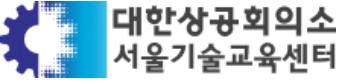


# 투게더

군집을 활용한 자율주행 경작지 관리 로봇

# 팀원 소개



김재용

담당 :  
팔로워 로봇



오민지

담당 :  
로봇 팔



황진영

담당 :  
리더 로봇



윤치영

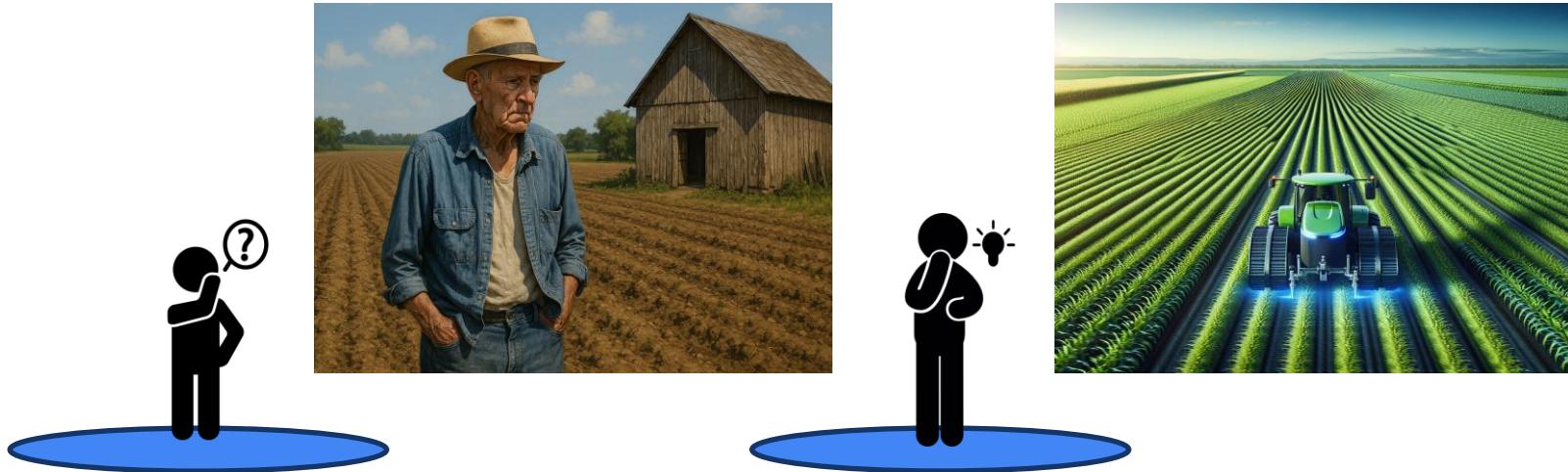
담당 :  
프로젝트 총괄 및 PM

# 목차

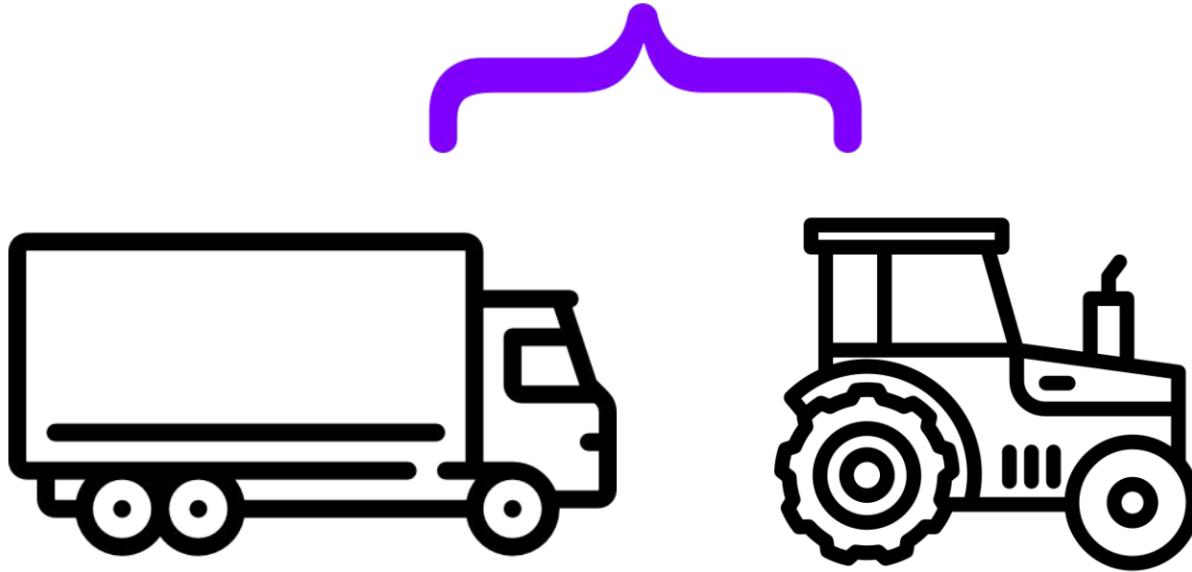


1. 개요
2. 수행 과정
3. 수행 결과
4. 문제 해결
5. 소감
6. 질문

