[5G 자율 무인이동체 시스템 개발자 양성과정]

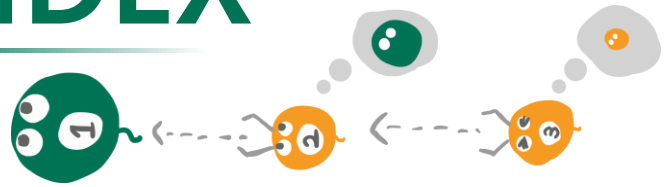
무인이동체 자율주행 시스템 개발 프로젝트

이미지인식을 활용한 AGV 자율주행 시스템

서재원 김연재 이주혁

2021.06.25

INDEX



1 팀원 소개

김연재 Project Manager

- 소프트웨어 엔지니어링
- 이미지 인식(Lane, Color Detection) 기능 구현

서재원

- 하드웨어 엔지니어링
- 그리퍼 기능 구현

이주혁

- 소프트웨어 엔지니어링
- 마커 인식, 로봇 컨트롤 기능 구현

2 프로젝트 일정

1주차
주제선정 및 계획수립

2-3주차
1차 기능 개발
- Lane Tracking
- Marker Tracking

4주차
1차 기능 테스트

5-6주차
2차 기능 개발
- Carry out Mission
- Color Detection
- Parking

7주차
통합 테스트

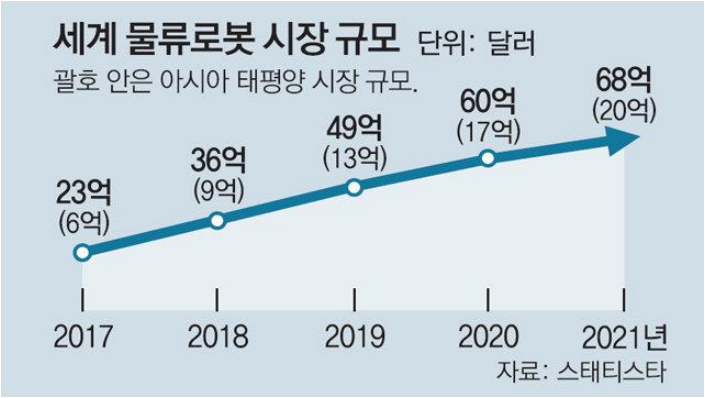
8주차
최종수정 및 발표

3 주제선정 배경 및 목적

- **AGV(Automated Guided Vehicle)란?**
  - 공장이나 창고에서 주로 활용되는 **무인운반차**(물류로봇)
- **주제선정 배경**
  - IFR(국제로봇연맹)에 따르면, 전체 업무용 서비스 로봇 시장의 약 **45%**를 **물류로봇**이 차지함
  - 코로나 팬데믹 상황으로 인해 언택트 기조 확산으로 물류로봇시의 폭발적인 성장이 예상됨
  - 그동안 제조·유통부문에서 단순 반복작업에 그쳤지만, 4차산업혁명 기술의 발달로 물류로봇의 자율적인 상황 대처가 가능해짐
- **프로젝트의 목적**
  - 로봇 간의 통신을 통해, **업무 수행을 자동화할 수 있는 시스템 개발을 목표**로 함

국내 물류로봇 시장 실적 및 전망

자료출처 : 중기부



4 개발환경 및 사용기술

개발환경

- **Remote PC**
  - Ubuntu 16.04 LTS Desktop
  - ROS Kinetic Kame
- **Turtlebot3** (Raspberry Pi 3B+)
  - Raspbian GNU/Linux 9 (stretch)
  - ROS Kinetic Kame

- **개발 언어**
  - Python2.7

구현기술

- **카메라**와 **AR마커**를 활용한 **객체탐지** 및 **트래킹**
- **라이다(LDS)**를 이용한 **주변거리 정보 인식**
- **영상처리(OpenCV)**를 통한 **차선** 및 **색 정보 인식**
- **오픈소스 하드웨어 플랫폼** 간 **시리얼 통신**
- **상호 간 통신**을 활용한 협업

