

SmartFactory

프로젝트 발표

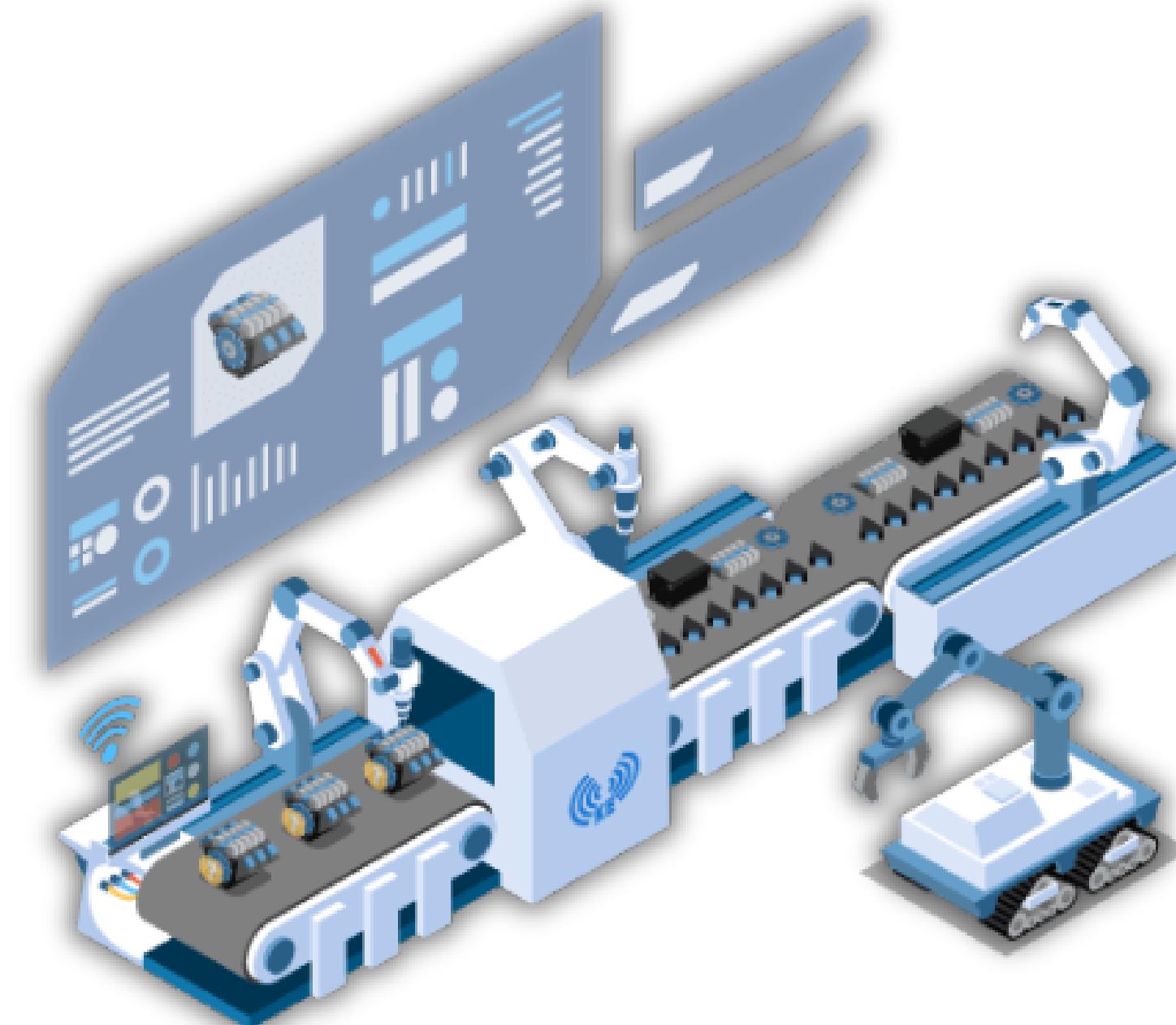
| 고의근

| 박정우

| 서창민

목 차

- 01** 프로젝트 개요
- 02** 구성도
- 03** 주요 기능
- 04** 시연 영상



프로젝트 개요

현대차 울산공장 질식 사고 발생
11월 19일 오후 3시께 공장 내 전동화 품질 사업부
차량 성능 테스트 공간(체임버)에서 연구원 3명 질식사

김영은 기자 2024.11.19

시험 운전 단새 만에 또 불...포스코 3파이넥스 공장 안전관리 '도마'