# 개요 – 선정 배경

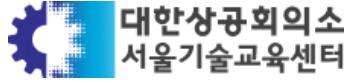


# 개요 - 목표



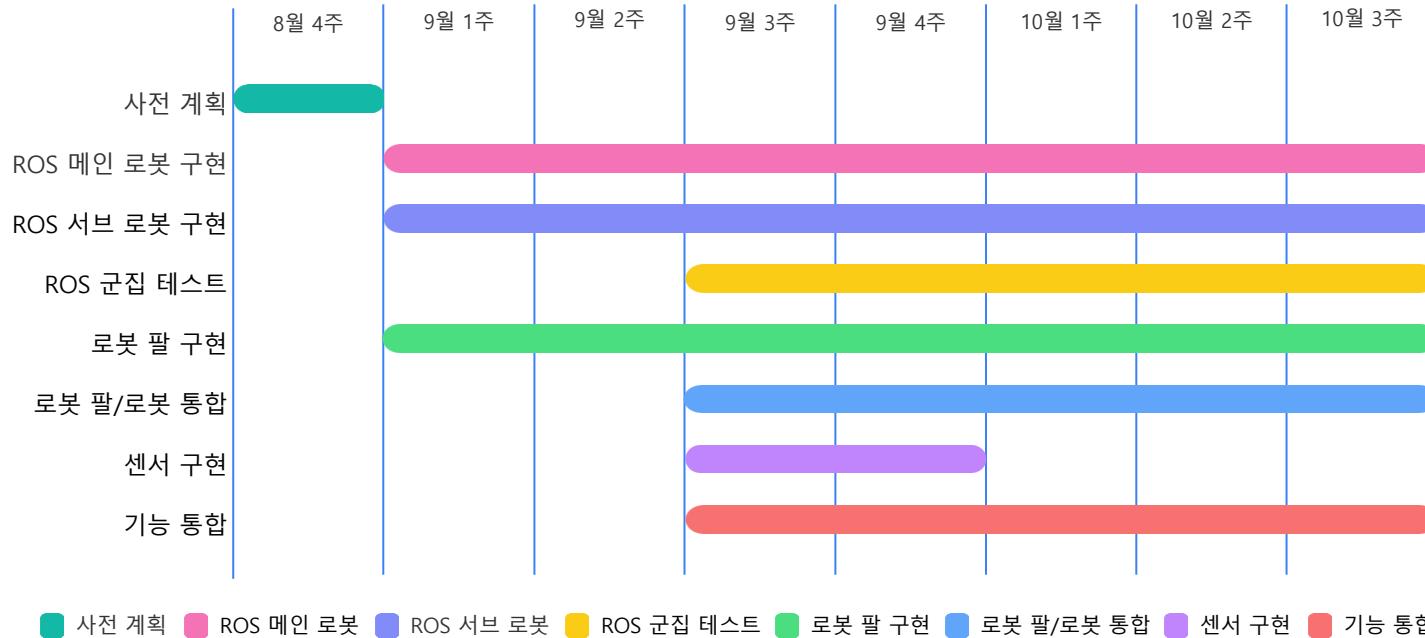
차량 군집을 사용한 자율 경작지 관리 시스템

# 개요 - 일정



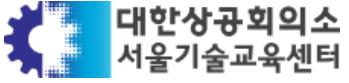
## 로봇 구현 간트 차트

8월 4주차 ~ 10월 2주차



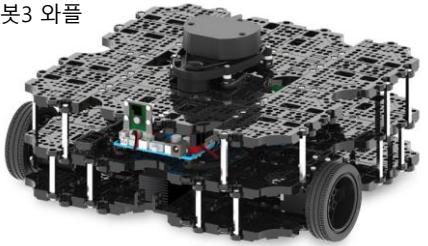
■ 사전 계획 ■ ROS 메인 로봇 ■ ROS 서브 로봇 ■ ROS 군집 테스트 ■ 로봇 팔 구현 ■ 로봇 팔/로봇 통합 ■ 센서 구현 ■ 기능 통합

