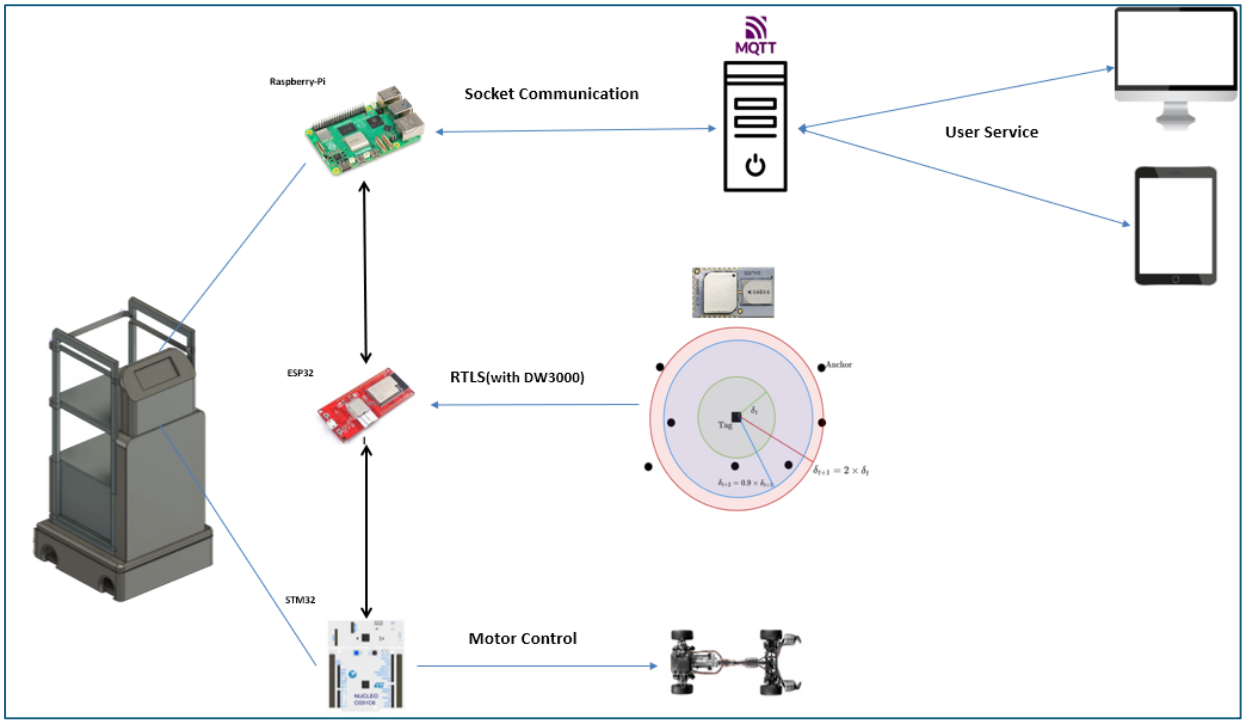
졸업과제 중간 보고서

(윤석원, 한재안, 구태헌)