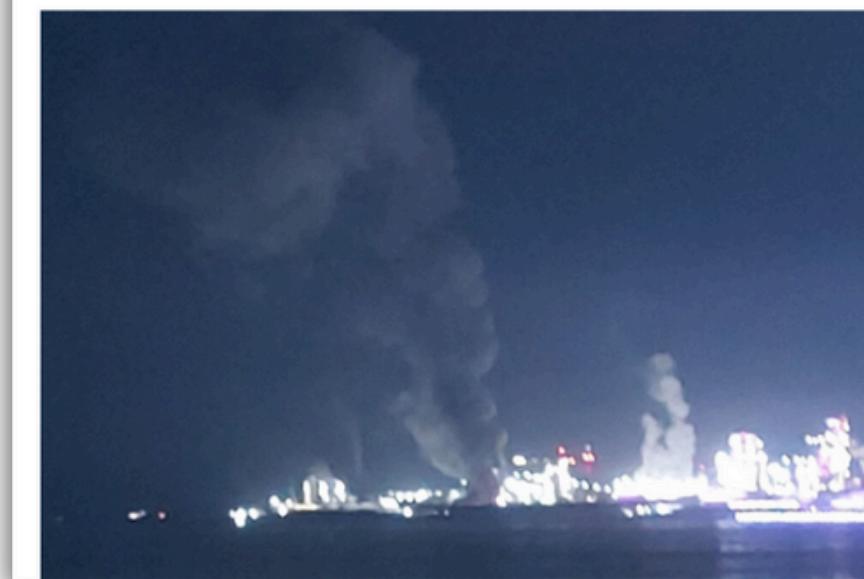
2주 새 2번 화재...'공정안전관리' 강등 이어 올해 상반기엔 과태료

주성미 기자

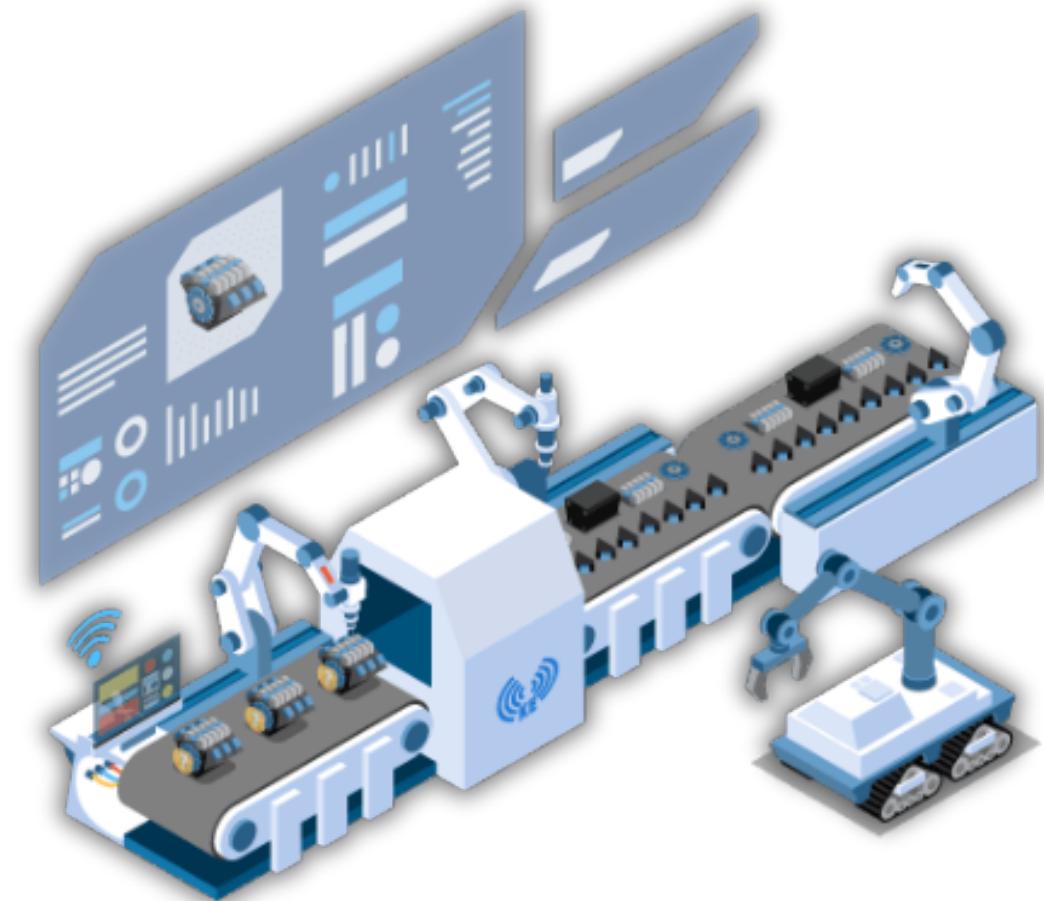
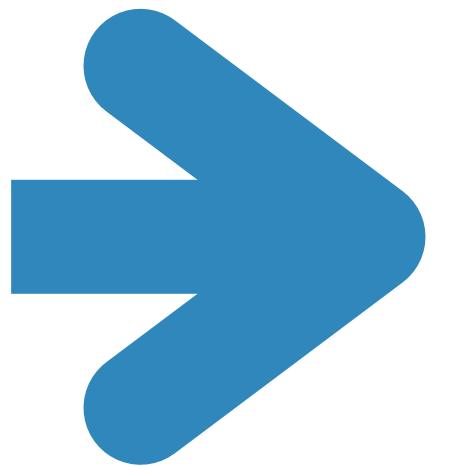
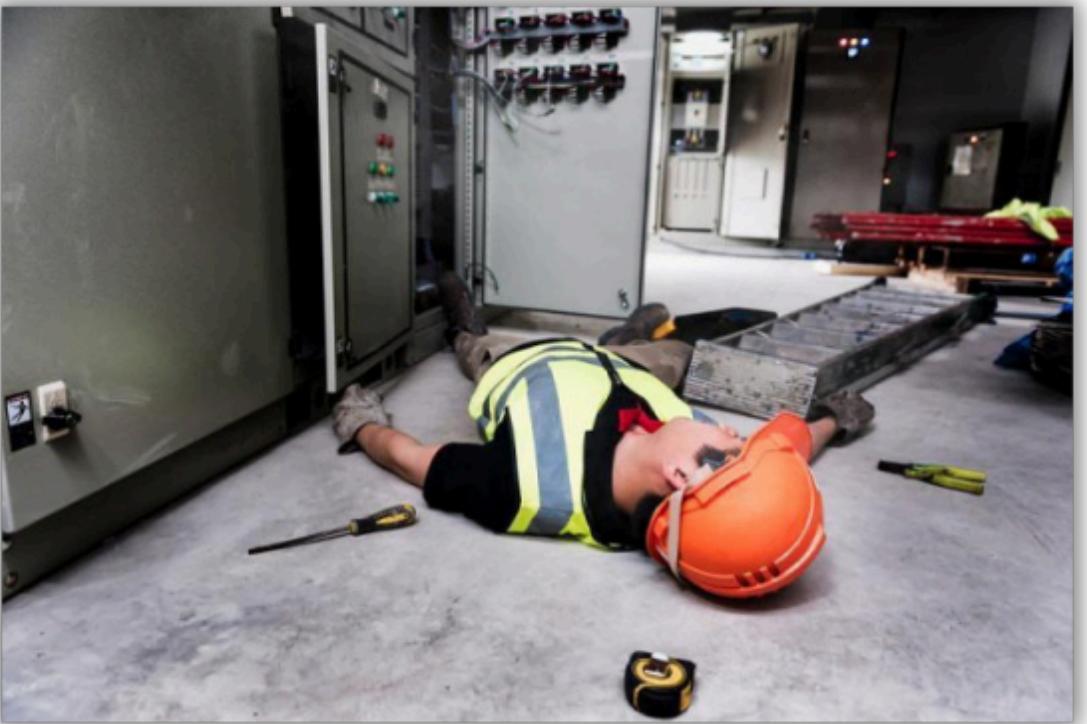
수정 2024-11-25 20:31 등록 2024-11-25 17:32