# 수행 과정 – 구성도



## 리더 로봇

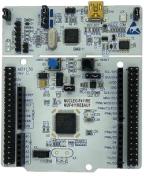
터틀봇3 와플



로봇 팔



STM32



거리측정



- 개발 환경

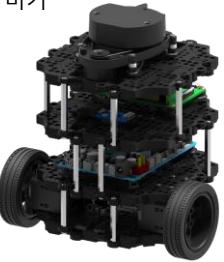


Linux



## 팔로워 로봇

터틀봇3 버거



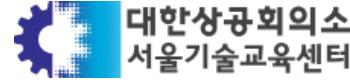
- 개발 환경



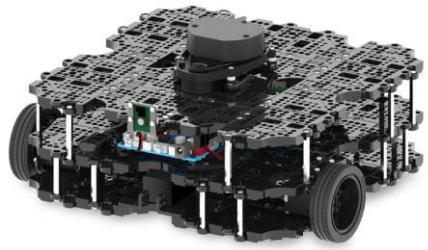
Linux



# 수행 과정 – 리더 로봇



리더 로봇

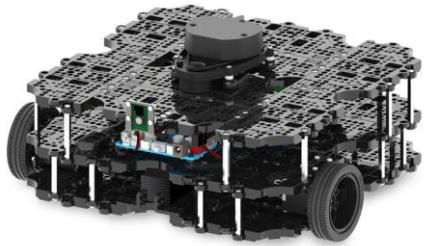


## Patrol 개요

- ROS2 Patrol은 로봇이 지정된 경로 또는 지점들을 **순찰(Patrol)**하며 이동하는 기능을 수행하는 노드다.
- SLAM 또는 맵 기반 네비게이션을 활용해 **여러 목표 지점을 순차적으로 방문**한다.
- 주로 보안, 감시, 모니터링, 물류 검사 등에 사용된다.

# 수행 과정 – 리더 로봇

리더 로봇

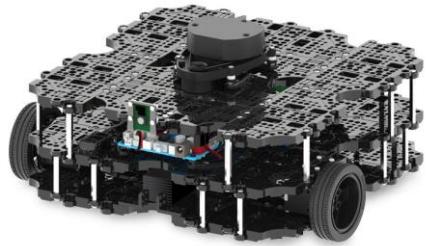


## 주요 역할

역할	설명
경로 관리	순찰할 좌표 리스트(waypoints)를 관리하고 순서대로 방문한다.
목표 전송	각 지점을 nav2의 NavigateToPose 액션으로 전송한다.
상태 모니터링	도착 여부, 실패, 장애물 감지 등의 상태를 확인한다.
루프 제어	모든 지점 방문 후 반복 순찰(Loop) 여부를 제어한다.
복구 처리	경로 장애 발생 시 재시도 또는 우회 처리한다.

# 수행 과정 – 리더 로봇

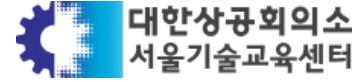
리더 로봇



## 구성 요소

구성 요소	설명
<b>토픽(Topics)</b>	<ul style="list-style-type: none"> <li>- /amcl_pose: 현재 로봇 위치</li> <li>- /map: 지도 정보</li> <li>- /cmd_vel: 속도 명령</li> <li>- /scan: 장애물 감지</li> </ul>
<b>액션(Actions)</b>	<ul style="list-style-type: none"> <li>- /navigate_to_pose: nav2_msgs/action/NavigateToPose (목표 이동)</li> </ul>
<b>노드(Nodes)</b>	<ul style="list-style-type: none"> <li>- patrol_node: 순찰 로직 수행 - nav2_bt_navigator: 네비게이션 실행</li> </ul>
<b>파라미터(Parameters)</b>	<ul style="list-style-type: none"> <li>- waypoints: 순찰 좌표 리스트 - loop: 반복 여부 - tolerance: 도착 허용 오차</li> </ul>
<b>제어 알고리즘(Control)</b>	<ul style="list-style-type: none"> <li>- 네비게이션 스택(nav2)에서 경로 계획 및 장애물 회피 자동 수행</li> </ul>
<b>Launch 파일</b>	<ul style="list-style-type: none"> <li>- SLAM / 맵 로딩, nav2 bringup, patrol node 실행 포함</li> </ul>

# 수행 과정 – 팔로워 로봇



팔로워 로봇



## Follower 개요

- ROS2 Follower는 리더(Leader) 로봇 또는 목표 대상의 위치·속도·경로 정보를 추종하는 노드이다.
- 리더-팔로워 구조를 통해 다중 로봇 제어, 사람 추종, 라인 트레이싱 등의 기능을 구현한다.
- 기본 원리는 리더 상태 구독 → 상대 위치 계산 → 추종 제어 명령 생성이다.

