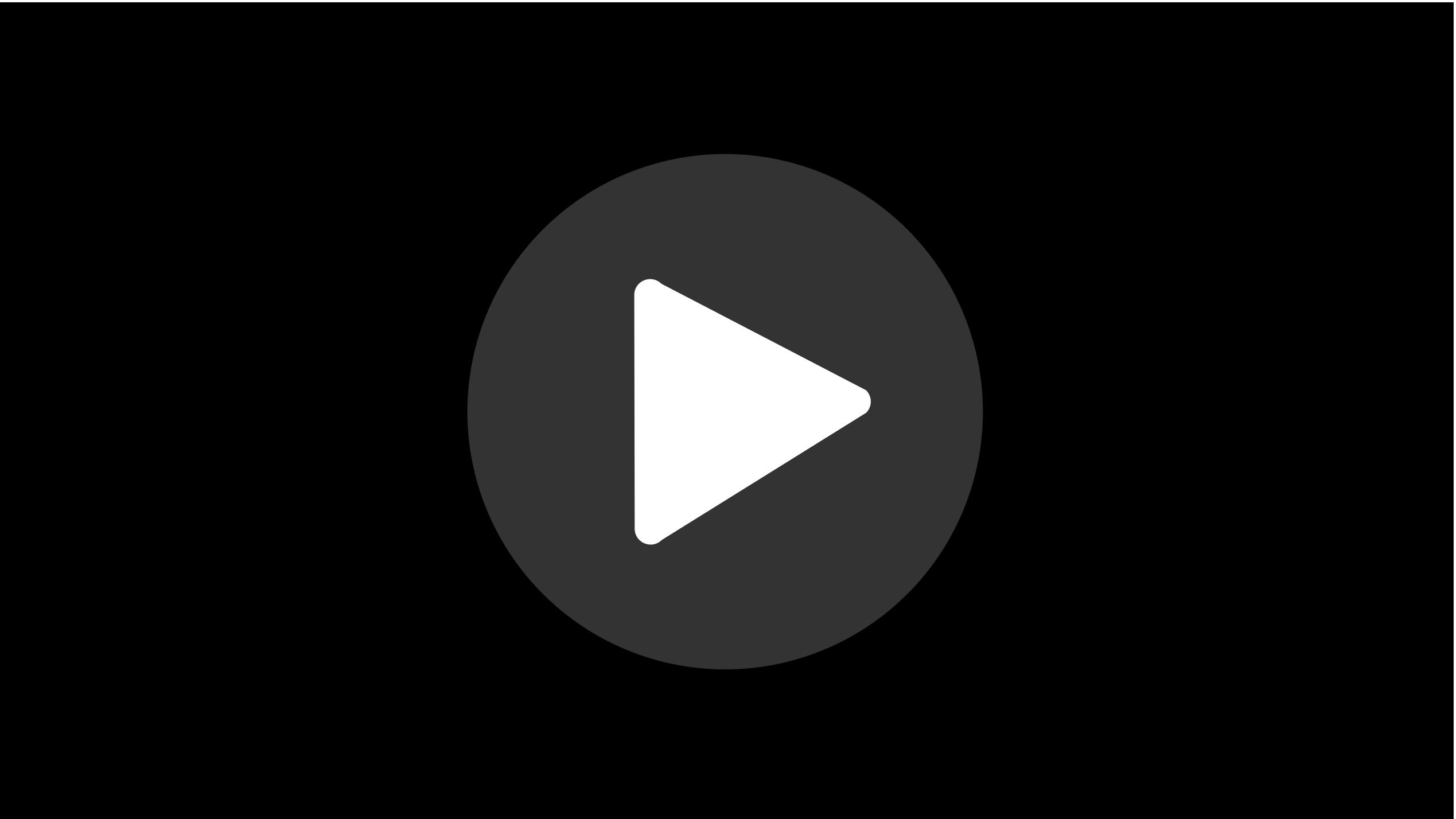
- 실시간 종합상태



- 관리자 영상확인

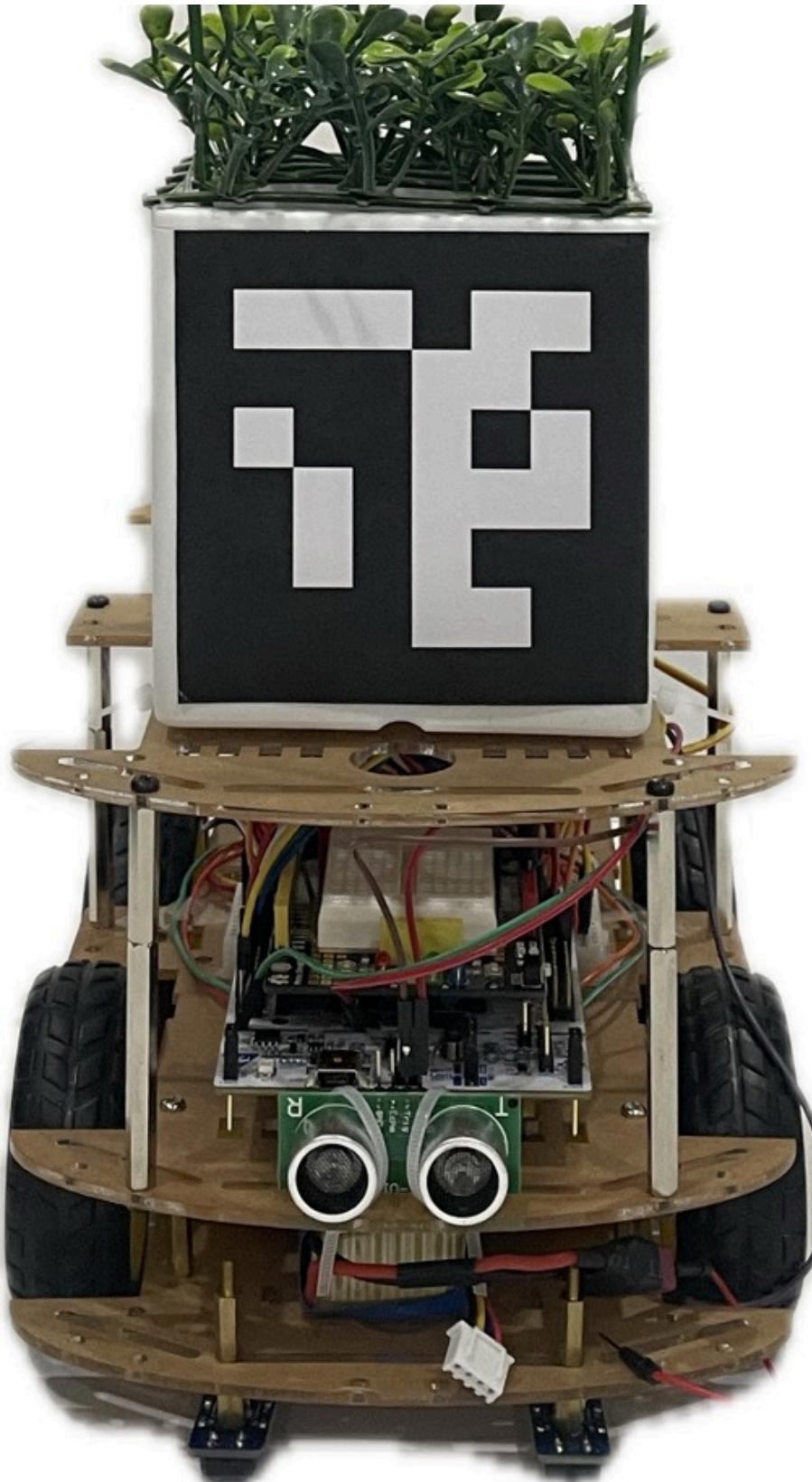


# 시연 영상



# **프로젝트 기술서**

# Linetracer



개요

기능

1. linetracer
2. 온습도 체크
3. 서버와 socket 통신