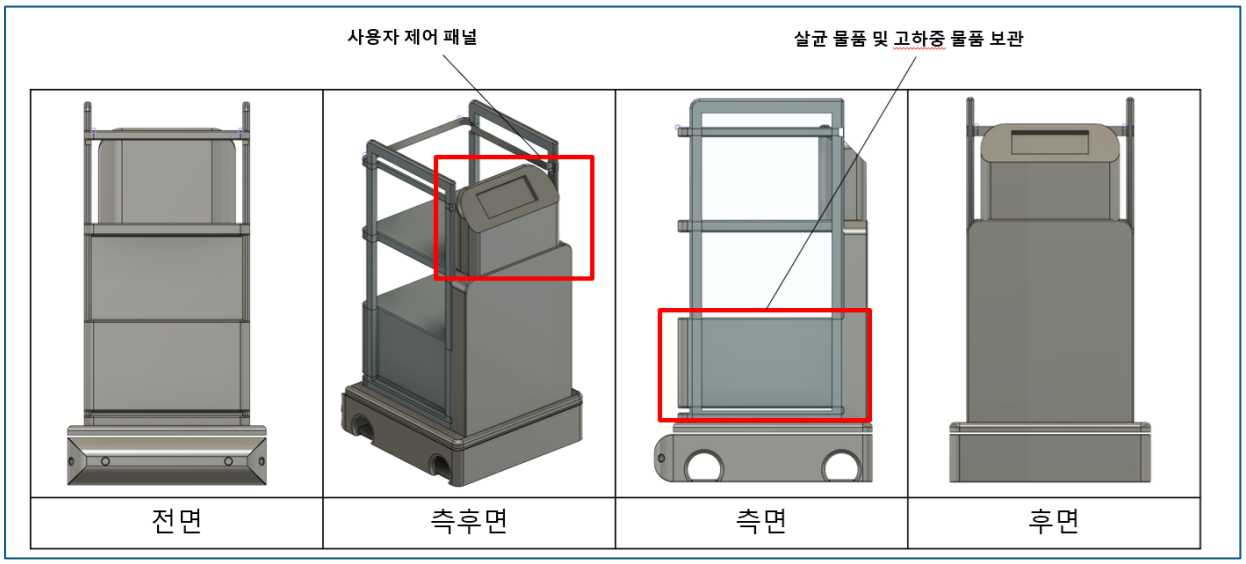
UWB 기반 실내 위치 추적과 비전기술을 활용한 로봇 운행

---

0. 시스템 전체 개요



시스템 전체 모식도



로봇 모델링

1. 요구조건 및 제약 사항 분석

초기 분석에서 제시된 요구조건과 제약 사항에 대해 다음과 같은 수정 및 보완 작업이 이루어짐.

- 카메라 및 Aruco 마커 기반 위치 및 각도 추적: 로봇의 방향을 실시간으로 추정할 수 있도록, 카메라를 통해 특정 Aruco 마커를 인식하여 위치와 각도를 추적하는 기능을 추가함. 이는 실시간 위치 제어를 위해 필수적임.

- UART 통신의 비동기 처리: 실시간 데이터 교환 요구사항을 충족시키기 위해 UART 통신의 비동기 처리가 필요했으며, 이에 따라 통신 코드를 보완함.

- 멀티스레드 환경에서의 데이터 보호: 다중 스레드 환경에서 동시 접근 문제를 해결하기 위해 세마포어를 활용한 전역 변수 보호 메커니즘을 도입함.