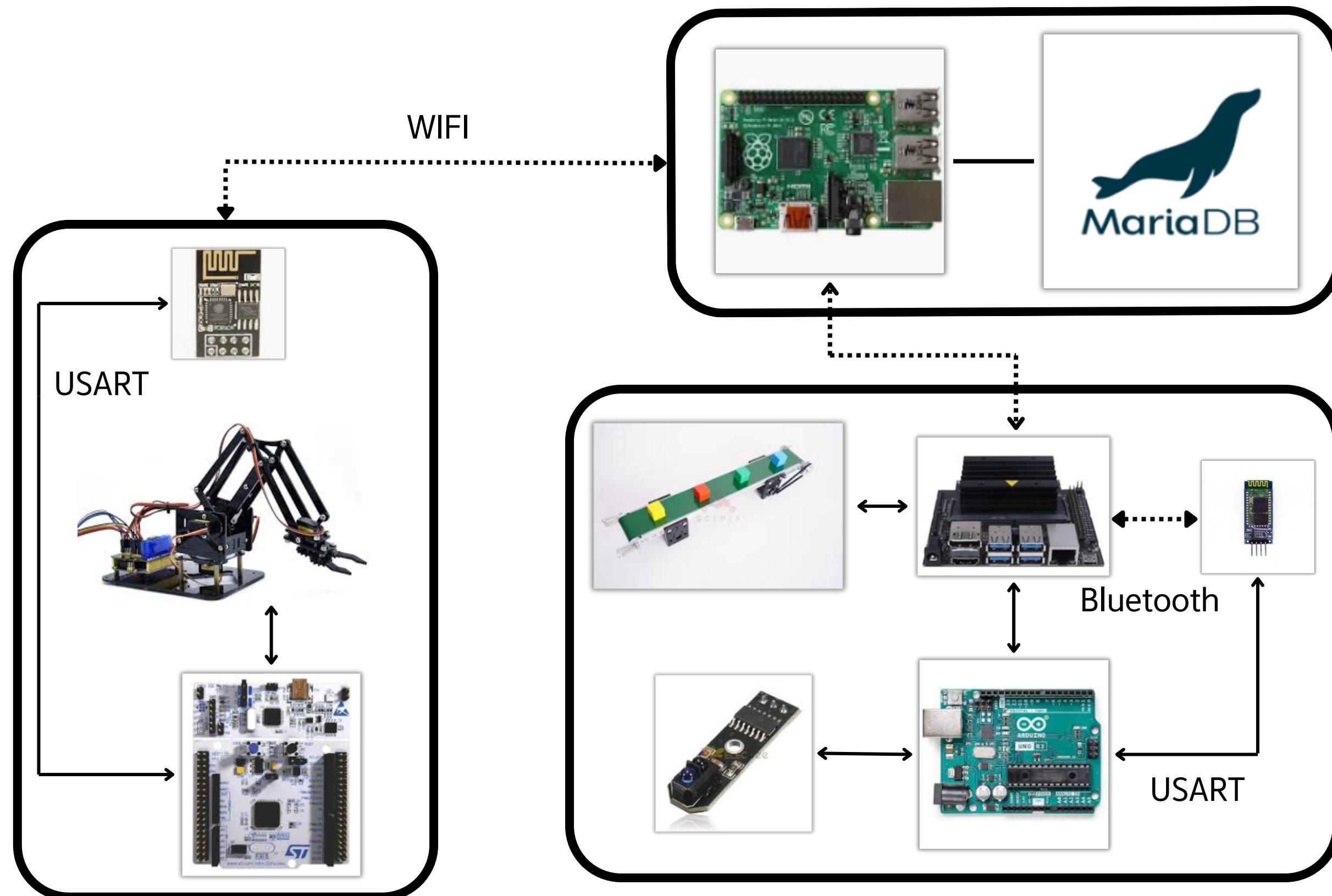
기사를 읽어드립니다

4:53 ▶ 🔍



프로젝트 개요





구성도



역할



서창민

ARDUINO 제어
컨베이어 벨트 제어
JETSON NANO IMAGE SENSING
지역 DB 관리



박정우

RASPBERRY PI 제어
전체 서버망 구축
통신 관련 규격 정의
전체 DB 관리



고의근

STM32 제어
로봇 기구학 구현
로봇 티칭

주요 기능

JETSON NANO



**1. OPENCV
HSV 기반 R.G.B 분류하여 분류 결과 DB에 저장**

**2. Raspberry pi와 연결
감지 결과 및 정보를 Main DB에 전달**

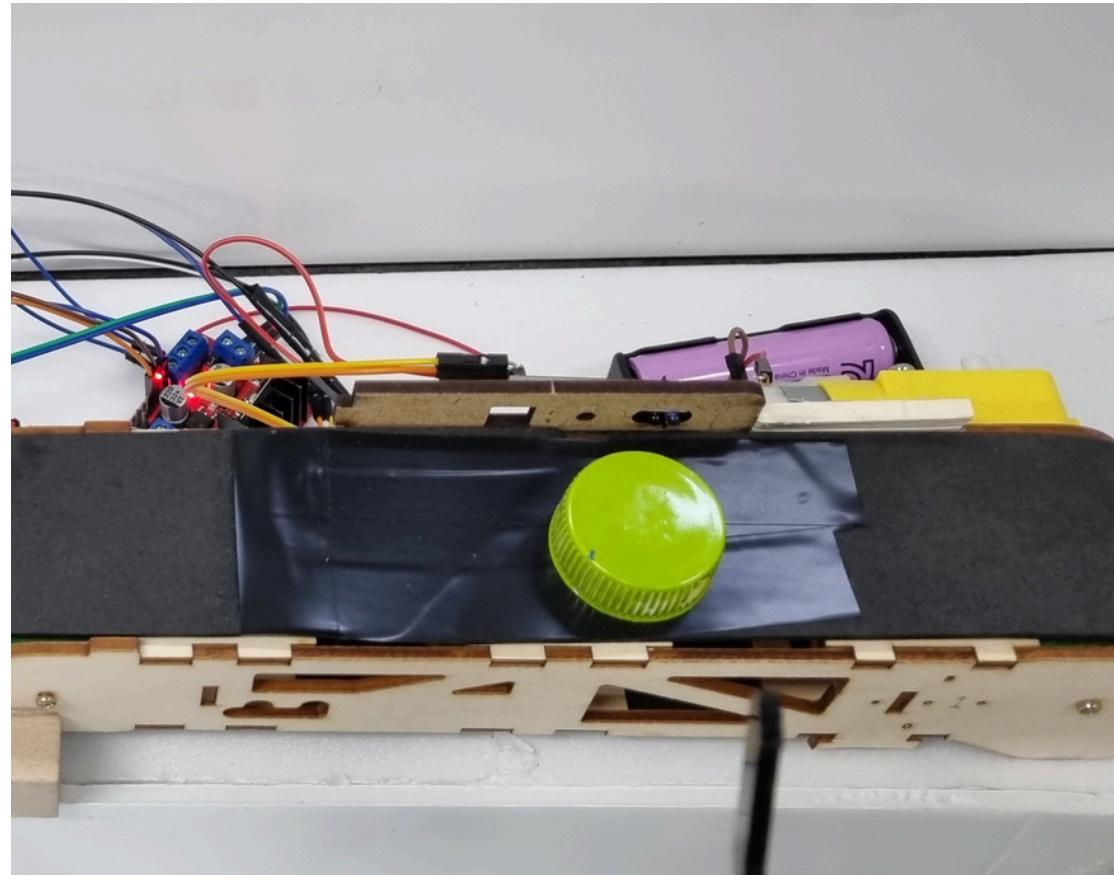
주요 기능

JETSON NANO

MariaDB [iotdb]> select * from ObjectDetection;					
ID	Color	CameraTimestamp	IRStatus	IRTimestamp	
1	green	2024-11-27 07:01:46	0	2024-11-27 07:29:53	
2	red	2024-11-27 07:01:47	1	2024-11-27 07:29:53	
3	red	2024-11-27 07:01:48	NULL	NULL	
4	red	2024-11-27 07:01:49	NULL	NULL	
5	red	2024-11-27 07:01:51	NULL	NULL	
6	red	2024-11-27 07:01:53	NULL	NULL	
7	red	2024-11-27 07:01:54	NULL	NULL	
8	red	2024-11-27 07:01:56	NULL	NULL	
9	red	2024-11-27 07:01:57	NULL	NULL	
10	green	2024-11-27 07:01:58	NULL	NULL	
11	red	2024-11-27 07:01:59	NULL	NULL	
12	red	2024-11-27 07:02:00	NULL	NULL	
13	red	2024-11-27 07:02:01	NULL	NULL	
14	red	2024-11-27 07:02:03	NULL	NULL	
15	red	2024-11-27 07:02:04	NULL	NULL	