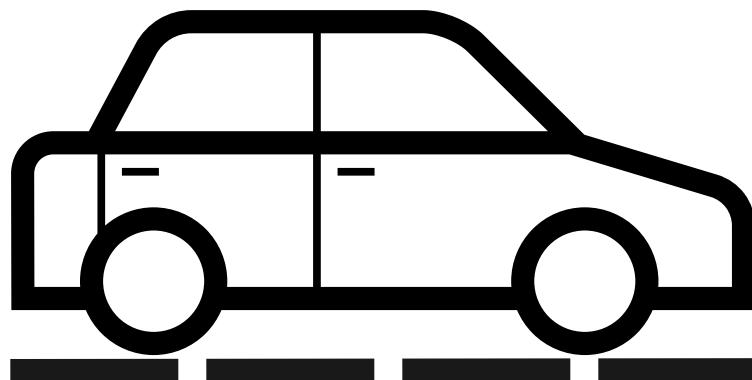
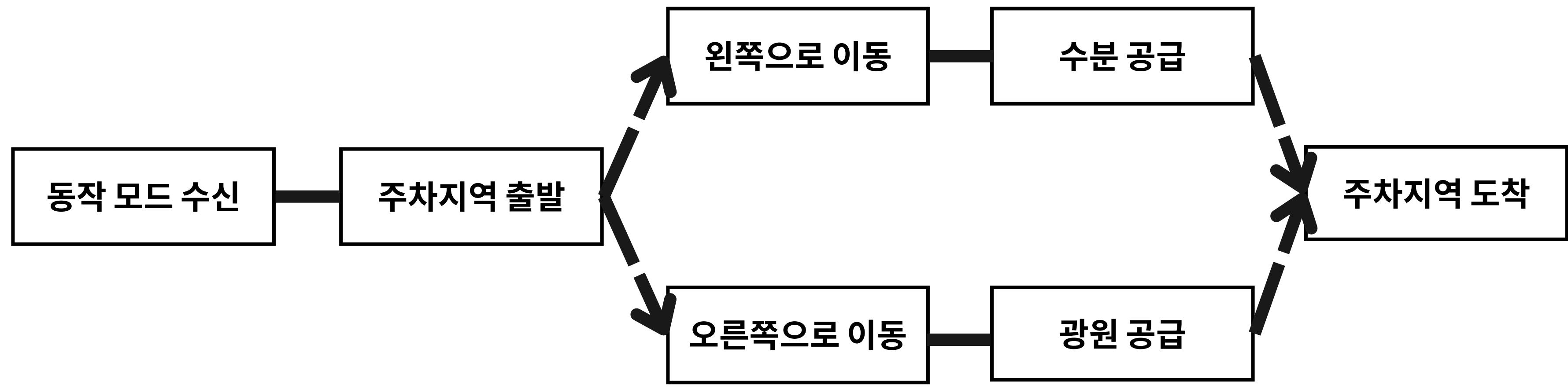
사용 기술

1. STM32 제어
2. PWM, Timer, UART

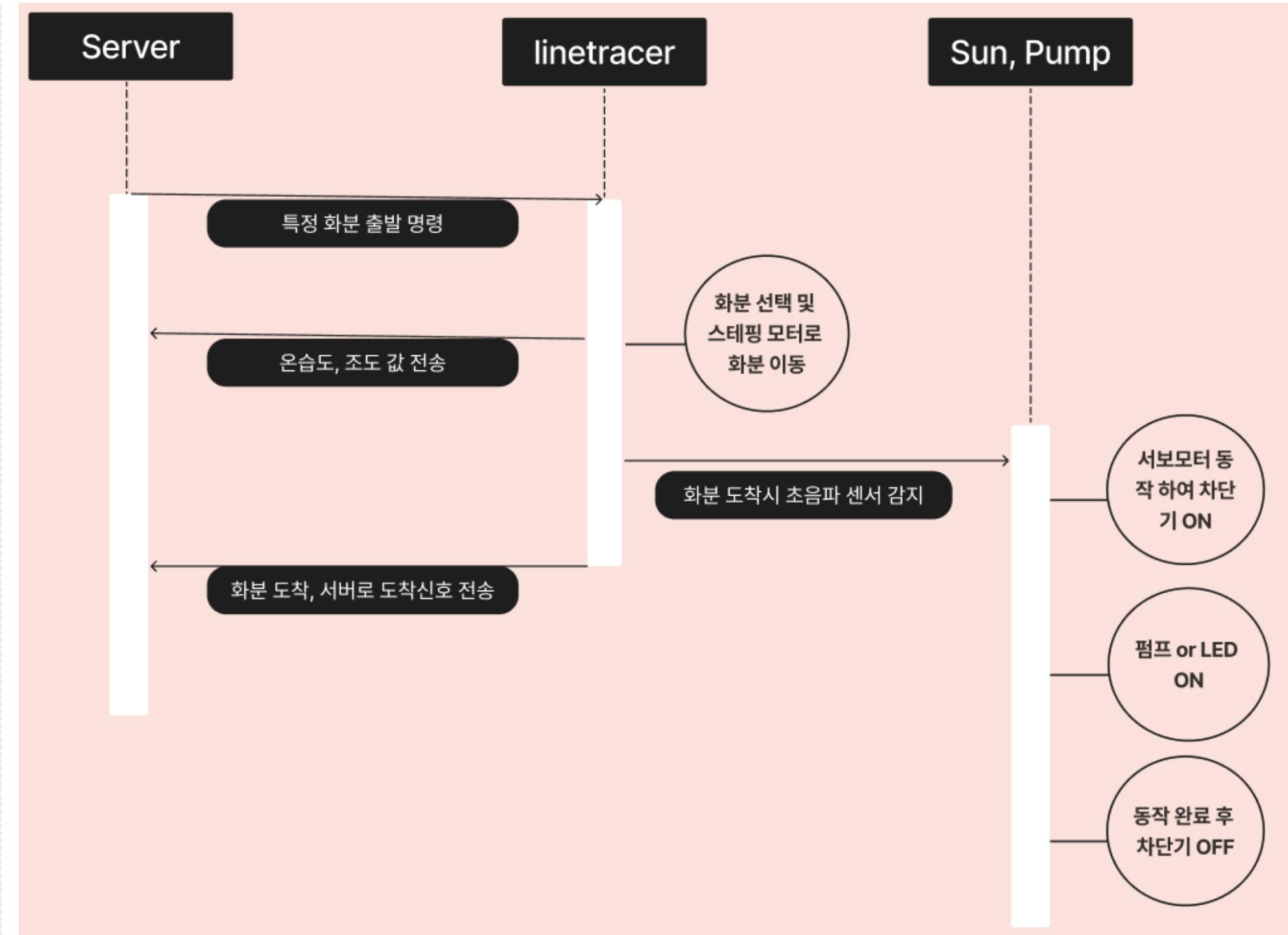
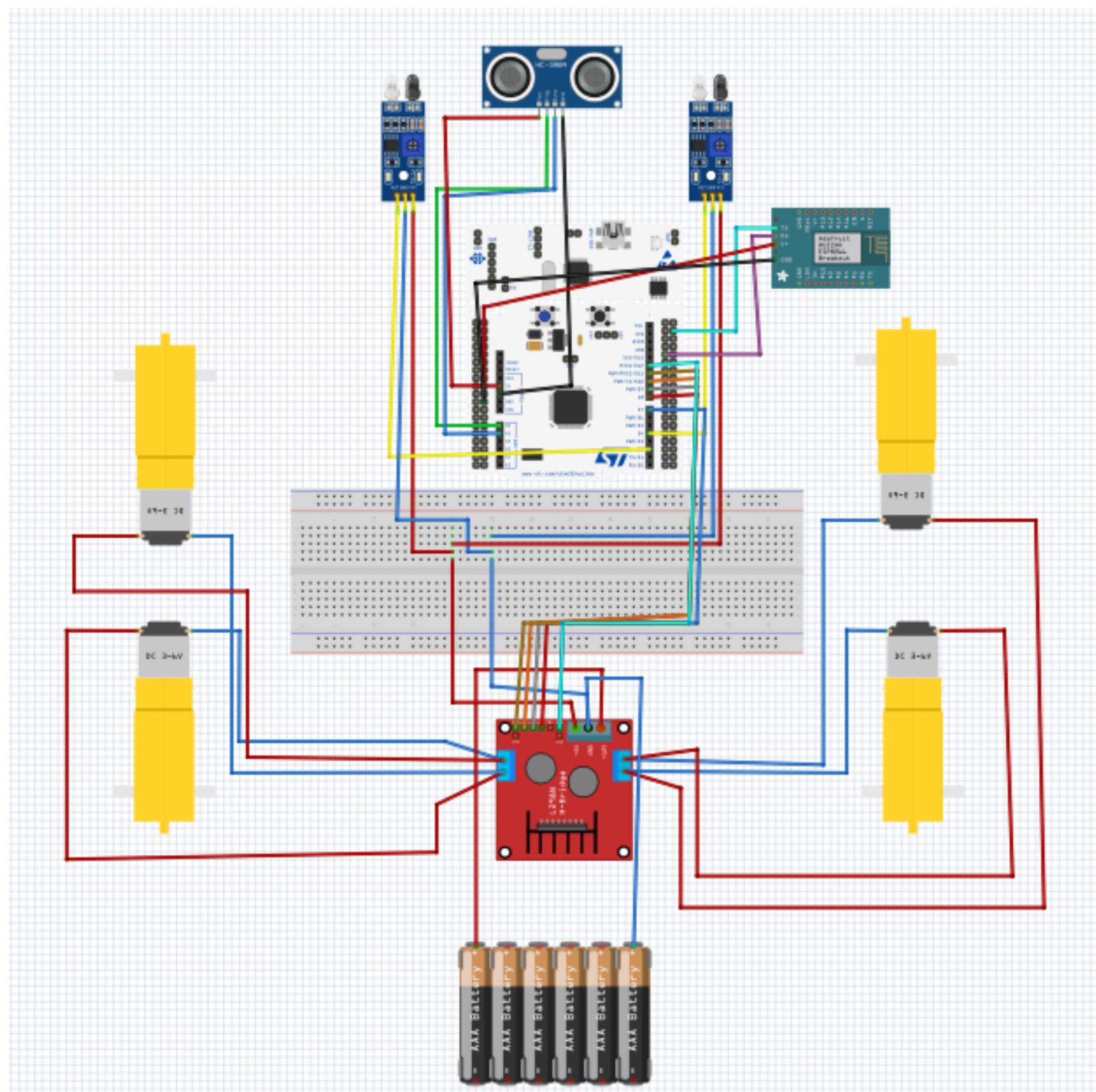
개발 툴

