산학프로젝트 주간 보고서 ( 2019. 06. 13)

프로젝트명 : AI planning

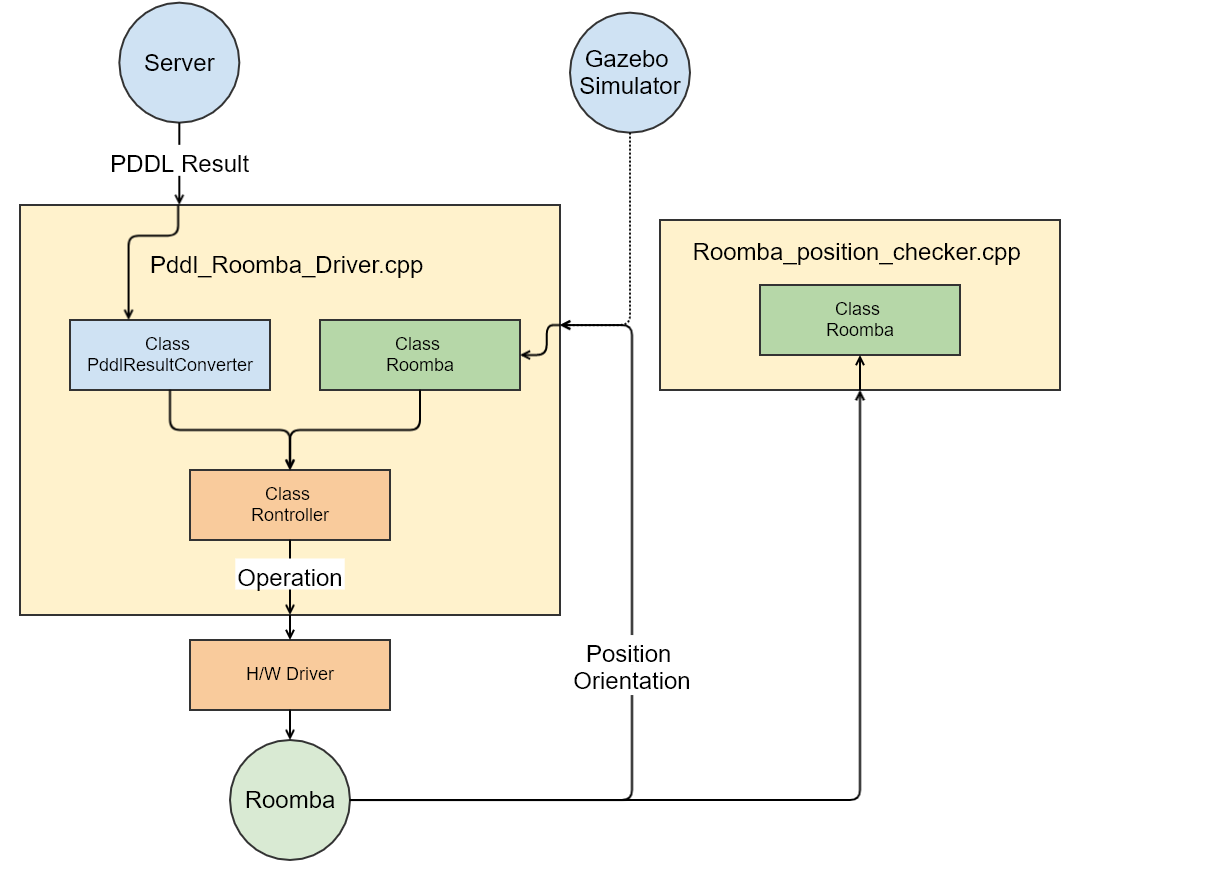
이름/팀명 장환석 /AP

**1.개인별 주간 목표(상세 개발 계획서)**

|  |  |
| --- | --- |
| 14주차 | 1. 다중 룸바 제어 구현 및 테스트 |
| 15주차 | 1. 룸바 Parameter값 최적화 |