# 수행 과정 – 팔로워 로봇

## 팔로워 로봇



### 주요 역할

역할	설명
리더 정보 수신	리더의 위치, 속도, 방향 등 상태 정보를 구독한다.
상대 거리·방향 계산	Follower와 Leader 간 거리 및 각도 오차를 계산한다.
추종 제어	오차를 줄이기 위해 속도(linear, angular) 명령을 생성한다.
자체 센서 피드백 사용	LiDAR, 카메라, IMU 등 센서로 추종 정확도 보정.
안전 제어	장애물 회피 및 속도 제한 처리.

# 수행 과정 – 팔로워 로봇

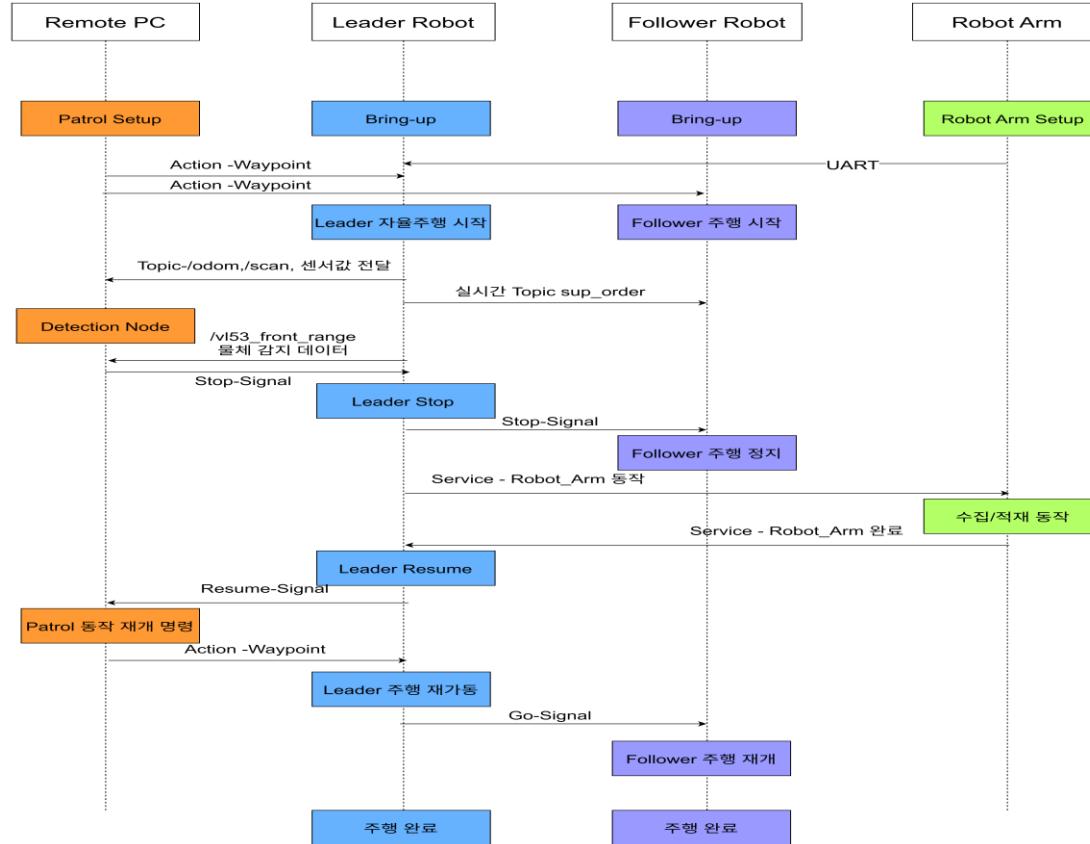
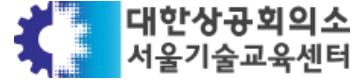
## 팔로워 로봇