1. STM32 CUBE IDE

## line 동작 순서도



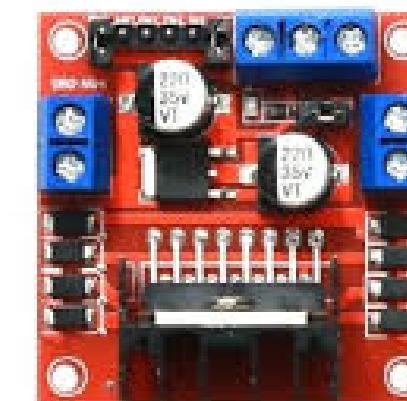
# 회로도, 시퀀스 다이어 그램



# Line Tracer 개발 과정

## 1. 라인 트레이서 개발

- 1) 모듈을 통해 라인 트레이서 동작 구현
- 2) 초음파 센서 제어
- 3) 온습도 센서 제어
- 4) 서버와의 통신 구현



## 2. Home Garden 개발

- 1) 초음파 센서 제어
- 2) Water Pump , Servo Motor , LED



# Line Tracer 개발 허들



1. 모터들의 노이즈 관리
  - a. 센서 동작 독립화
  - b. 전원부의 분리



3. 경로 알고리즘 구현
  - a. 모드 변경을 통한 센서 사용
  - b. 경로의 단순화

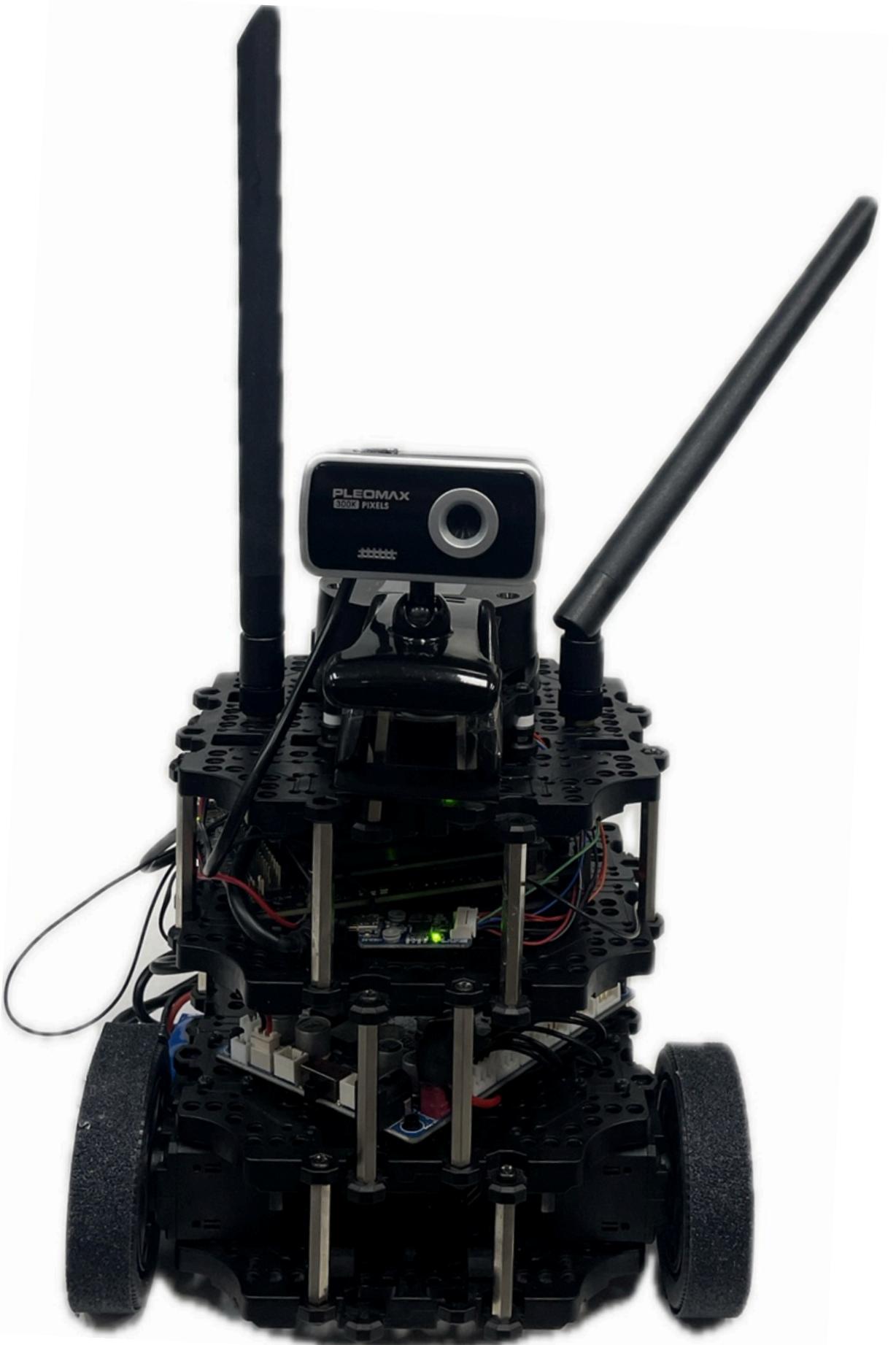


2. DC모터의 제어
  - a. PWM을 통한 속도 제어
  - b. 외부전원 인가



4. 화분 선택 방법
  - a. 서버와의 wifi통신
  - b. 화분 발판의 렉앤 피니언

# Turtlebot



개요

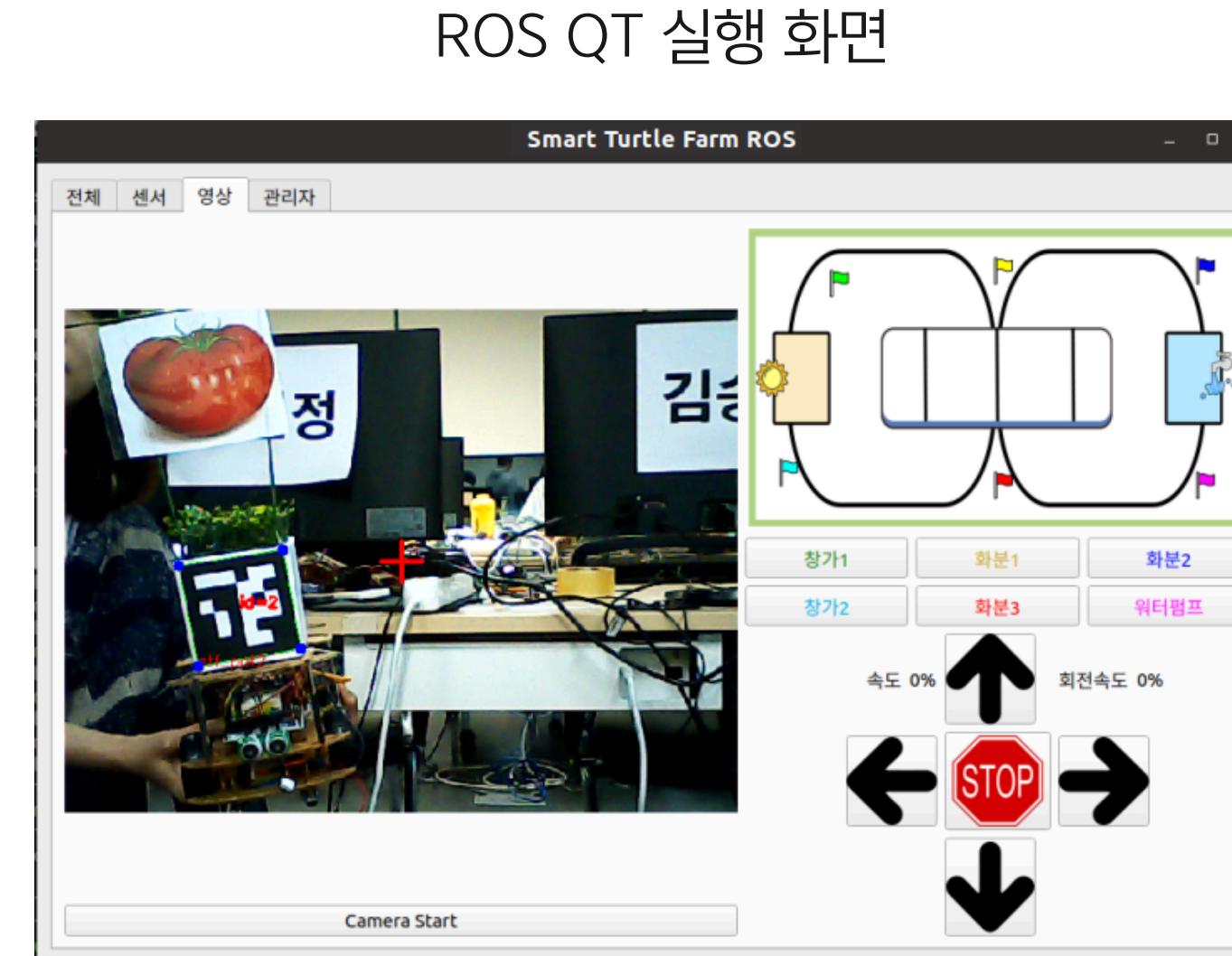
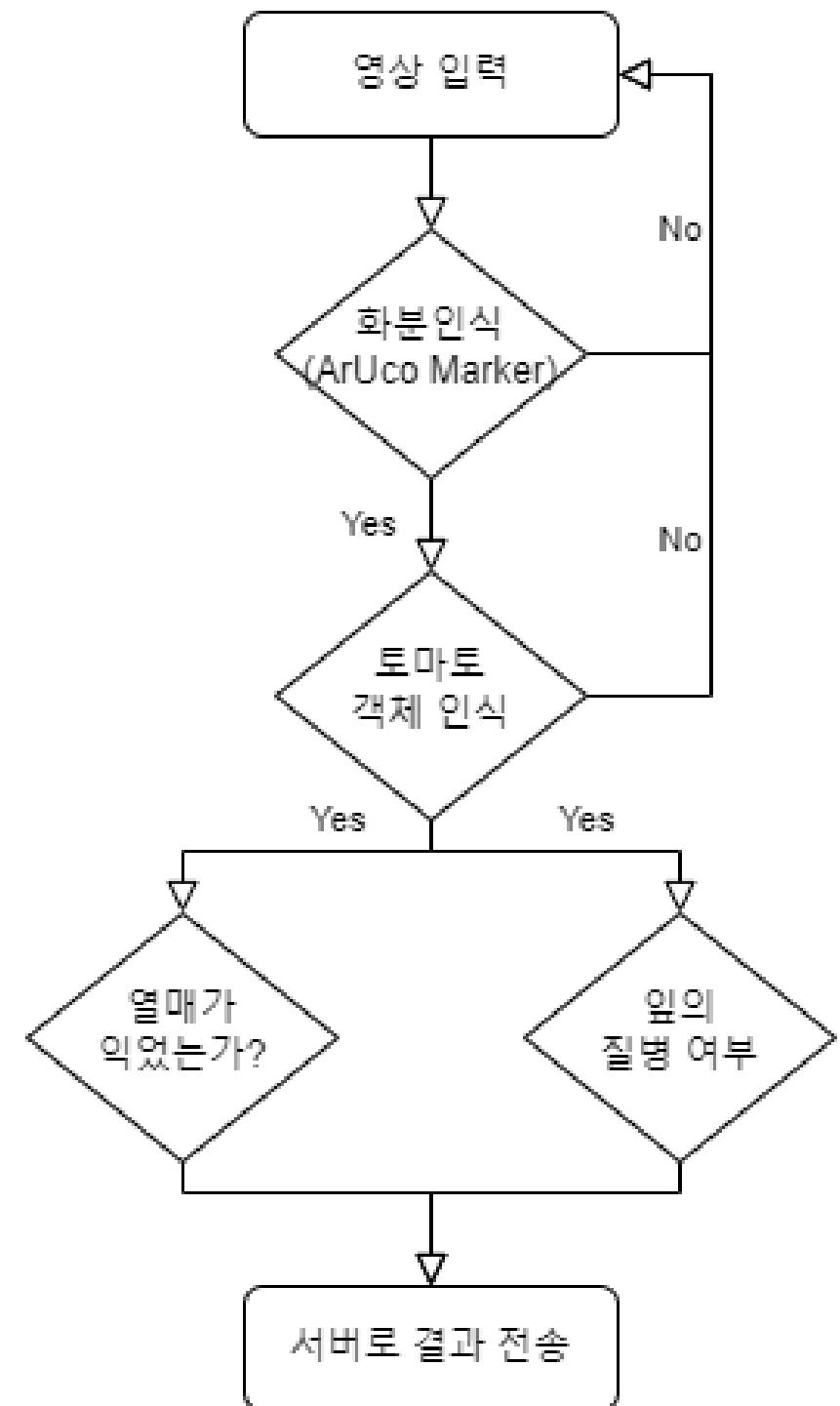
## 기능