주요 기능

ARDUINO



1. IR센서

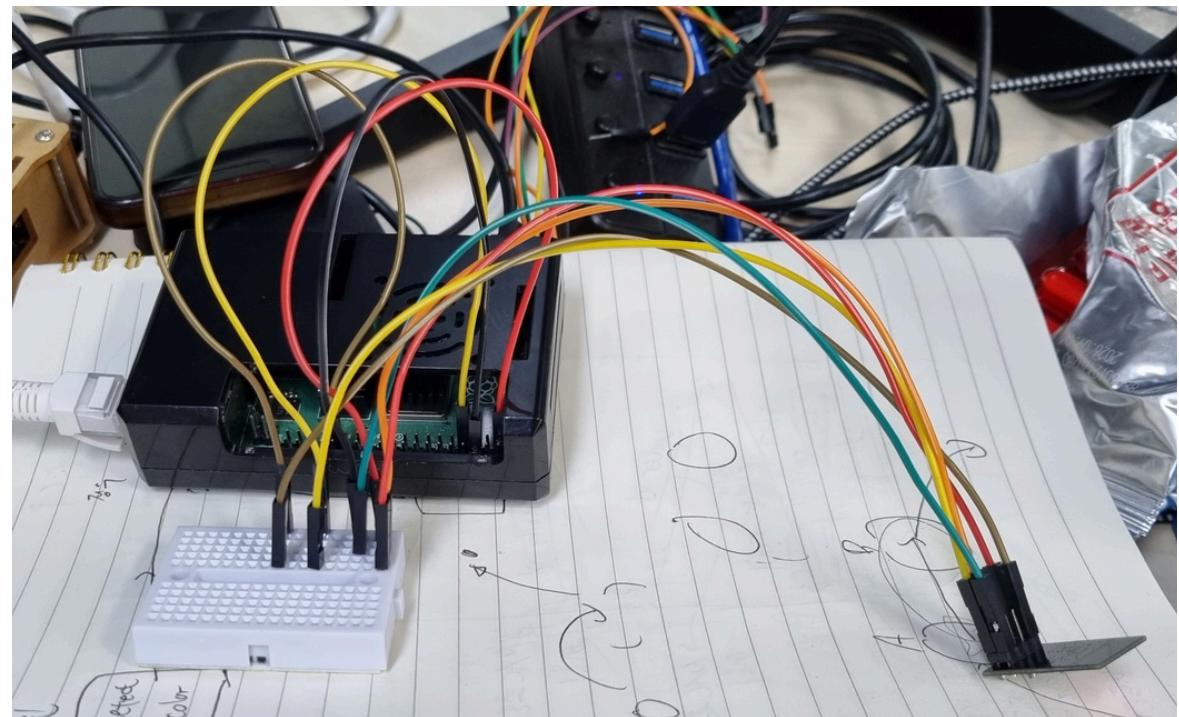
IR센서에 물체가 감지되면 DC모터 중지

2. HC-06

IR센서에 물체가 감지 → 거리로 감지 결과 전송

주요 기능

RASPBERRY PI



1. 통신망 구현

JETSON NANO와 STM32를 RASPBERRY PI에 연결

2. 데이터 짜장 및 정보 송수신

컨베이어에 검출된 결과를 로봇에 전송

3. 총괄 커버로 활용

모든 데이터가 커버에 짜장되어 활용됨.

목적 및 기대효과

RCP(Robot Control Protocol)

요구 사항

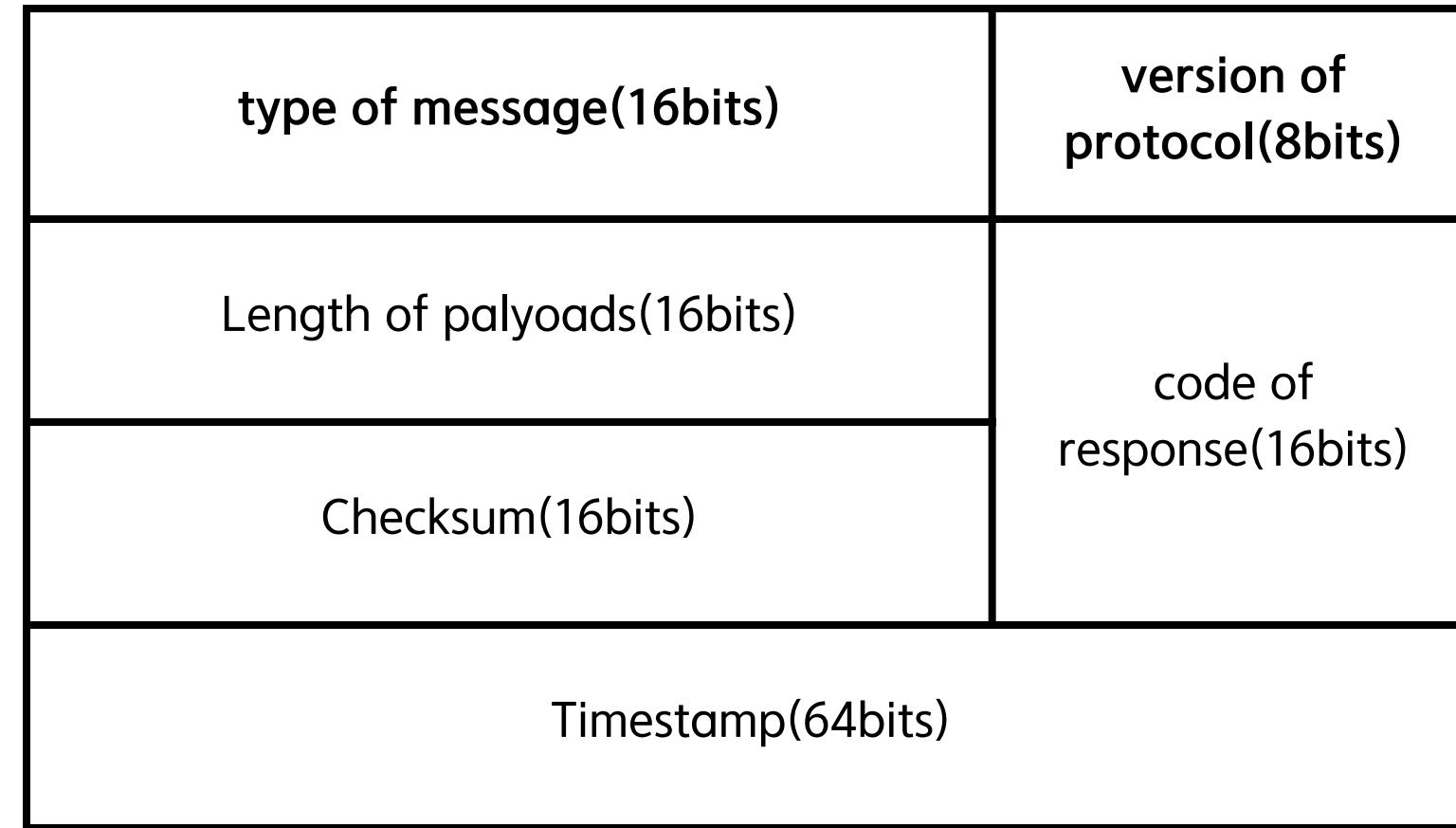
- 실시간성: 빠르고 정확한 스마트 팩토리 로봇 제어
- 동시성: 다수의 로봇 관리
- 장애 처리: 비상 상황 대처 능력

[프로토콜 구격](#)

구조의 적합성

- 고성능 비동기 처리 (epoll)
- 병렬 작업 최적화 (스레드 풀)
- 안정적 메시지 관리 (메시지 큐)