### 구성 요소

구성 요소	설명
토픽(Topics)	- /leader/cmd_vel: 리더의 속도 명령 - /odom: 팔로워의 자기 위치 - /cmd_vel: 팔로워가 실제 주행 명령을 퍼블리시
노드(Nodes)	- leader_node: 리더 상태 퍼블리시, - follower_node: 리더를 추종
메시지(Messages)	- geometry_msgs/msg/Twist: 속도 - nav_msgs/msg/Odometry: 위치·자세
제어 알고리즘(Control)	PID 제어, Pure Pursuit, Visual Tracking 등
센서(Sensors)	LiDAR, 카메라, 초음파, IMU 등 (선택적 사용)
Launch 파일	리더와 팔로워 노드를 동시에 실행하는 설정 포함

# 수행 과정 – 시퀀스 다이어그램

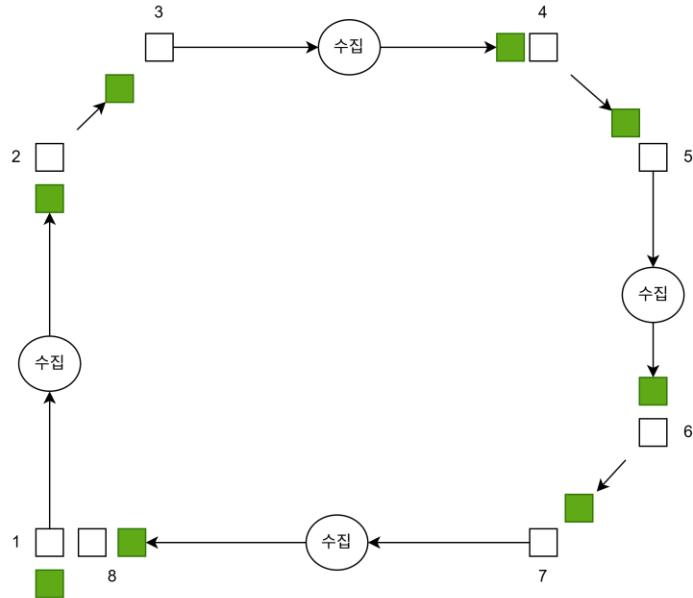


# 수행 과정 – 로봇 경로

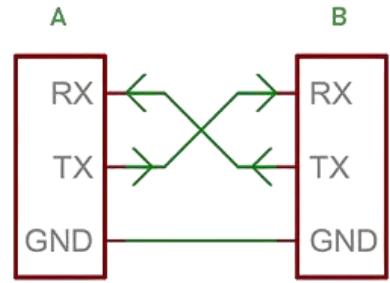
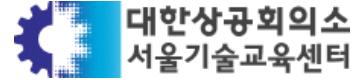
Leader Robot Patrol Autonomous Driving 