2. 설계 상세화 및 변경 내역

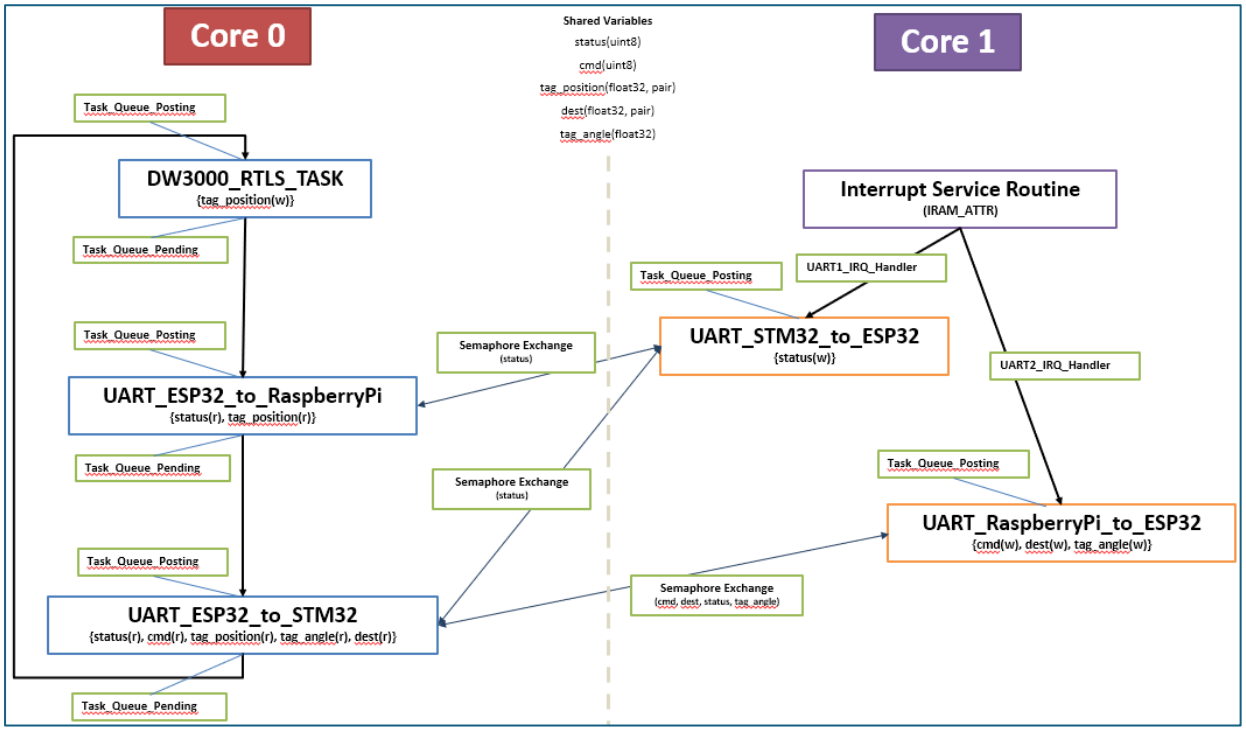
설계 단계에서 다음과 같은 상세화 및 변경이 이루어짐.

- Aruco 마커 인식 및 각도 계산: 카메라 모듈과 OpenCV를 활용하여 특정 Aruco 마커의 방향을 인식하고, Y축을 기준으로 각도를 계산하는 기능을 추가함. 이를 통해 로봇의 회전 방향을 제어할 수 있게 됨.

- UART 통신 설계 변경: ESP32와의 통신을 위한 UART 프로토콜을 설계함. 데이터를 `cmd`, `dest_x`, `dest_z`, `angle` 형식으로 패킹하여 전송하며, 수신된 데이터는 로봇의 상태 및 위치 정보를 포함하도록 설계함.

- MQTT 통신을 통한 명령어 수신: 외부 서버로부터 명령어와 도착지 정보를 수신하기 위해 MQTT 통신을 도입함. 이를 통해 로봇이 지정된 목적지로 이동할 수 있도록 설계함.

3. 로봇 펌웨어 개발



RTOS Task Diagram

로봇 제어 모듈은 크게 Raspberry Pi, ESP32, STM32 세 가지로 나누어짐. 각 모듈의 역할은 다음과 같음.

- Raspberry Pi: 소켓 통신을 통해 사용자 명령과 로봇의 목적지를 입력받음.
- ESP32: DW3000과 SPI 통신으로 연결되어 UWB 기반의 RTLS(실시간 위치 추적 시스템)를 수행하며, 로봇의 움직임에 대한 종합적인 판단을 STM32로 전송함. 이때, RTOS 기반의 이벤트 드리븐(Event-Driven) 방식으로 각 신호를 처리해 높은 반응성을 확보함.
- STM32: ESP32로부터 로봇 움직임에 대한 명령을 수신하여 모터, 근접 센서, 가속도 센서 등 다양한 하드웨어를 종합적으로 제어하여 실질적인 로봇 움직임을 구현함.

4. 개인 진행도

- DW3000 UWB 모듈 기반 RTLS 구현 완료
- UART 이벤트 드리븐 방식을 기반으로 한 세 가지 모듈(STM32↔ESP32↔Raspberry Pi) 간 연결 완료
- 기본적인 로봇 RTOS 펌웨어 구현 완료
- 로봇 제어부 구현 완료
- 로봇 동작에 필요한 기본 회로 구성 완료

5. 갱신된 과제 추진 계획

프로젝트 진척도에 따라 과제 추진 계획을 다음과 같이 갱신함.