프로토콜



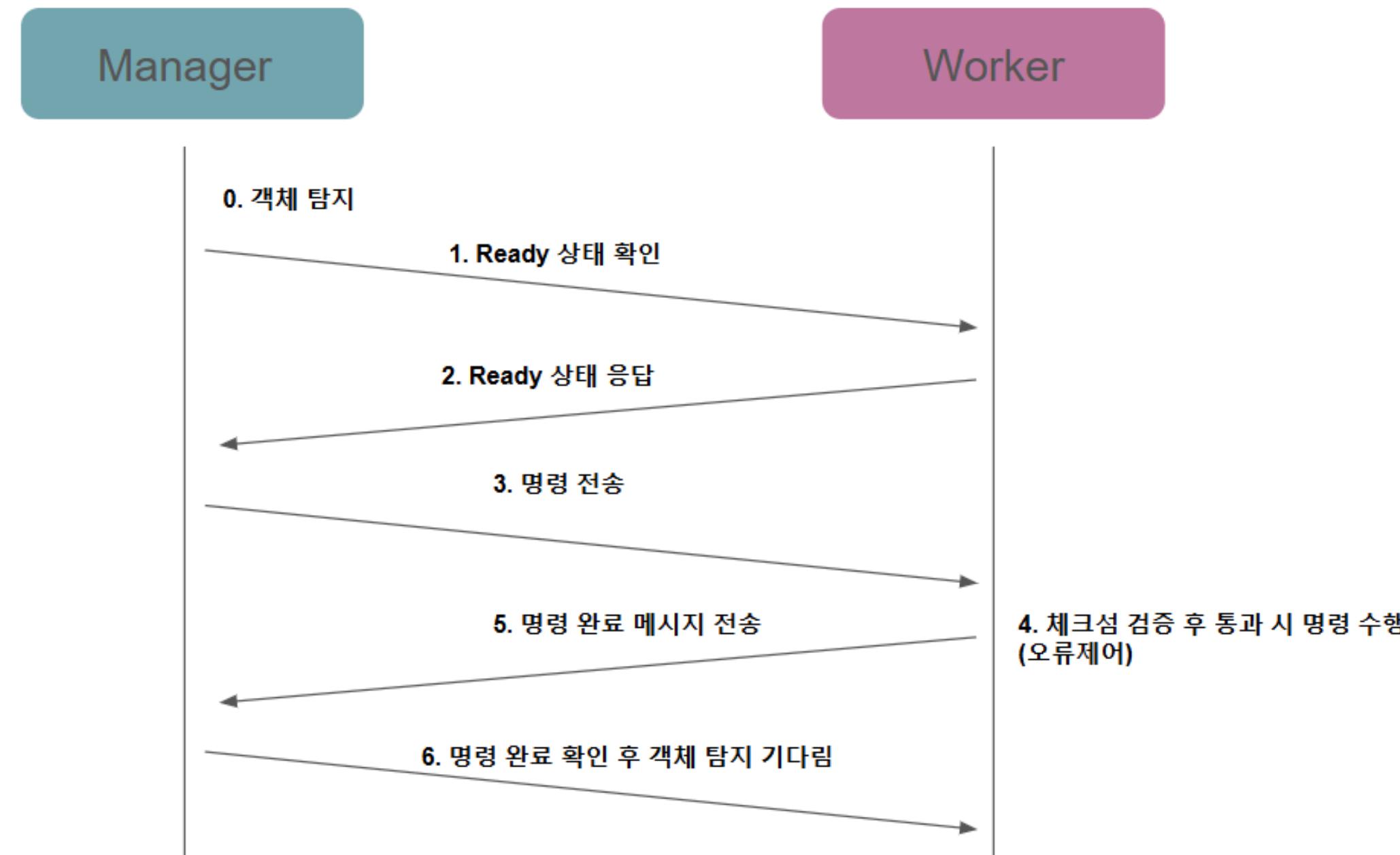
RCP(Robot Control Protocol) header

프로토콜

yaw (64bits)	pitch (64bits)	roll (64bits)
x_pos (64bits)	y_pos (64bits)	z_pos (64bits)
command (50bytes)		

RCP(Robot Control Protocol) data

프로토콜



기술 스택

- Epoll
- Pool of Posix Multi-threads
- Posix Message Queue

기술 스택

E POLL

- 고성능 비동기 I/O 처리
- 특징: 다수의 연결을 비동기적으로 처리
- 적합성:
 - 로봇 제어 명령 및 센서 데이터를 효율적으로 처리
 - 리소스 효율적: 수천 개의 장치 동시 연결 처리
 - 주의사항: 실시간 성능 최적화 필요 (지연 최소화)

기술 스택

스레드 풀(Pool of threads)

병렬 작업 처리

- 특징: 작업을 병렬로 처리해 성능 향상
- 적합성:
- 로봇의 각 제어 명령(이동, 센서 처리 등) 동시 처리
- 스레드 재사용으로 자원 낭비 감소

주의사항:

- 스레드 풀 크기 조절로 응답 시간 최적화
- 컨텍스트 전환 비용 관리

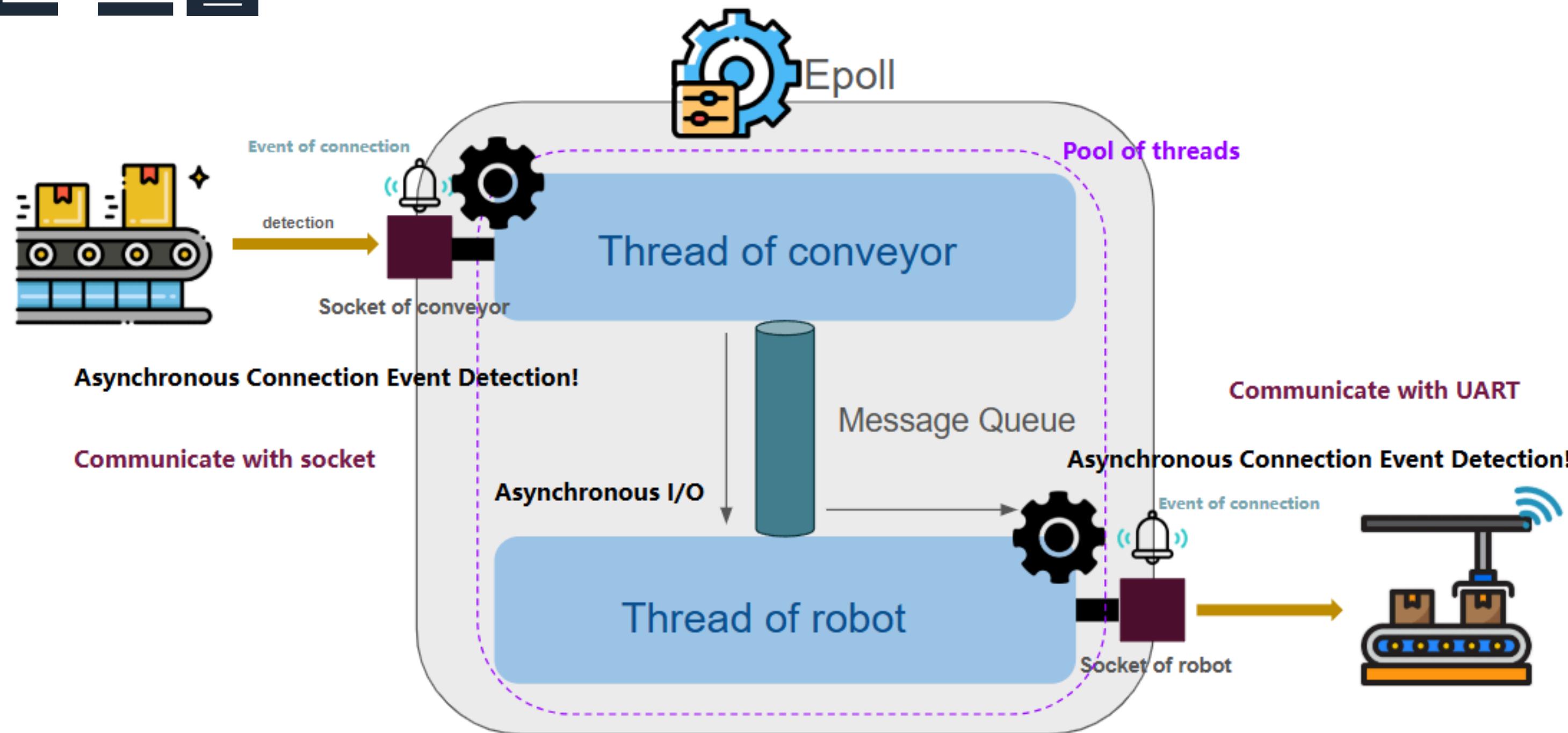
기술 스택

메시지 큐(Message Queue)