1. 실내 주행
2. 식물 상태 모니터링
3. 서버와 socket 통신

## 사용 기술

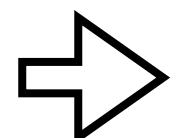
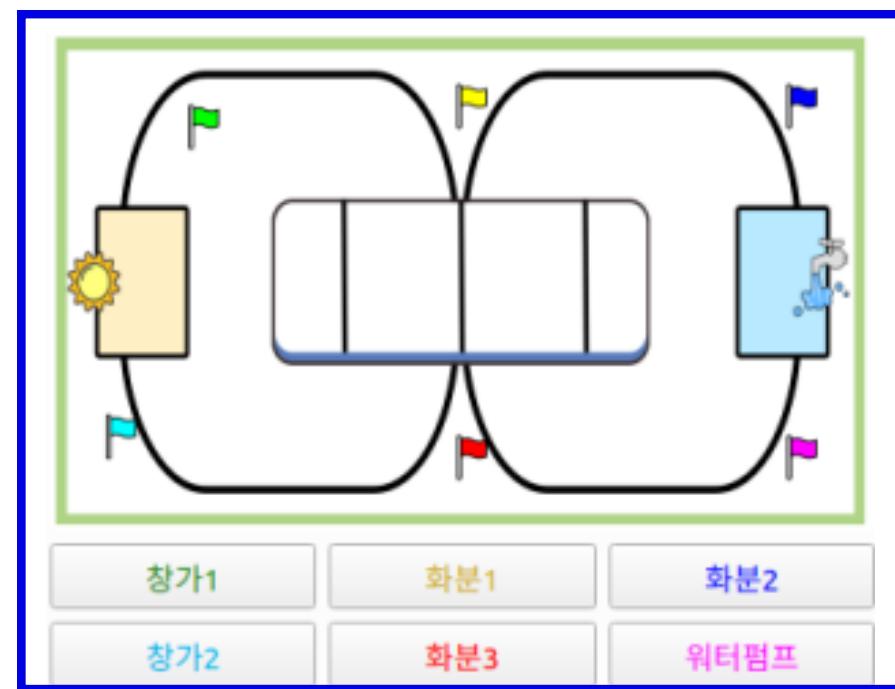
1. Jetson Nano
2. ROS, QT, C++
3. Yolov8, Tensorflow  
OpenCV, Python

# Turtlebot 동작 순서도 & 실행 화면

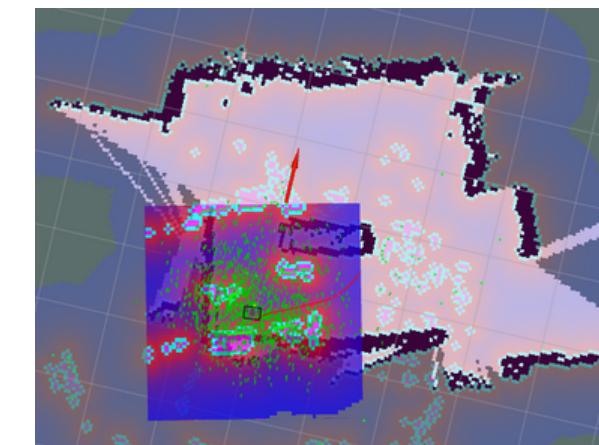


[STF\_SQL]{pot\_str}@Leaf@{status}@Tomato@{status}

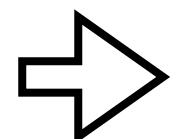
# Turtlebot Navigation



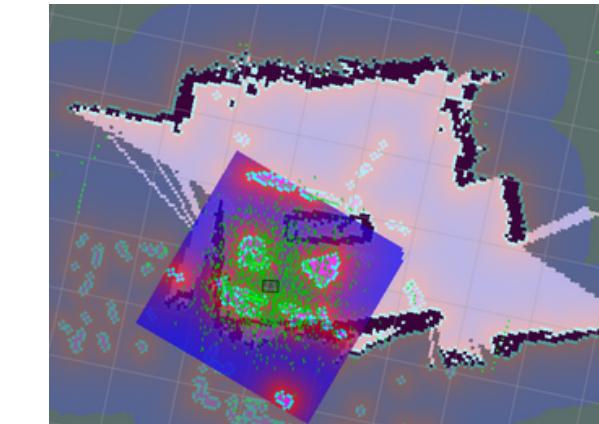
## 1. ROS 자동 주행 모드



```
void RosNode::go_goal(QString frame, double x, double y,  
                      double z, double w) {  
    geometry_msgs::PoseStamped goal;  
  
    goal.header.frame_id = frame.toStdString();  
  
    goal.header.stamp = ros::Time::now();  
    goal.pose.position.x = x;  
    goal.pose.position.y = y;  
    goal.pose.position.z = z;  
    goal.pose.orientation.z = w;  
    goal.pose.orientation.w = w;  
  
    goal_pub.publish(goal);  
}
```