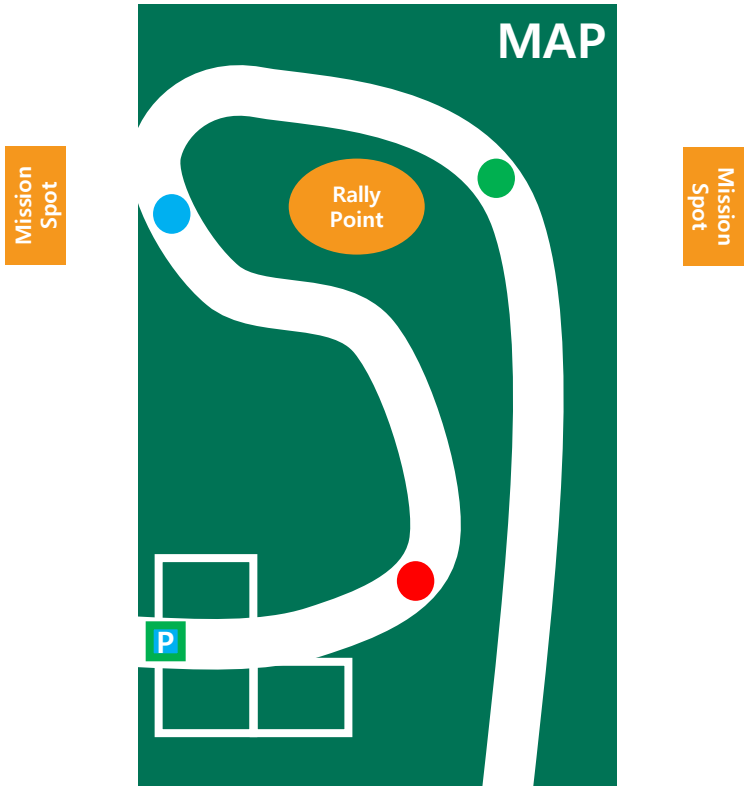
5 시나리오

① Lane & Marker Tracking

② Color Detection

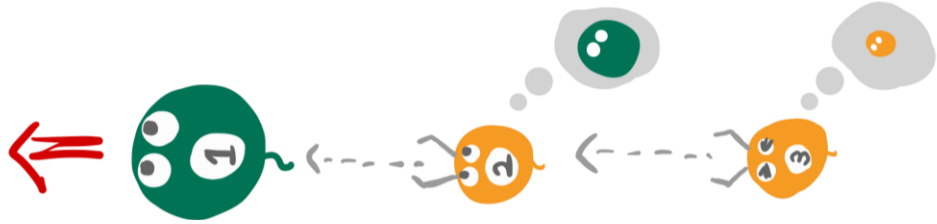
③ Parking

5 시나리오 ① Lane & Marker Tracking



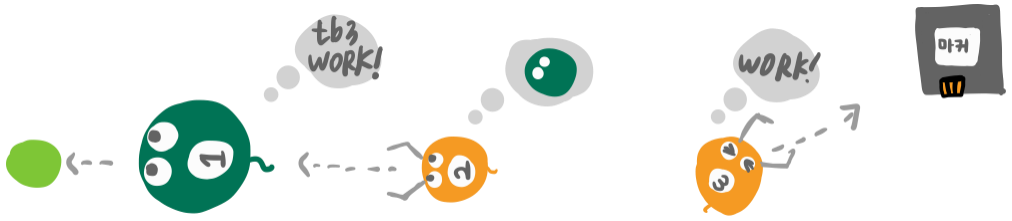
1) 앞에 위치한 터틀봇의 마커를 찾아 정렬

2) 정렬 완료 후, 1번은 Lane Tracking, 2/3번은 Marker Tracking을 수행

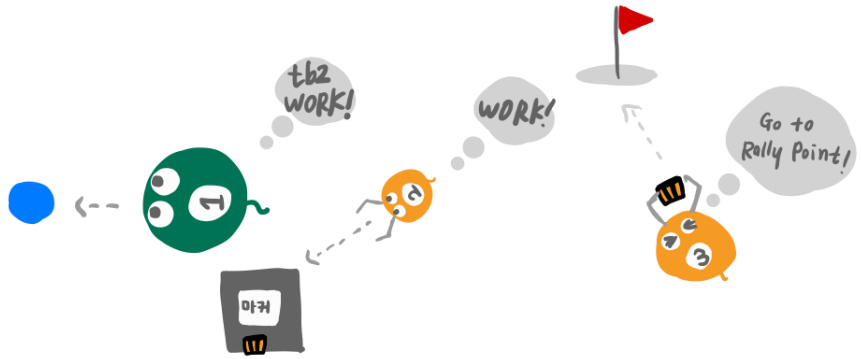


5 시나리오 ② Color Detection