- 최종 시스템 통합 및 테스트 (예정): 모든 기능을 통합하여 로봇이 실시간으로 목적지를 설정하고 이동할 수 있도록 최종 시스템을 구성한 후, 실제 환경에서 테스트할 계획임.

6. 보고 시점까지의 과제 수행 내용 및 중간 결과

현재까지의 진행 상황과 중간 결과는 다음과 같음.

- Aruco 마커 인식 및 각도 측정: 카메라를 통해 특정 Aruco 마커를 인식하고, 로봇의 현재 방향을 계산하는 기능을 구현함.
- UART 통신 구현: ESP32 와의 안정적인 비동기 통신을 통해 로봇의 상태와 위치 정보를 주고받는 기능을 구현함. 데이터 패킹 및 언패킹이 정상적으로 이루어지고 있으며, 통신 안정성을 위한 최적화가 진행 중임.
- MQTT 통신을 통한 명령어 수신: 서버로부터 실시간 명령어(cmd)와 목적지 정보(dest_x, dest_z)를 수신해 로봇의 이동 경로를 설정하는 기능을 구현함. 멀티스레드 환경에서의 안전성을 보장하기 위해 세마포어를 활용하고 있음.
- 현장 비트맵 표현: 비트맵 표현을 통해 로봇의 이동 가능한 영역과 불가능한 영역을 표시하고, 로봇의 현재 위치와 도착지 정보를 주고받는 기능을 구현함.

7. 사용자 상호작용 파트

- 웹의 역할: 사용자(간호사)는 웹을 통해 다양한 메뉴와 옵션을 통해 로봇 관련 명령을 요청할 수 있음.
- 서버의 역할: 사용자(간호사)는 서버에 HTTP 로 요청을 보내고, 서버는 로봇에게 MQTT 로 요청을 전달함. 예를 들어, 간호사가 13:00 에 로봇 A 를 201-6518 로 이동시키고자 하는 요청을 보내면, 서버는 201-6518 주소를 비트맵 좌표로 변환해 로봇 A 에게 변환된 좌표(예: 0, 52, 84)와 커맨드(GO)를 전송함.

8. 문제점 및 해결 방안

- RTLS 오차 문제: LOS(직선 시야) 상황에서는 10cm 미만의 위치 오차를 보이지만, NLOS(비직선 시야) 상황에서는 최대 30cm 까지 오차가 발생함.
- 통신 데이터 밀림 문제: 각 모듈 간 비동기 통신에서 데이터 밀림으로 인한 잘못된 값 전송 문제가 발생함. 이를 해결하기 위해 CRC(Cyclic Redundancy Check)를 적용할 예정임.
- MQTT 환경 구축 문제: 동일 네트워크 내에서 다른 기기 간의 MQTT 구독 및 발행 환경을 구축할 때 특정 공유기를 사용해야 하는 문제가 발생함. 이를 포트 포워딩으로 해결할 계획임.

- 서버 -> 로봇 간 단방향 통신: 현재 서버에서 로봇으로의 단방향 통신만 구축된 상태이며, 이를 개선하기 위해 로봇이 다른 토픽을 사용해 구독 및 발행 구조를 변경하는 방안을 검토 중임.

9. 착수 보고서 피드백

기존 서빙 로봇과의 차별점 설명 부족: 착수 보고서 피드백에서 기존 서빙 로봇과의 차별점이 충분히 설명되지 않은 점이 지적됨. 기존 서빙 로봇은 천장에 부착된 고유 패턴을 사용해 위치를 추정하는 방식으로, 넓은 공간보다는 좁은 공간(예: 식당)에 적합함. 반면, 이번 프로젝트에서 사용한 UWB 기반 위치 추정 방식은 공간이 넓어지더라도 추가적인 UWB 앵커와 동일한 패턴의 마커만 설치하면 되므로, 병원과 같은 넓은 공간에 적합함.