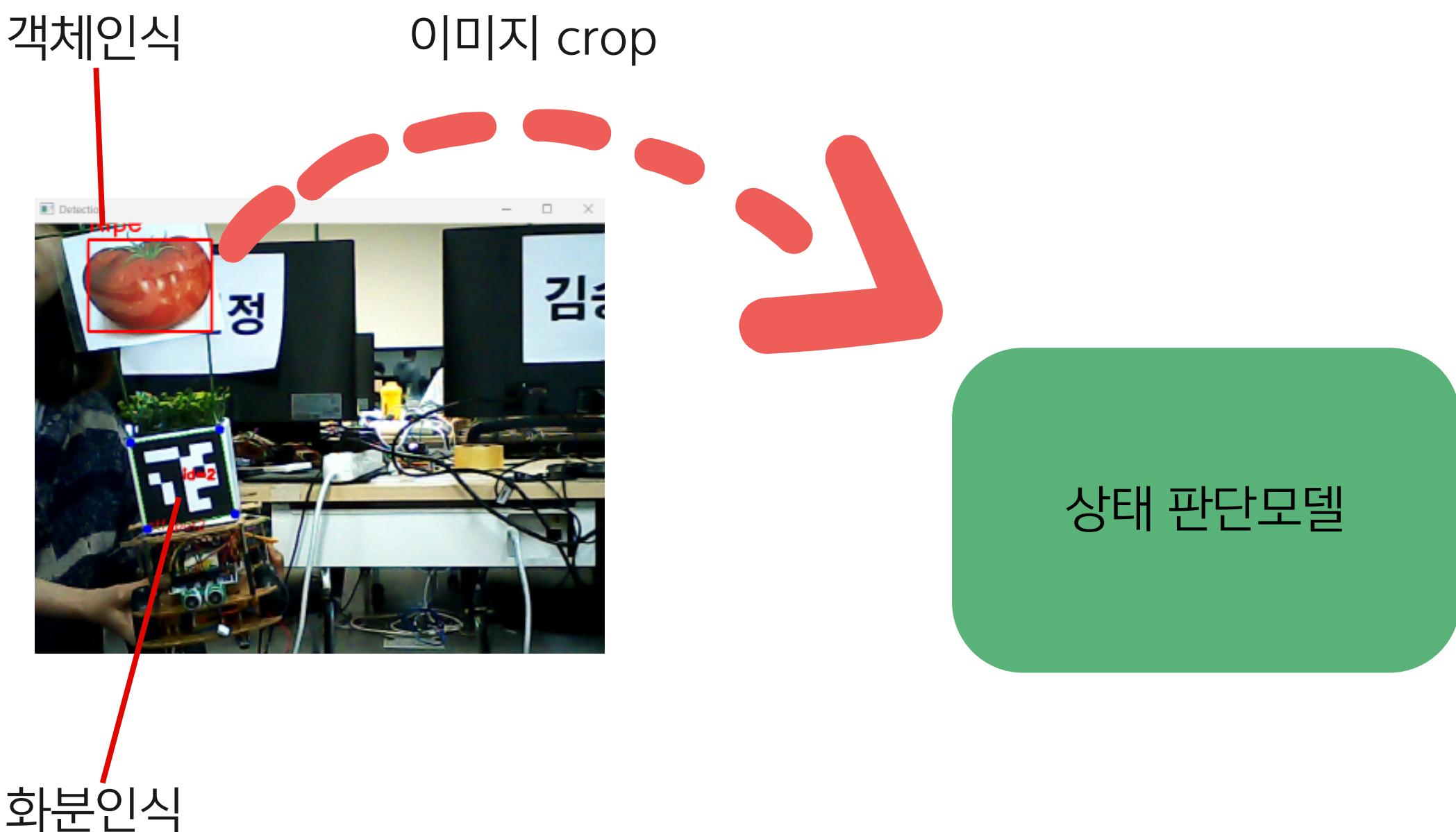
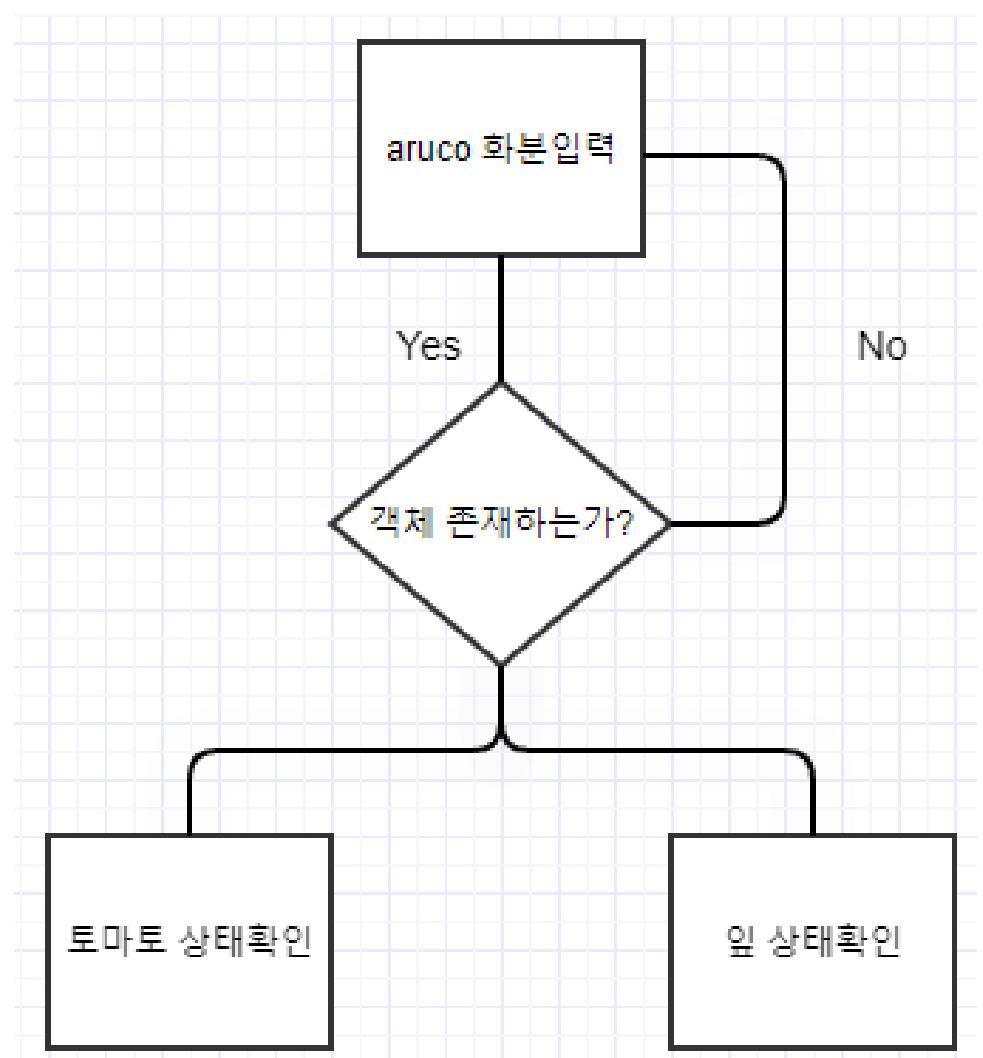
## 2. ROS 수동 주행 모드



```
prosNode->set_cmd_vel(base_cmd);  
prosNode->set_cmdBaseCmd(true);  
  
void RosNode::run()  
{  
    ros::Rate loop_rate(20); //20hz : 0.2Sec  
    while(ros::ok())  
    {  
        if(baseCmdFlag)  
        {  
            cmd_vel_pub.publish(base_cmd);  
        }  
        ros::spinOnce();  
        loop_rate.sleep();  
    }  
    emit rosShutdown();  
}
```

roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=\$HOME/map.yaml

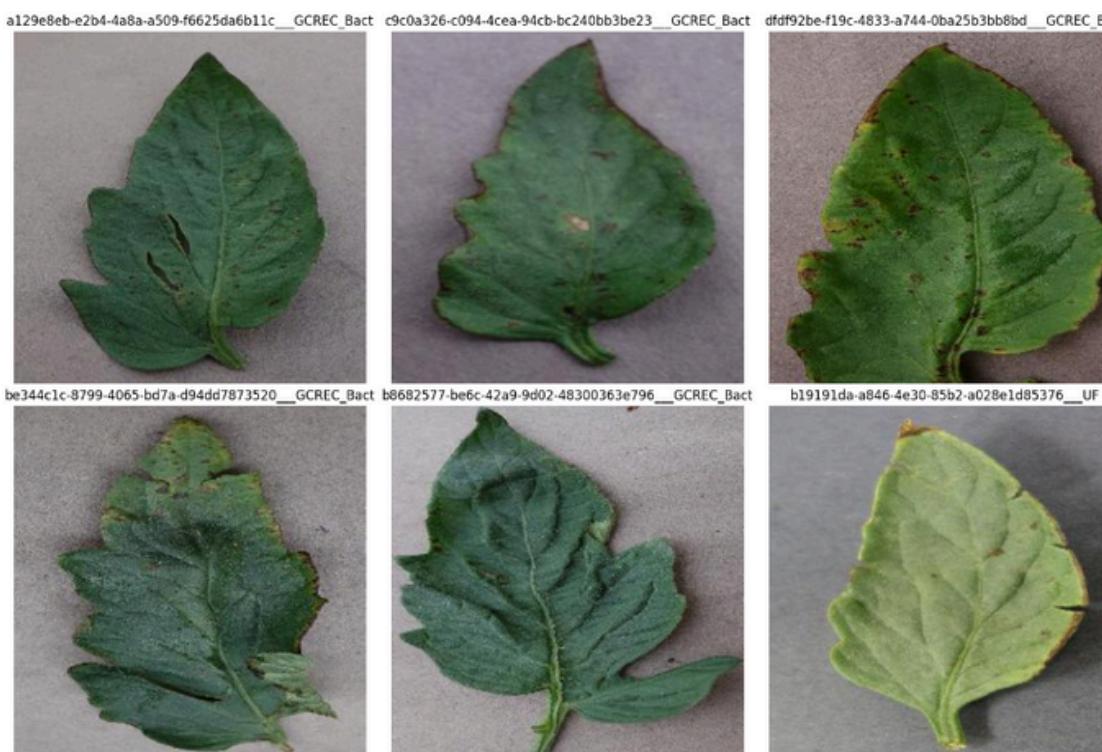
# 모델사용조건 알고리즘



# Tomato leaf disease classification (Tensorflow)

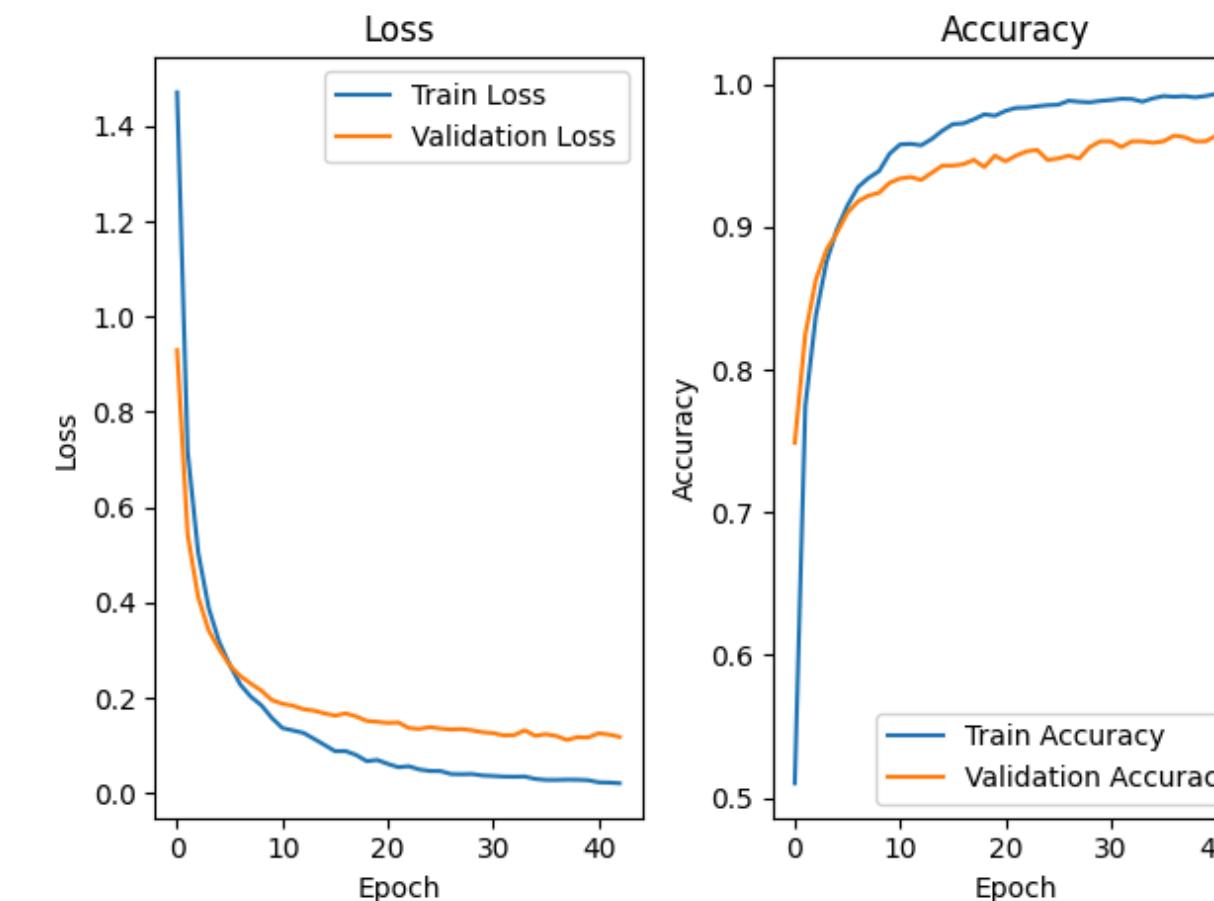
- 학습 데이터 : 토마토 잎 질병 (kaggle)

- Total : 11,100 image
  - . Train : 10,000 image
  - . Valid : 1,000 image
  - . Test : 100 image



- 수행 결과

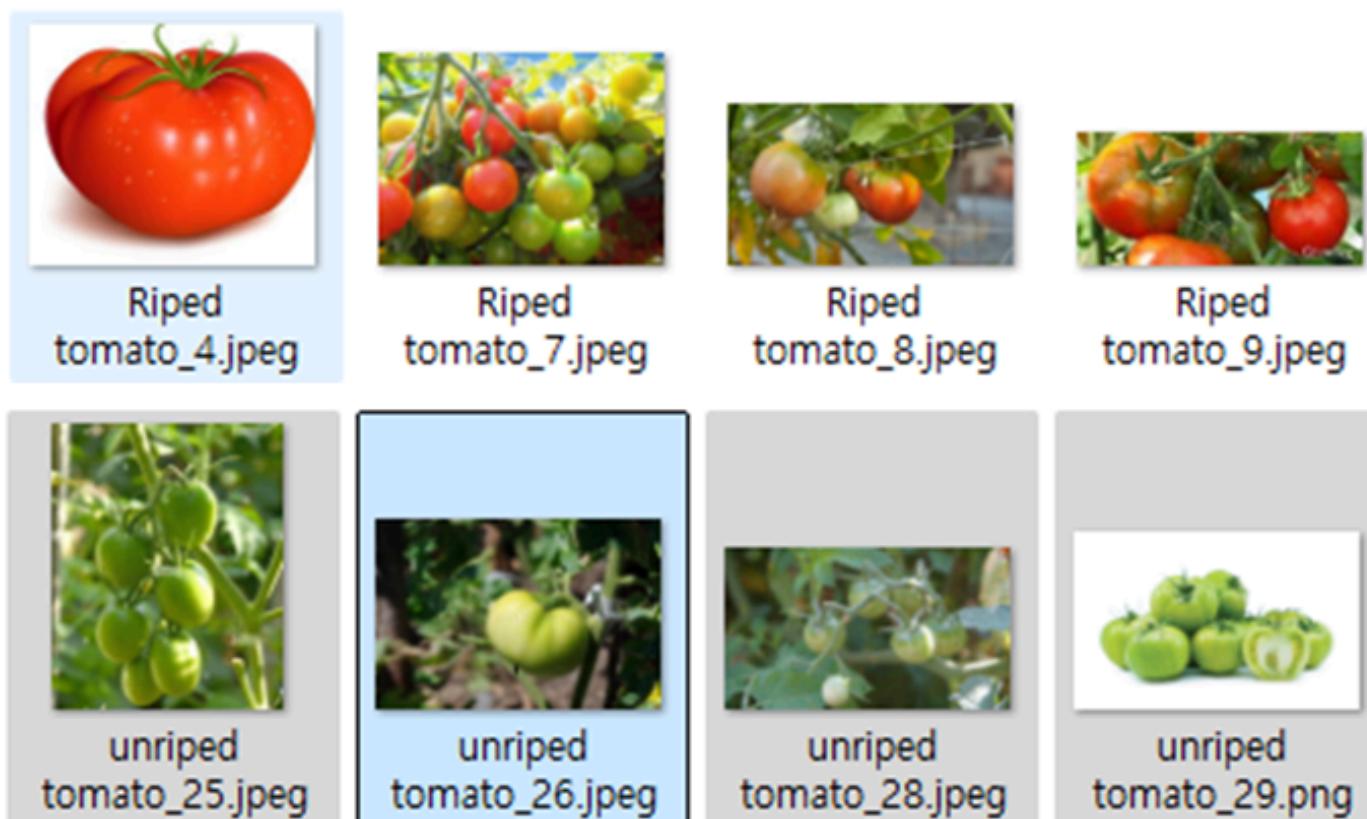
- Epoch : 100
- 학습이미지 질병 (10가지): class\_labels = ["Bacterial\_spot", "Early\_blight", "Late\_blight", "Leaf\_Mold", "Septoria\_leaf\_spot", "Spider\_mites", "Target\_Spot", "Yellow\_Leaf\_Curl\_Virus", "Tomato\_mosaic\_virus", "Healthy"]