Follower Robot Patrol Autonomous Driving 

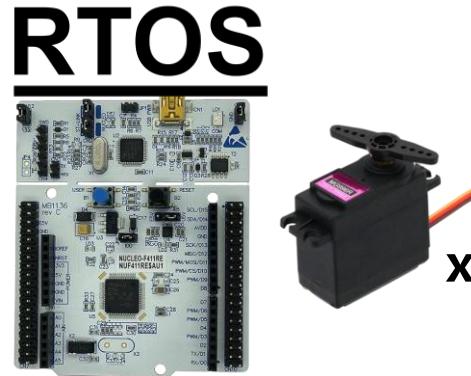
1~8 : Waypoint



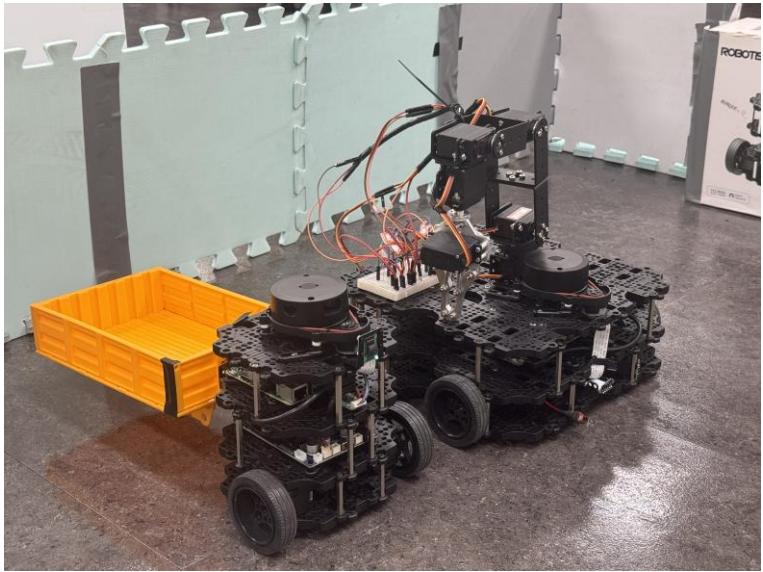
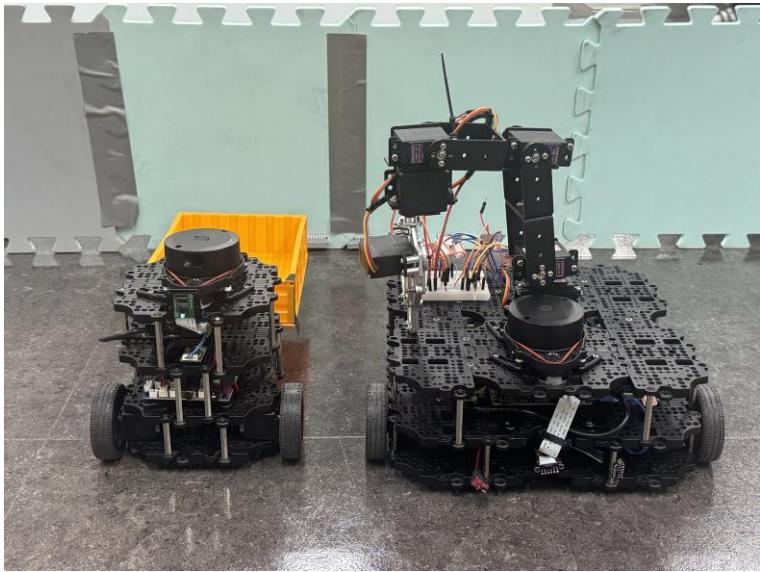
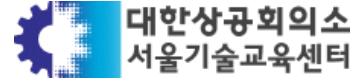
# 수행 과정 – 로봇 팔



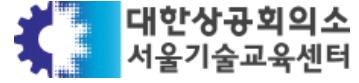
DMA + IDLE 인터럽트



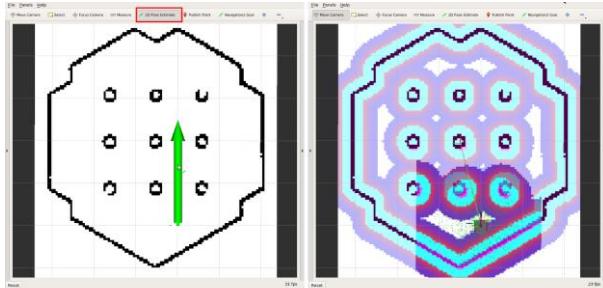
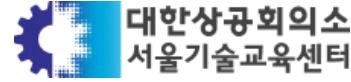
## 수행 결과 – 사진



# 구현 영상



# 문제 해결 – 리더 로봇



## Navigation2 정밀한 경로 이동 문제

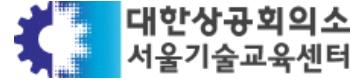
문제 : Navigation2를 활용하여 경로를 이동 시킬 경우 **수직 이동을 하지 않음**

현상 : 로봇이 자리를 찾지 못함, 지그재그 운행, 잦은 재계획등 반복

원인 : LiDAR 불안정, 네트워크 지연 등 복합적 상황으로 예측

해결 : **Patrol 기반**을 활용하여 안정적 주행과 **웨이포인트 기반** 동작 구현

# 문제 해결 – 리더 로봇



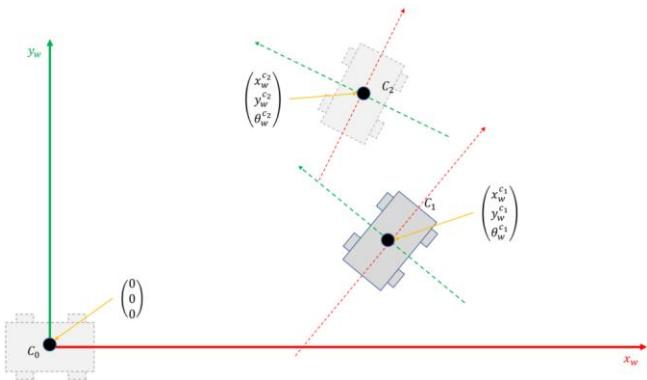
## 시뮬레이션과 실제 환경 경로 좌표 오차

### 문제

1. Gazebo 시뮬레이션에서 Patrol 좌표를 구성 및 해당 맵 기반 좌표를 추출하여 검증
  2. 실제 현장에서 같은 좌표를 적용하였지만 좌표 인식 문제로 원하는 Patrol 동작이 안됨
- 원인 : 바닥 재질/ 좌우 비대칭, 미끄러짐, 급가감속등이 문제로 예측
- 해결 : 시뮬레이션 환경이 아닌 실제 환경에 맞춰 좌표값을 직접 추출하여 Patrol 운용