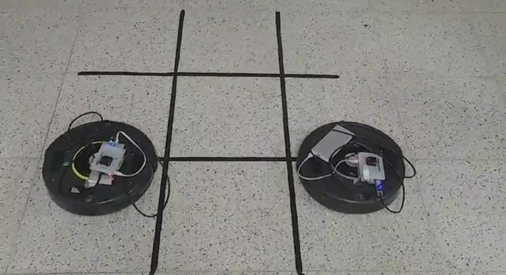
**2.+추진 실적**

**<다중 룸바 멀티 쓰레드 구현 및 테스트>**

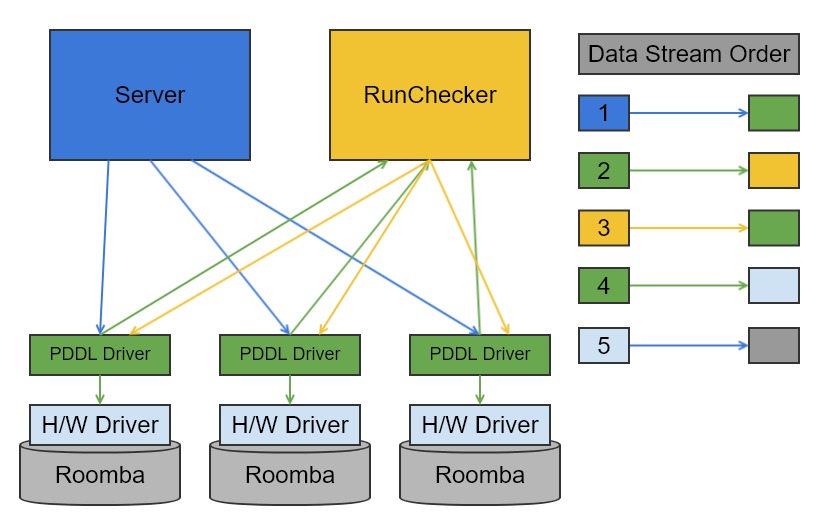


저번주에 구상했던 구조로 Class들을 재구성하여 테스트 해보았습니다

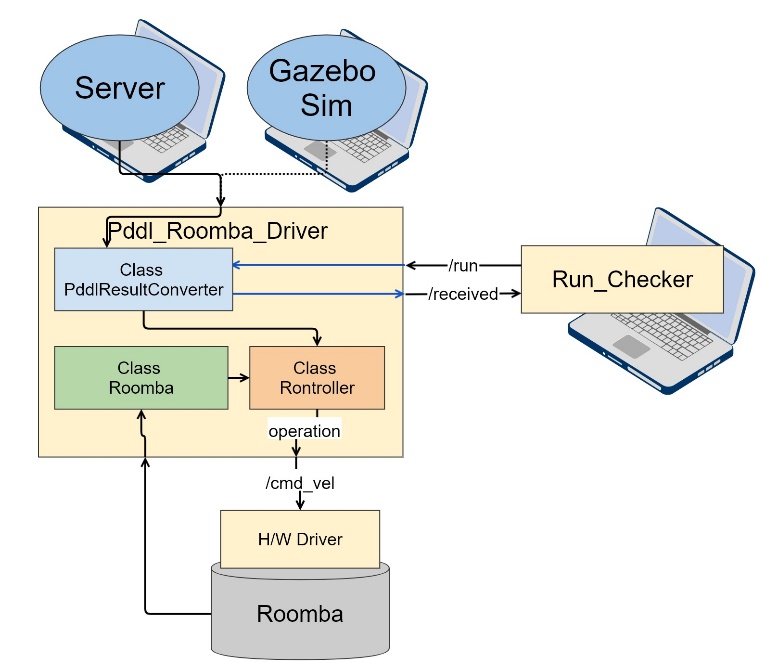
다중 룸바를 순차적으로 움직이지 않고, 동시에 움직이는 것은 성공하였으나 서로의 명령어가 혼선되는 문제가 발생했습니다. 결과적으로 다수의 룸바에게 같은 명령을 내릴 때에는 정상적으로 작동하지만, 서로 다른 명령을 수행해야할 때는 하나의 룸바만 명령을 수행하였습니다.

****

이를 해결하기 위해서 ROS에서 Multi-Thread의 개념이 Call-back함수로 구현되었다는 점을 착안하여 아래와 같이 구조도를 재구성하였습니다.

위 구조는 크게 다섯 번의 데이터 통신을 통해서 멀티 쓰레딩을 구현합니다.

1. 서버에서 각룸바에게 명령을 전송합니다.
2. 각 룸바는 자신이 받은 명령을 전저리하면서 RunChecker에게 자신이 명령을 받았다는 것을 알립니다.
3. RunChecker는 모든 룸바에게 명령을 받았다는 신호를 수신하면, 전체 룸바에게 명령을 수행하라는 명령을 내립니다.
4. PDDL\_Driver는 연산을 후처리하여 자신의 바퀴를 제어하기 위한 신호를 전송합니다.
5. 이를 H/W 드라이버가 수신하여 룸바를 움직입니다.
6. 4.~5.번의 과정은 각 룸바에 연결된 라즈베리파이가 개별적으로 연산을 진행하기 때문에 다수의 룸바를 개별적으로 제어할 수 있게됩니다.

****

**변경된 구조 사항에 대해서 간략하게 표현해보았습니다. 서버와 RunChecker는 독립된 머신에서 역할을 수행합니다. PDDL 드라이버와 H/W 드라이버는 모두 룸바에 연결된 라즈베리파이에서 연산을 진행합니다.**

**Issue 및 대책**

**<다중 룸바 동시 제어 테스트 >**

ROS의 멀티쓰레딩 환경에 익숙하지 않아 드라이버의 구조를 전체적으로 바꾸는 과정을 여러 번 수행했습니다. 이 과정에서 시간이 많이 소요되어 아직 마지막으로 재구성한 드라이버를 테스트해보지 못했습니다.

서버로부터 명령을 받고 이를 순차적으로 제어하는 부분은 이미 구현이 완료되었고, 다중 룸바의 제어를 편하고 정확하게 수행할 수 있는 테스트를 진행해야 합니다. 이 과정 이후에는 각각의 룸바가 명령을 수행할 때 필요한 Parameter값을을 최적화하여 최고의 결과를 이끌어내야합니다.