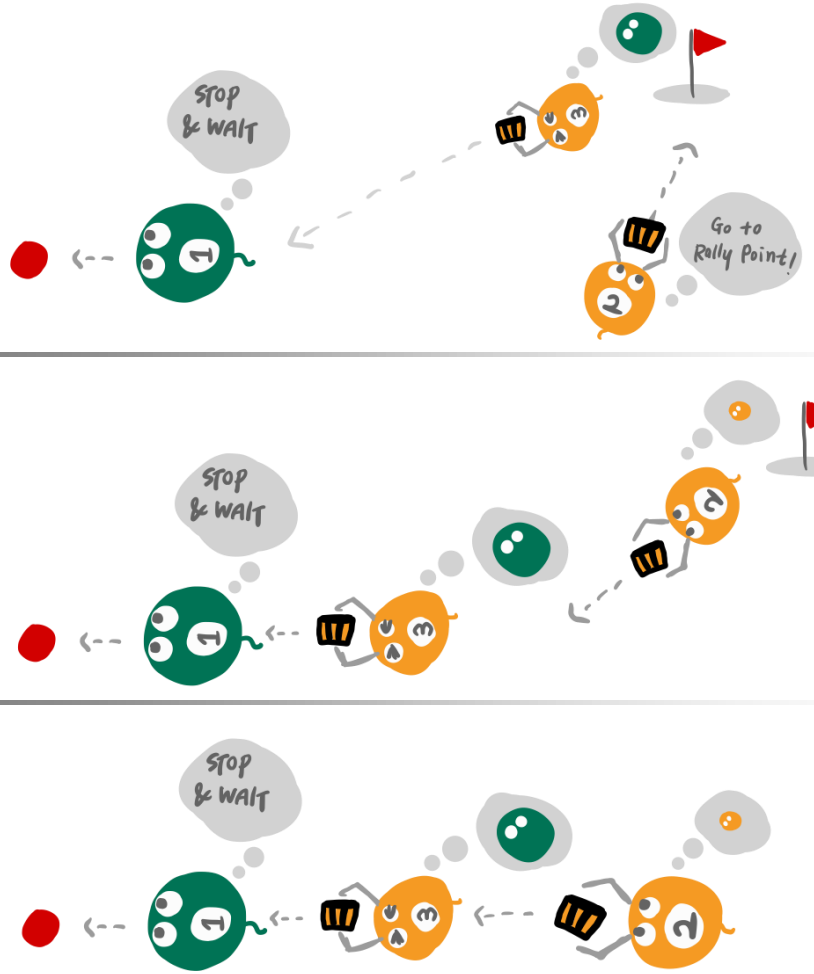
1) 초록색을 탐지하면 tb3가 미션을 수행



2) 파란색을 탐지하면 tb2가 미션을 수행



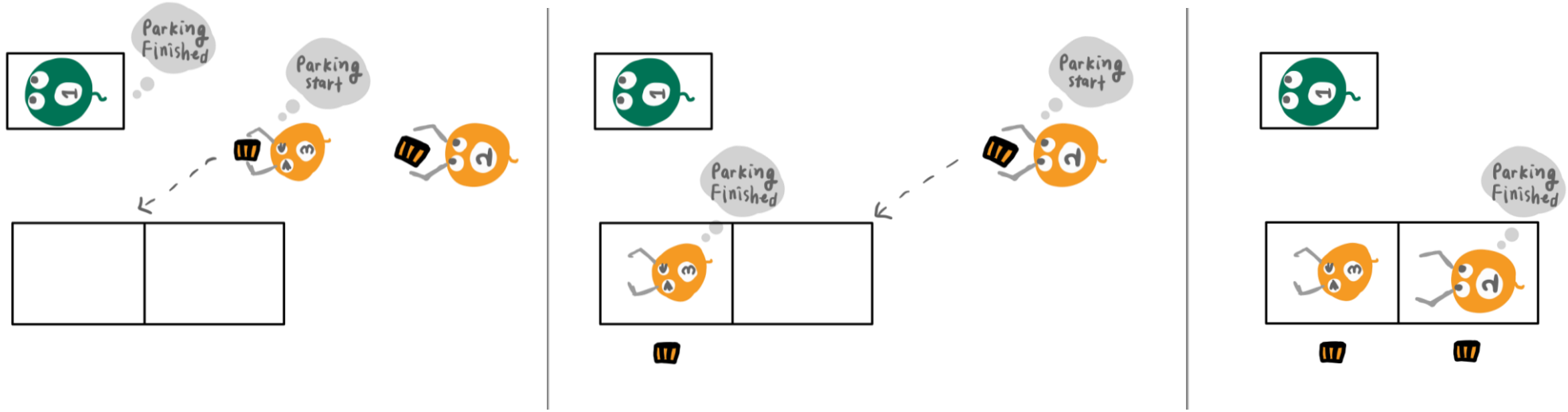
3) 빨간색을 탐지하면 tb1은 모두 정렬될 때 까지 대기



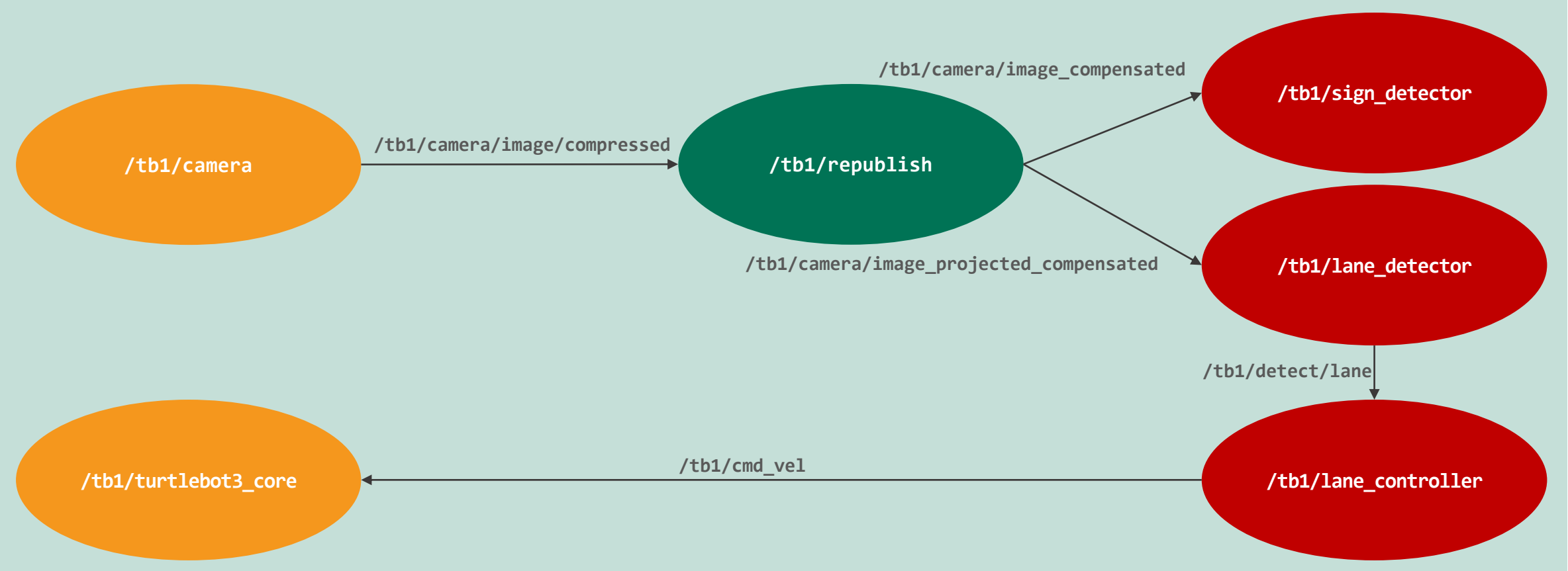
5 시나리오 ③ Parking

1) 주차표시를 탐지하면 tb1 주차 수행

2) tb1이 주차를 마치면 정해진 위치에 물건을 내려두고 주차 수행

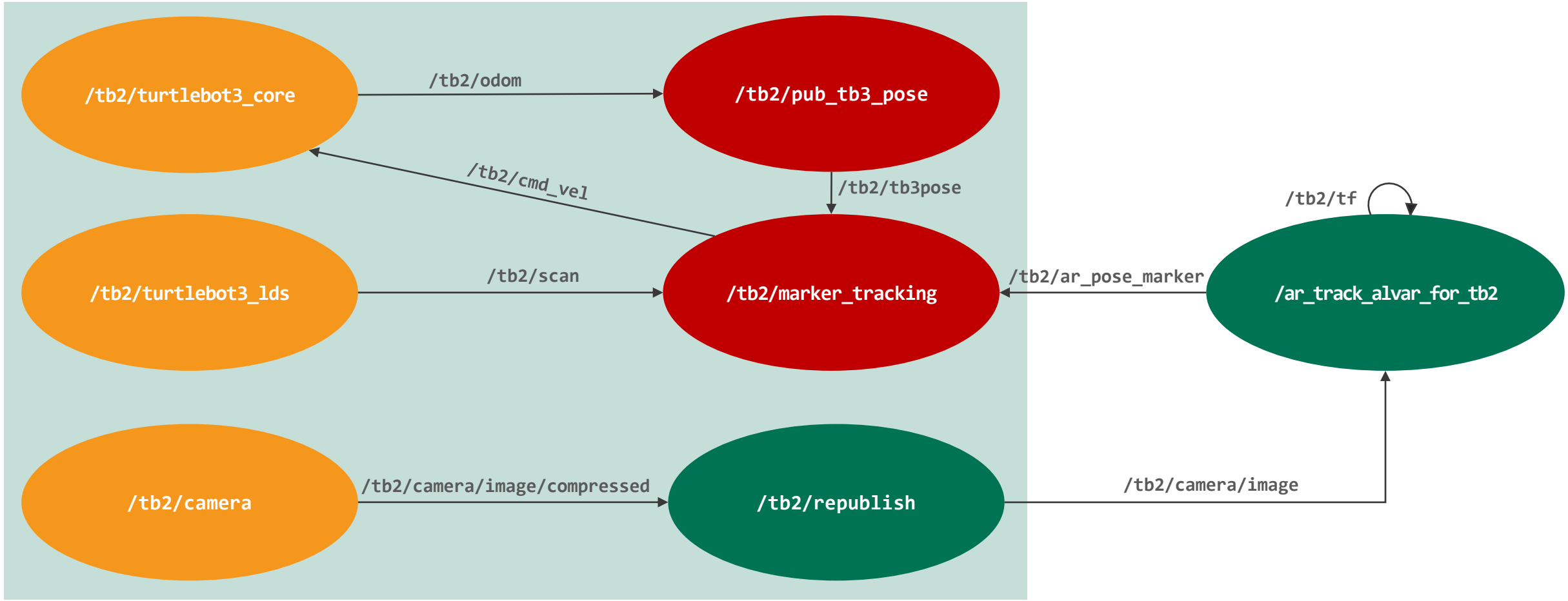


6 노드 구성 (Leader)




```