# Tomato detection (Tensorflow)

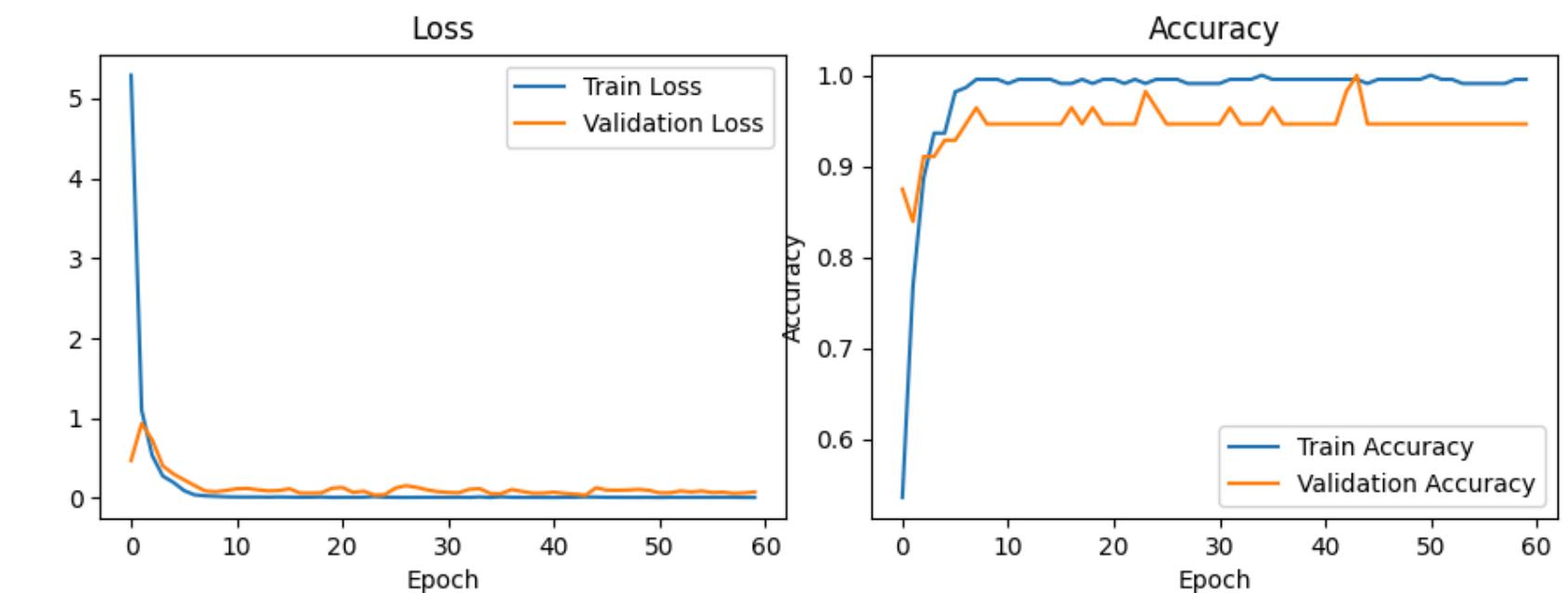
- 학습 데이터 : 토마토 열매

- Total : 298 image
  - . Riped : 145 image
  - . Unriped : 143 image

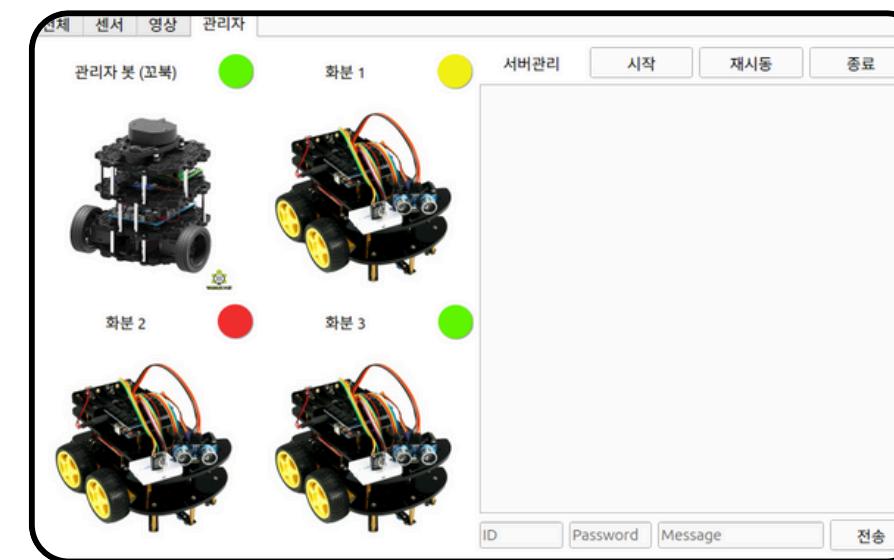
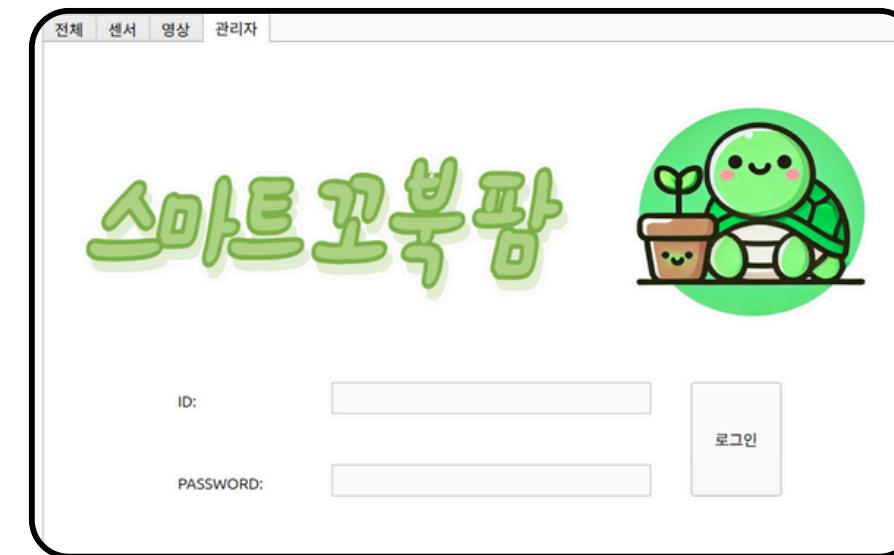
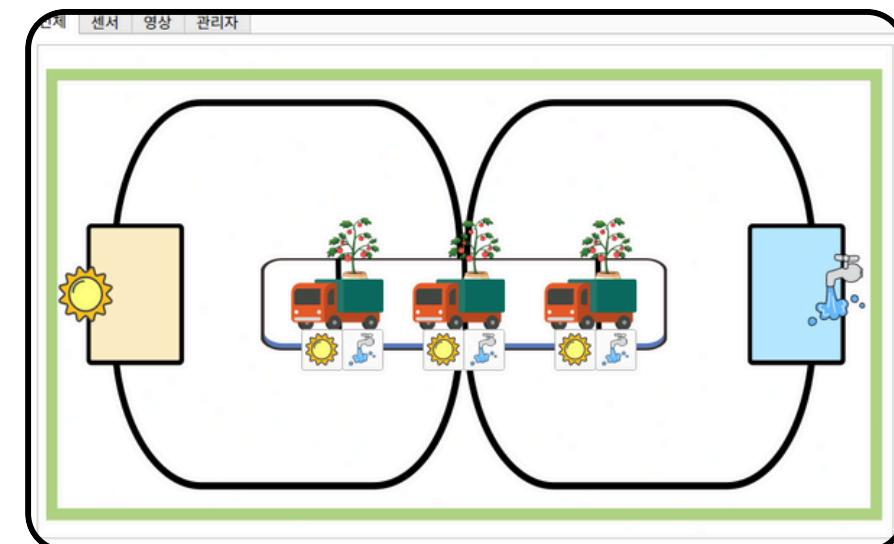
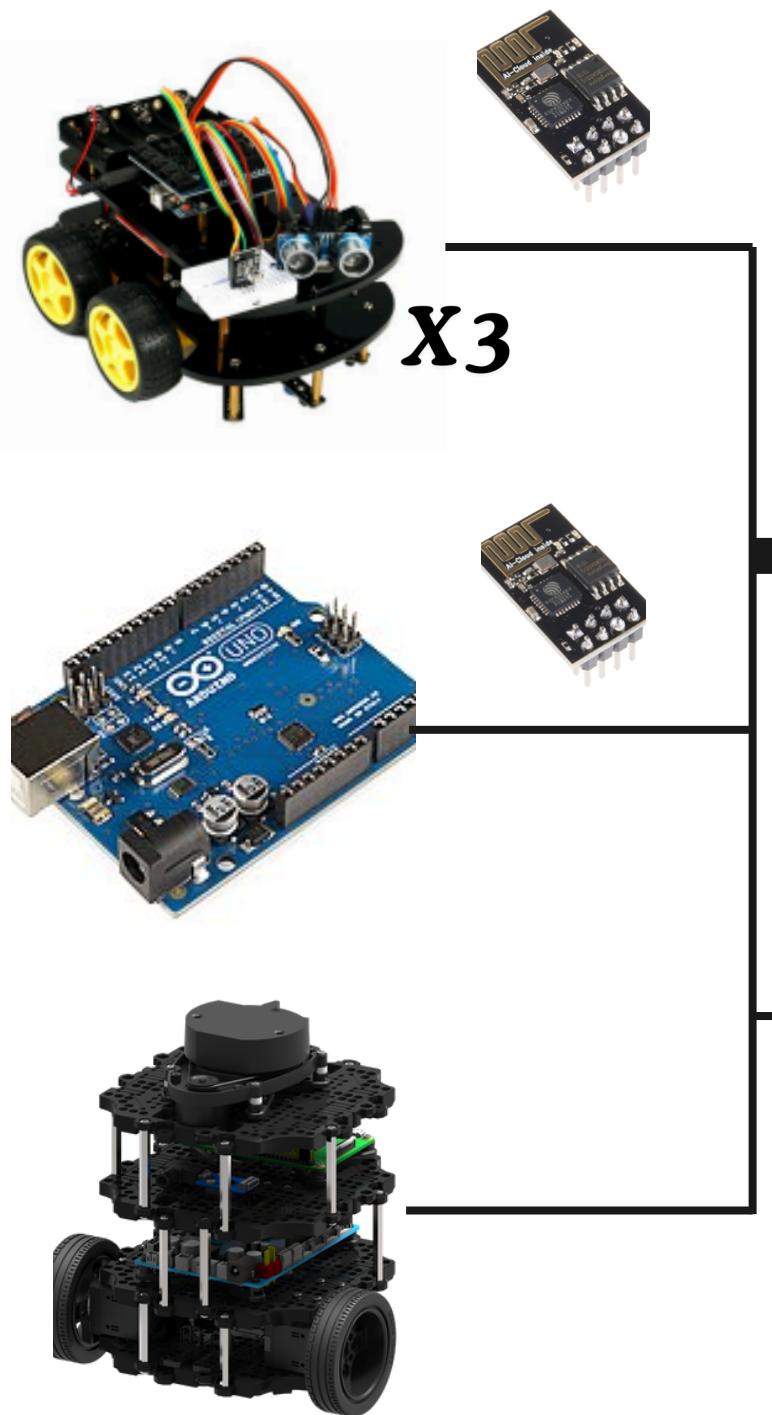