# 문제 해결 – 팔로워 로봇

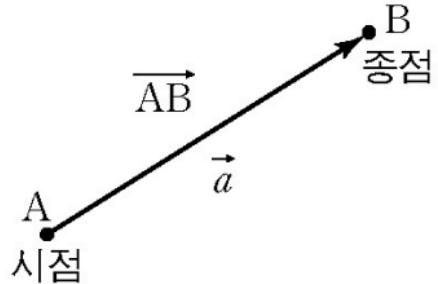


## TF2 오차

원인 : 시뮬레이션과 달리, 현실에선 TF2 좌표계 사용 시 오차로 인해 섬세한 대형 유지에 난항

해결 : 좌표계의 비중을 줄이고, 초기 상태의 거리 차이를 그대로 사용하는 방식으로 변경

# 문제 해결 – 팔로워 로봇



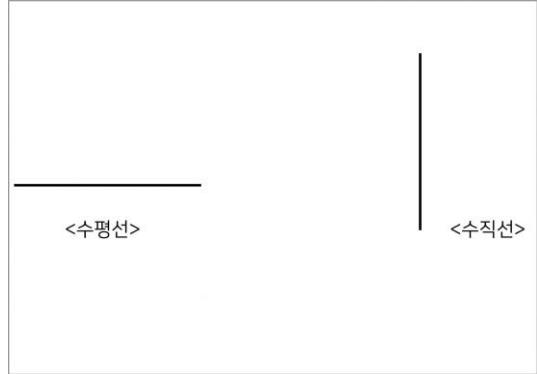
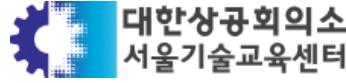
## 방향 이슈

원인 : 네트워크 지연, 좌표계 등의 복합적 이유

시도 : map 사용, 회전 시 Following 중지 등 다양한 방법 시도

해결 : 회전 이후 Service로 신호, 팔로워를 재정렬하는 방식 적용

# 문제 해결 – 팔로워 로봇



## 수평상 이동시 경로 문제

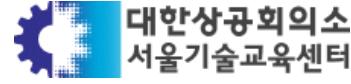
원인 : 수평상 이동시 리더 로봇과 팔로워 로봇 충돌

현상 :

1. 파라미터 튜닝 -> 정교한 동작 구현
2. 시뮬레이션과 달리 실제로는 실제 환경에서는 오작동
3. 회전 시 정지 & 회전 이후 동작 -> 다른 대형으로 작동
4. 회전 이후 대형 유지를 위해 cmd\_vel을 조작 -> 대형 유지 실패