작업 큐 처리

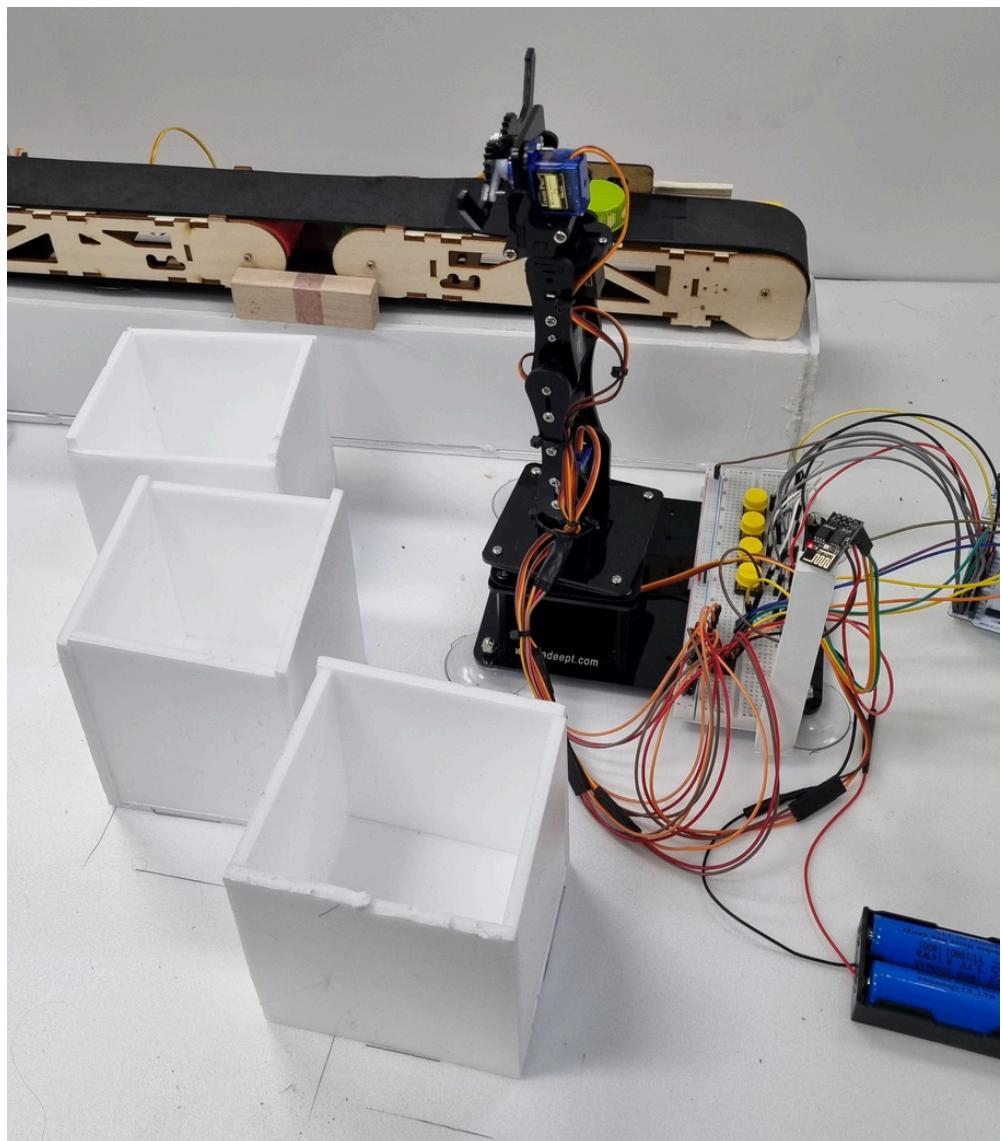
- 특징: 작업 요청을 비동기적으로 큐에 저장 및 처리
- 적합성:
 - 비동기 처리: 명령 큐를 통해 안정적 데이터 처리
 - 부하 분산: 작업 요청을 순차적으로 처리
- 주의사항:
 - 우선순위 큐로 긴급 명령 처리 보장
 - 큐 동기화 및 처리 지연 최소화