- 수행 결과

- Epoch : 60
- 분류 (2가지): class\_labels = ["Riped", "Unriped"]



# Server 소개



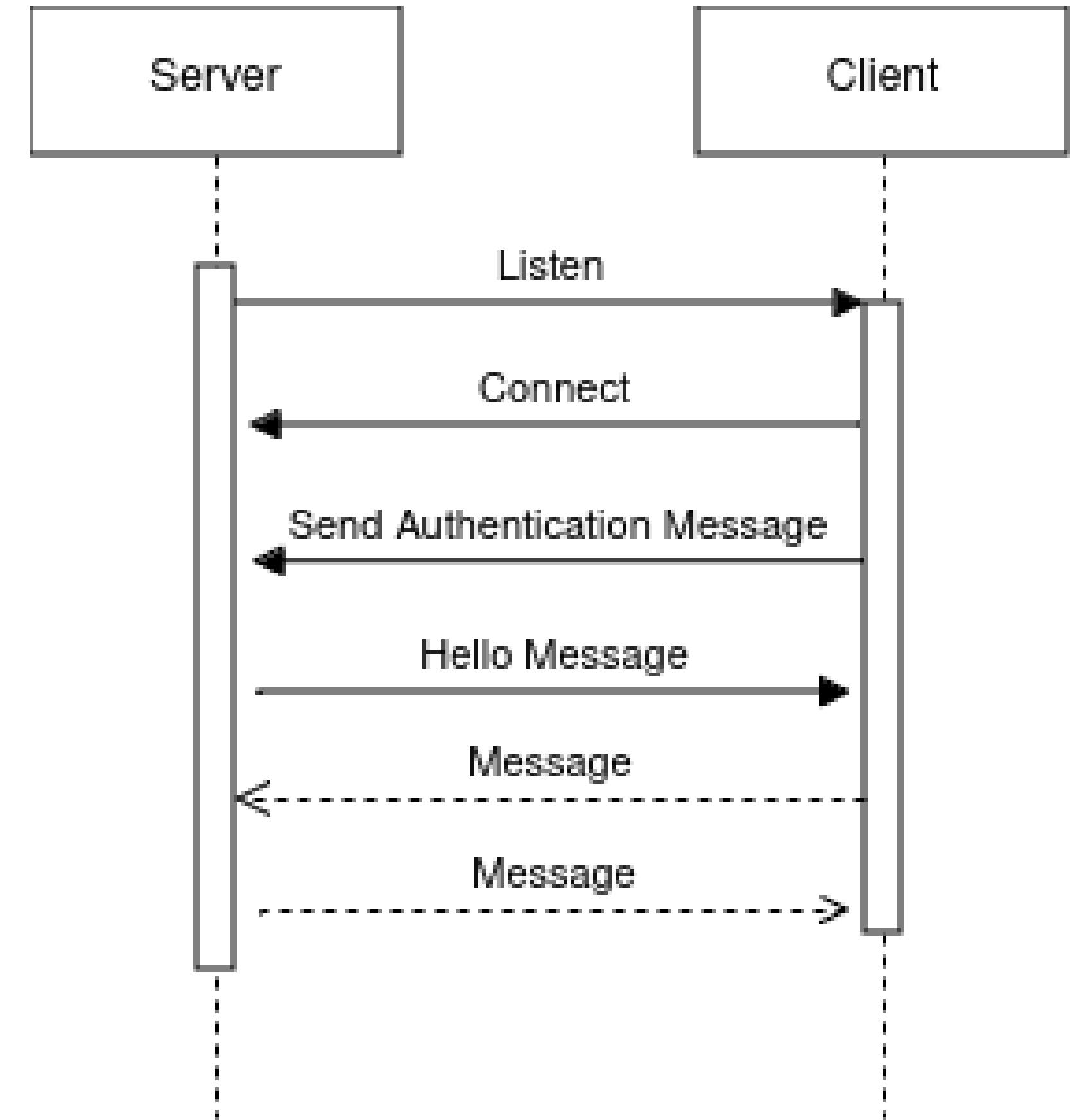
## 사용 기술

1. Qt
2. TCP/IP
3. Wi-Fi (ESP-01)
4. C++
5. Linux 20.04

# Server 시퀀스 다이어 그램

## 주요 기술 사항

1. MVVM 모델을 적용하여, 유지보수성 강화
2. 테스트 커버리지 비율 70% 이상
3. 서버 객체에 싱글톤 패턴 적용하여 안정성 강화



# Server 개발 허들 : Wi-Fi를 정복하라

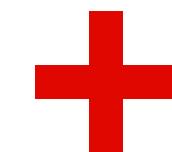
DATA SHEET  
CHECK!

## 1. ESP-01 모듈의 안정화

Microcontroller	ATmega328P	
DC Current for 3.3V Pin	50 mA	
Rx 802.11b, 1024 bytes packet length , -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA

1) 구동 전압 안정화

ESP-01 소비 전류>아두이노 공급 전류

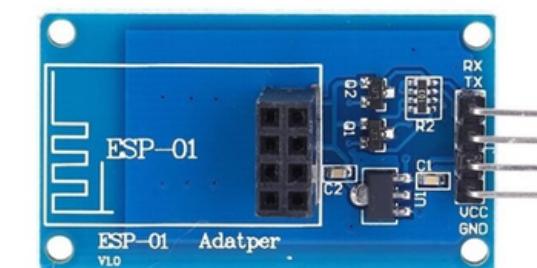
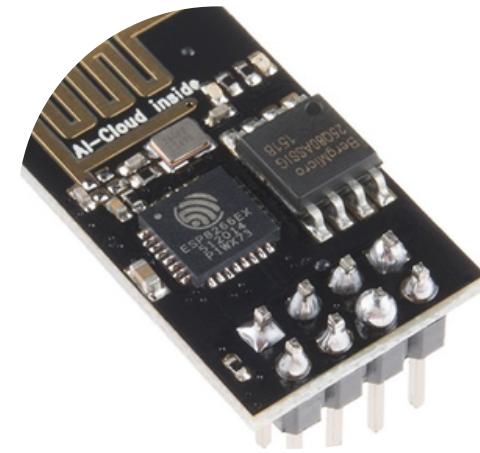


2) 펌웨어 업데이트



3) Baud rate 최적화

ESP-01 모듈의 전압 강하 현상



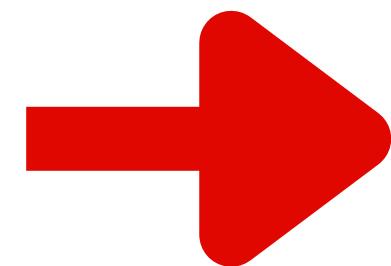
접속 속도 10배 개선  
(평균 10분 -> 1분)

# Server 개발 허들 : Wi-Fi를 정복하라

## 2. 혼잡한 Wi-Fi 환경 극복

1) 네트워크 혼잡도 체크

2) Soft AP 모드 사용



접속 안정성 10배 강화

시간 당 Reset 약 9~10회 -> 0회



# 소감

# 소감

정용재

1

천리길에서 이제 한 걸음 뗐다!

김윤우

2

모델의 성능을 분리하기위해서 모델 사용하는 상황을 설정하고 알고리즘을 작성하는 부분이 어려웠지만 해결하는 재미가있었다.

김지훈

3

팀원 분들에게 많이 배웠습니다.

유지승

4

STM으로 여러가지 센서들을 구현하면서 어려움이 많았지만 해결하며 동작시키는 게 재미있었고 많이 발전하게되는 계기가 된것 같습니다. RTOS를 통해 온습도를 하지 못한게 아쉽습니다.

이지원

5

여러가지 개발 환경을 경험해 보았고, 과정 별 프로젝트마다 조별 협업을 했던 것이 기억에 남습니다. 모델 optimization에서 아쉬움이 남습니다.

*Q&A*

## 질의 응답