$ roslaunch turtlebot3_bringup turtlebot3_bringup.launch
$ roslaunch turtlebot3_autorace turtlebot3_autorace_camera_pi.launch
$ roslaunch agv_autorace leader_camera.launch
$ roslaunch agv_autorace leader_tracking.launch
```

6 노드 구성 (Followers)




```
$ roslaunch turtlebot3_bringup turtlebot3_bringup.launch
$ roslaunch turtlebot3_autorace turtlebot3_autorace_camera_pi.launch
$ roslaunch agv_autorace follower.launch tb_num:=tb2
```

7 시연영상

8 후기

- **개발후기**
  - 마커의 최대인식거리가 0.5미터 정도로 짧아서 최대인식거리 이상일 경우 인식이 불가능한 경우가 발생
    → 마커 두개를 사용하여 해결
  - 개발 수행 시 각각의 roscore에서 개발을 진행한 후 하나의 roscore로 통합하는 중에 토픽명의 중복 등의 문제 발생
    → Remap 등의 방법을 통해 문제 해결
  - 고성능의 하드웨어를 사용했을 시 발생하지 않을 문제들이 발생
    → 하드웨어 관리 및 코드를 경량화하여 해결
- **후속연구**
  - 이후 확장성을 고려하여 멀티 마스터 환경으로 구현

Q&A

감사합니다.