통신 흐름



주요 기능

STM32F411RE

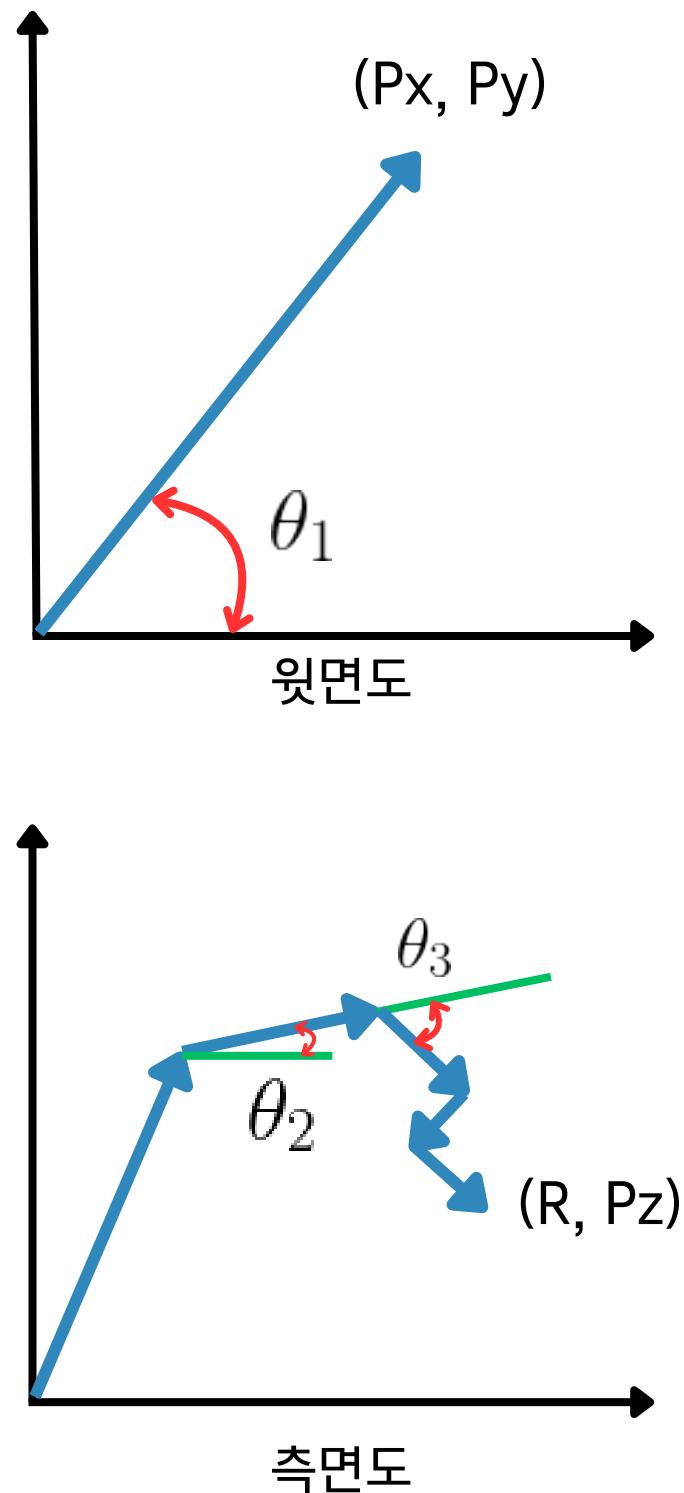
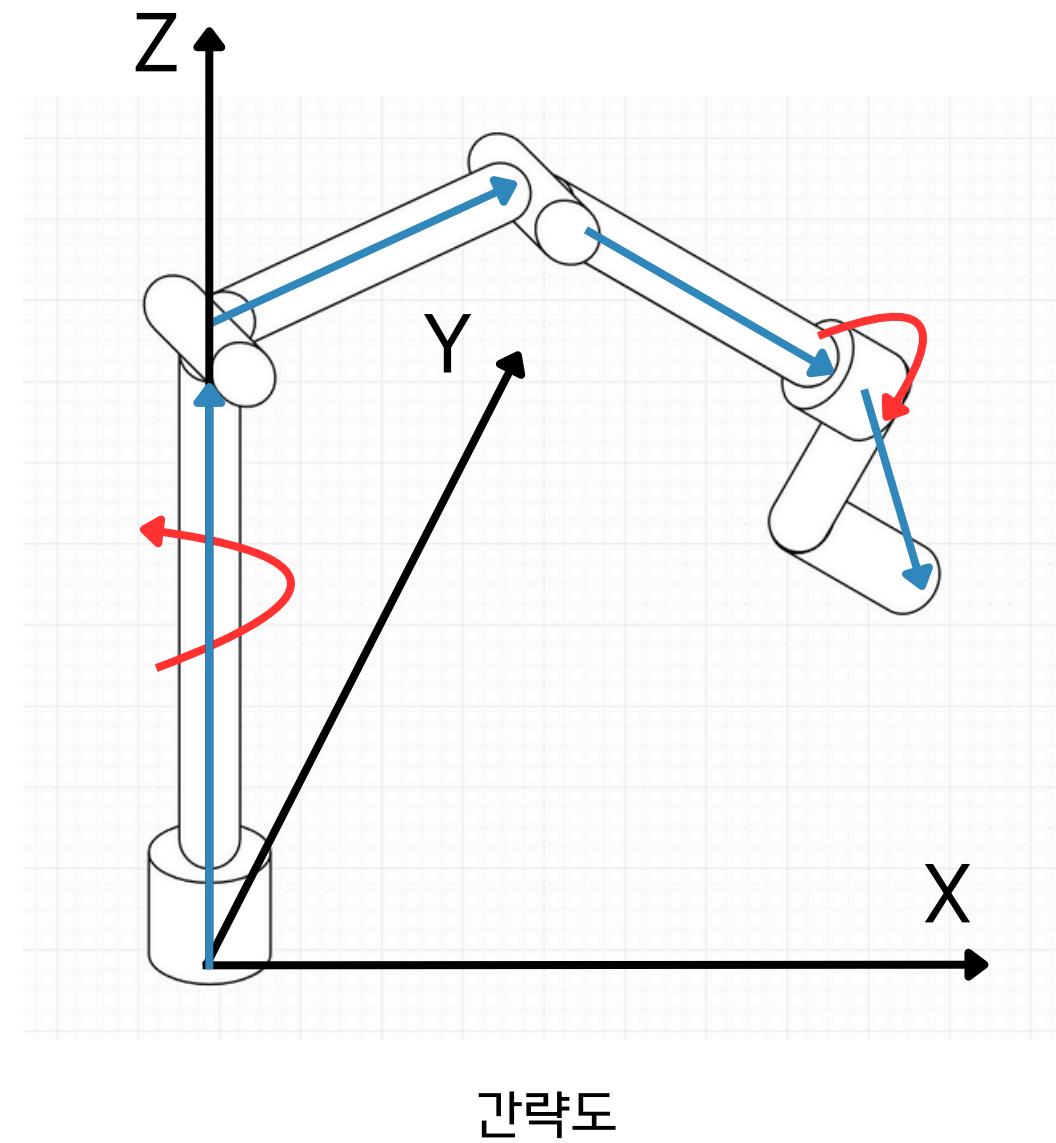