해결 : 수평상 대형 유지가 아닌 수직선상 대형 유지로 해결

# 문제 해결 – 팔로워 로봇



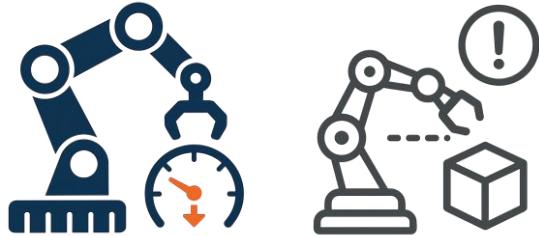
## Follower 동작에 따른 대형 유지 및 방향전환 문제

원인 : 리더 로봇과 팔로워 로봇의 경로 겹침 및 잦은 충돌

현상 : 충돌로 인한 좌표 틀어짐 대형도 많이 틀어져 원하는 동작 수행 불가

해결 : Follower기반 대신 Patrol 기반으로 군집 활동을 구현

# 문제 해결 – 로봇 팔

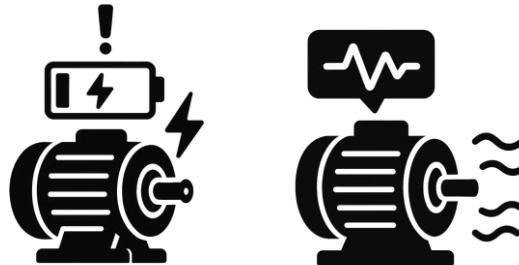


## 1. 토크 및 길이 부족

문제 : 로봇팔 길이·토크 부족

원인 : AD002 서보의 낮은 토크

해결 : MG996R 서보로 교체



## 2. 전력 문제 및 노이즈

문제 : 서보 모터 부하로 인한 노이즈 문제

원인 : 전원 불안정, 높은 부하의 시작 자세

해결 : 외부 전원 분리·공급, 자세 재조정, 불량 모터 교체

# 문제 해결 – 로봇 팔



## 3. 부하 분산 및 제어 성능 향상

문제 : 모터별 독립 제어 및 각도·속도 조절 필요

원인 : 서보 모터 성능 향상을 위해 세밀한 각도 제어 필요

제한 : Arduino는 RTOS 적용 부적합

해결 : STM32로 보드 교체 후 RTOS 기반 서보 모터 Task 분리 제어



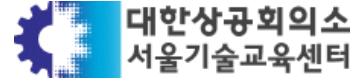
## 4. 그리퍼 모터 각도 문제

문제 : 각도 제어 시 노이즈 및 과전류 발생

원인 : 기구 동작 범위 이상의 PWM 값 사용

해결 : PWM 값 수정

# 소감 – 김재용



김재용

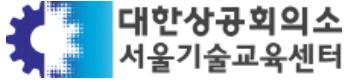
담당 :

팔로워 로봇

## 소감 :

- 인텔 AI 과정은 **지식을 쌓는 것을 넘어, 모르는 문제에 접근하는 방법을 배움**
- BSP, AI, ROS 등 어떤 분야를 다루든 **문제를 해결하는 방법은 다양하며, 그중 가장 효율적이고 안정적인 접근법을 신중히 선택해야 한다는 사실을 배움**
- **메뉴얼 숙지 및 자신만의 프로그래밍 방법론을 확립의 중요성을 배움**
- 이번 교육을 통해 **문제 해결 과정을 폭넓게 고려할 줄 아는 임베디드 개발자로 성장함**

# 소감 – 오민지



오민지

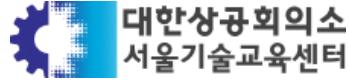
담당 :

로봇 팔

## 소감 :

- 인텔 교육 과정을 통해 펌웨어, BSP, AI, ROS 등을 배울 수 있었으며, 대학교에서는 접하기 어려운 **실무 중심의 임베디드 교육**이라고 생각함
- 실습과 프로젝트를 진행하면서 **임베디드 개발자로 성장함**
- 시행착오 속에서도 팀원들과 **함께 문제를 해결하며 큰 보람을 느낌**
- 하드웨어와 소프트웨어를 함께 다루면서 단순한 코드 작성이 아니라 시스템 전체의 안정성과 완성도를 고려하는 **개발자의 시각을 배움**

# 소감 – 황진영



황진영

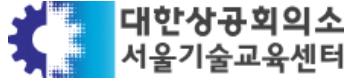
담당 :

리더 로봇

## 소감 :

- 안정적인 작동을 위한 **검증 과정이 중요하다는 것을 체감**
- 자율주행 및 군집 매커니즘 구현한 것은 **확장성과 안정성을 동시에 확보한 성과**
- **문제 정의, 검증, 개선, 테스트**를 진행하며 **완성도를 높이는 방법을 경험**
- 인텔 7기 교육을 수강하면서 펌웨어, 엣지 AI, ROS등 미들웨어, GUI에 이르기까지 전 스택을 하나의 흐름으로 연결하는 경험
- 이번 교육과정을 통해 원하는 결과를 얻기 위해 끝까지 **책임지는 임베디드 개발자**로 성장하고 싶음

# 소감 – 윤치영



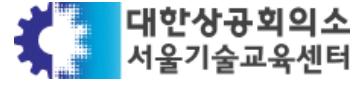
**윤치영**

담당 :

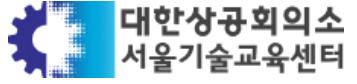
프로젝트 총괄 및 PM

## 소감 :

- 최종 프로젝트를 진행하면서 팀장의 역할의 중요성을 느낀
- 각 팀원의 성향 파악, 업무 분담, 문제 해결, 아이디어 제안, 일정 관리를 배움
- 팀원 모두 처음 접해보는 환경에 적응하며 어려운 문제를 부딪히고 해결 하며 많은 것을 배움
- 인텔 엣지 AI 과정을 통해 평소 배웠던 내용을 복습 할 수 있었고, 평소 궁금하고 배우고 싶었던 BSP를 다뤄보면서 임베디드 시스템을 이해하고 임베디드 개발자의 역량을 키워 성장함



# Q & A



# 감사합니다.