- 1. 거리에서 정보값 전송.
물체 감지 및 정보 확인**

- 2. 정보에 따라 물체를 집음.
세팅된 위치로 물체로 이동하여 그리퍼로 집음.**

- 3. 지정된 색에 따라 물체를 분류
전달받은 색 정보에 따라 물체를 분류**

기능 구현



$$\theta_1 = \tan^{-1}\left(\frac{p_z}{R}\right)$$

$$Q = \sqrt{R^2 + p_z^2}$$

$$P = \sqrt{R^2 + (d_1 - p_z)^2}$$

$$K = \sqrt{(a_3 + d_4)^2 + a_4^2}$$

$$R = \sqrt{p_x^2 + p_y^2}$$

$$\phi = \cos^{-1}\left(\frac{P^2 + a_2^2 - K^2}{2Pa_2}\right)$$

$$\gamma = \cos^{-1}\left(\frac{P^2 + d_1^2 - Q^2}{2Pd_1}\right)$$

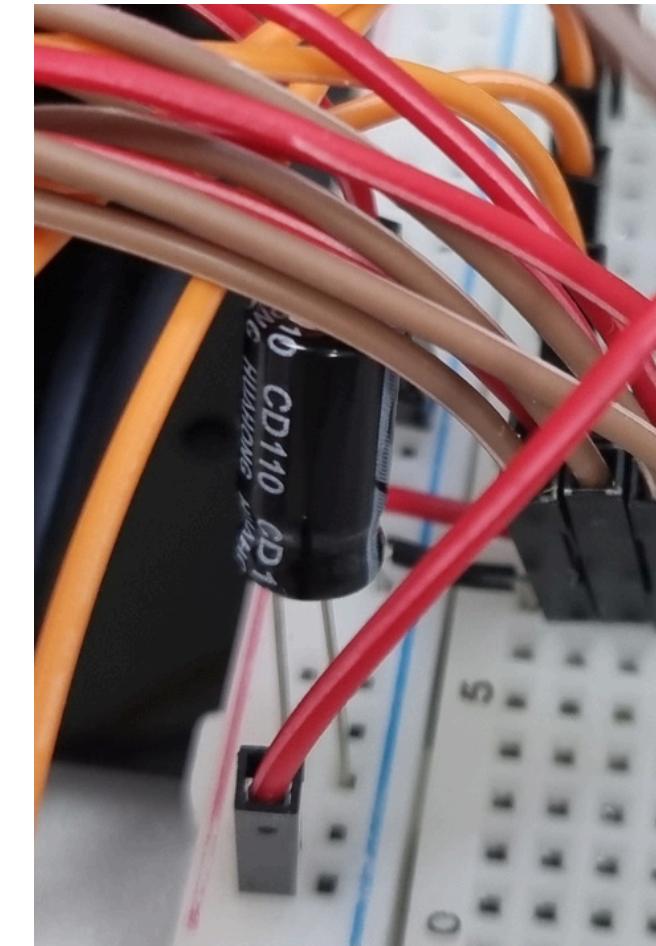
$$\theta_2 = \phi - (90 - \gamma)$$

$$\psi = \cos^{-1}\left(\frac{K^2 + a_2^2 - P^2}{2Ka_2}\right)$$

$$\beta = \tan^{-1}\left(\frac{a_4}{a_3 + d_4}\right)$$

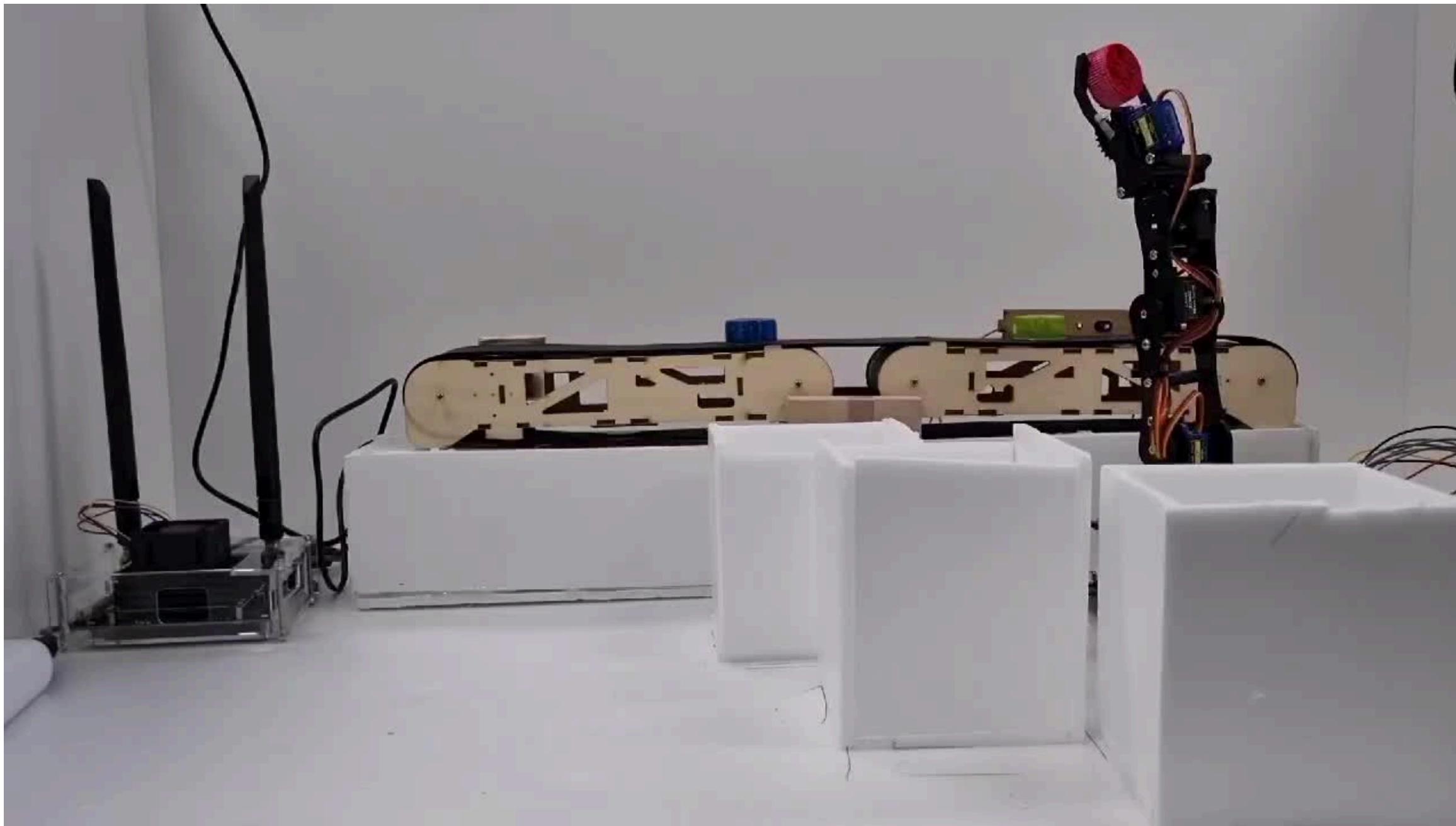
$$\theta_3 = 180 - \psi - \beta$$

기능 구현



서보 모터 5개를 안정적으로 구동하기 위하여 외부 전원을 사용 - Li 배터리 (7.4V)
전원을 안정적으로 부여하기 위한 커패시터 사용 (460uF)

영상



고찰

고의근

- 느낀점

이번 프로젝트를 통하여 STM보드에서 PWM 제어와 통신을 활용하면서 TIMER 활용 방법과 통신에 관련하여 많이 배울 수 있었다. 로봇을 제어하기 위하여 gripper의 좌표가 주어졌을 때 기구학 해석을 통하여 범용성 있게 로봇을 활용하도록 하여 공학과 소프트웨어 설계를 같이 활용할 수 있는 시간을 가질 수 있어서 뜻 깊은 프로젝트였다.

- 주의했던 점

모터의 각도 방향과 수식의 각도 방향이 서로 달라 수학식에서 나온 값과 다를 수 있어 이를 맞춰서 값을 환산할 수 있도록 하였다. 상호간의 통신을 맞춰 진행하여야 하기 때문에 대부분의 기능을 함수화하여 진행하였다.

- 아쉬운 점

서보 모터의 과열로 인하여 initial 값이 매번 달라져서 문제가 많이 발생하였다. 추후에는 과열에 신경 쓸 수 있도록 해야겠다. 서보 모터의 출력이 낮아 제대로 동작을 구현하지 못했다. 추후에 더 좋은 로봇을 사용하여 구현할 수 있도록 하겠다. 통신을 완전히 구현하지 못하여 아쉬움이 남았다. 아직 완벽하게 통신을 이해하지 못한 것 같아 추후에 통신을 이해하여 통신까지 완벽하게 구현할 수 있도록 하겠다.

고찰

박정우

- 어려웠던 점:

- 멀티 스레드 및 비동기 통신 기반의 서버 프로그램을 처음 개발했는데, 흐름 제어를 하는 것이 상상 이상으로 어려웠음
- 프로토콜 규격 설계가 생각보다 고려해야 할 점이 많아서 시간이 많이 빼았겼지만, 그 만큼 프로토콜에 대해 많은 공부가 되었음

- 아쉬웠던 점:

- STM32와 연결 시, STM32 측에서 데이터 수신을 제대로 하지 못함
- 멀티 스레드 기반의 흐름 제어로 리팩토링하면서, 몇몇 기능을 생략 함

고찰

서창민

- 어려웠던 점:

- SQL 클라이언트 메시지 수신&SQL저장과 동시에 서버로 메시지 송신 처리

- 아쉬웠던 점:

- HSV기반 색 구분이 아닌 플라스틱 종류 분류 AI 사용
 - 인터럽트 사용이 하나도 안된 컨베이어 벨트