

## K-Digital Training 디지털핵심실무인재양성사업

<산업용 자율주행로봇 제어 테크니션 양성과정 프로젝트 활동보고서>

<b>프로젝트명</b>	조별 실내 자율주행 구현		
<b>일 자</b>	4/18~4/22 (4 주차)	<b>조 원</b>	안승수, 전우성, 김유경, 이동현
<b>1. 과제 개요</b>	<p>Scout Mini 에 부착된 센서 데이터를 활용하여 Mapping 과 Path planning 을 통한 실내 자율주행 구현</p>		
<b>2. 조원별 활동 현황</b>	<p>(1) 안승수 : SW</p> <ul style="list-style-type: none"> <li>- 로봇 구동을 위한 알고리즘 수정 및 개발</li> </ul> <p>(2) 전우성 : HW1</p> <ul style="list-style-type: none"> <li>- 실제 로봇 구동을 위한 업무 수행 및 로봇 관리</li> </ul> <p>(3) 김유경 : HW2</p> <ul style="list-style-type: none"> <li>- 전략 실행을 위한 자료조사 및 진행상황 기록</li> </ul> <p>(4) 이동현 : 기획</p> <ul style="list-style-type: none"> <li>- 문제 해결과 성능 개선을 위한 전략 수립</li> </ul>		
<b>3. 과제 수행 내용</b>	<p>[1] 미션 1 관련 추가 구현</p> <ol style="list-style-type: none"> <li>1) 속도 개선 및 장애물 회피 능력 향상을 위한 테스트</li> </ol> <p>[2] 미션 2 관련 구현</p> <ol style="list-style-type: none"> <li>1) 주차 능력 향상을 위한 parameter 테스트</li> <li>2) 주차 구역에서 정차를 위한 코드 작성</li> </ol>		
<b>4. 문제점 및 개선 방안</b>			

## K-Digital Training 디지털핵심실무인재양성사업

### [1] 미션 1 관련 추가 구현

#### 1) 속도 개선 및 장애물 회피 능력 향상을 위한 테스트

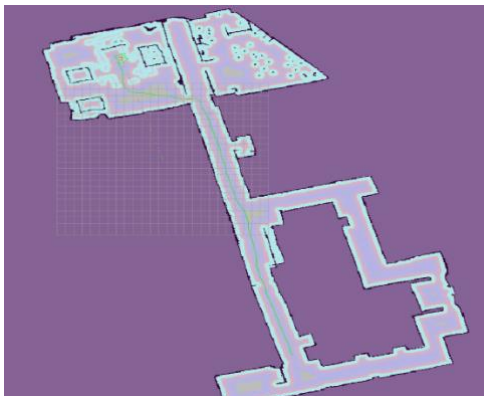
- 문제 인식 : Global Planner 가 최단 거리로 경로를 생성함에 따라 벽에 붙어서 가는 현상이 발생함.
- 개선 방안 : Global Costmap 의 inflation\_radius 를 0.4 -> 1.0 으로 변경함



[그림 1,2] Global costmap 의 inflation\_radius 0.4, 1.0 일 때 Global Path 의 변화

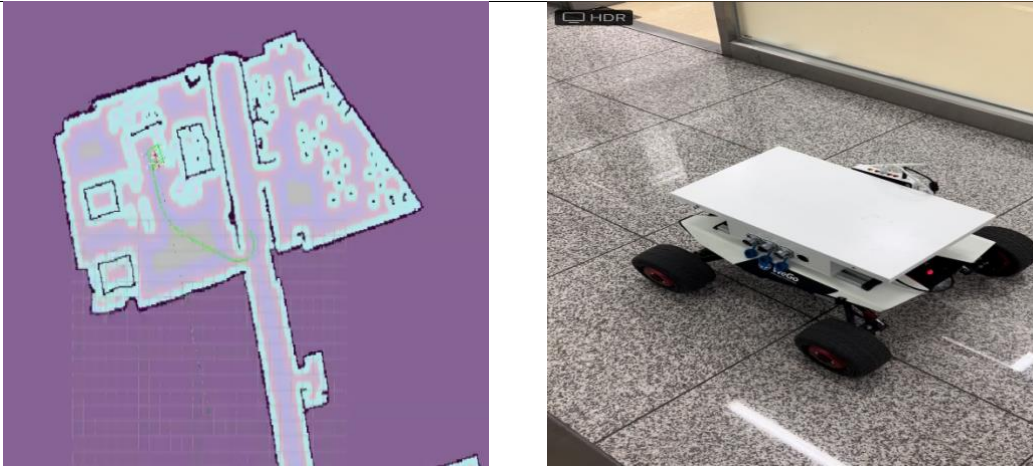
#### 2) 경로 최적화를 위한 테스트

- 문제 인식 : 제 4 강의장에서 나와서 우측으로 꺾어 진행하는 경로의 경우 로봇이 문에 부딪히는 현상이 발생함.
- 개선 방안 : 제 4 강의장에서 나와서 좌측지점에 중간 지점을 하나 추가한 뒤 이후 경로를 진행하도록 변경함.



[그림 3,4] 중간 지점을 추가하기 전 경로 및 충돌 발생 결과

## K-Digital Training 디지털핵심실무인재양성사업



[그림 5,6] 중간 지점을 추가 한 뒤 경로 및 정차 결과

### [2] 미션 2 관련 구현

#### 1) 주차 능력 향상을 위한 parameter 테스트

- 문제 인식 : 목적지 도착 여부를 move\_base 의 tolerance 로만 판단하면, 정확도가 떨어지는 경향을 보임.
- 개선 방안 : "Goal reached" 및 현재 위치와 목적지간의 거리 계산을 하여 도착 여부를 복합적으로 판단함.

```

if self.is_reach or (self.calc_distance() < self.dist_TH):
    self.service.request()
    rospy.loginfo('Finished {}'.format(self.idx+1))
    return True
# self.change_light_mode()
rospy.loginfo('Going to {}'.format(self.idx+1))
return False

# 목표지점과 현재위치 간의 거리를 계산하는 메서드
# xy goal tolerance 보다 작으면 계산 결과는 무의미 할 수도 있다.
def calc_distance(self):
    return ((self.cur_goal.position.x - self.cur_pose.position.x)**2 + (self.cur_goal.position.y - self.cur_pose.position.y)**2)**0.5
  
```

[그림 7] 도착 여부를 복합적으로 판단하는 코드 일부

## K-Digital Training 디지털핵심실무인재양성사업



[그림 8,9] 주차 능력 향상 전 후 결과 비교

### 2) 주차 구역에서 정차를 위한 코드 개선

- 문제 인식 : Sleep 명령을 통해 로봇의 움직임을 일정 시간 제어할 수 있으나, 시각적으로 판단하기에 부족함이 있음.
- 개선 방안 : Scout\_Mini 의 전방 Light 를 제어하여, 주행 중에는 라이트를 끄고, 정차시에는 라이트를 켜도록 구현함.

```
# 운행 중 혹은 도착 상태에 따라 전방 라이트 모드 변경 메서드 Only Activated in AutoMode(UP)
def change_light_mode(self,mode):
    new_mode = ScoutLightCmd()
    new_mode.enable_cmd_light_control = True
    new_mode.front_mode = mode
    self.light_pub.publish(new_mode)
```

[그림 10] Scout Mini 의 전방 라이트를 제어하는 코드 일부

### 5. 다음 일정 계획

#### [1] 미션 3 관련 구현

- (1) 실시간 Mapping 및 Navigation 을 위한 오픈소스 활용
- (2) 라이다 데이터에 의존하여 직접적으로 cmd\_vel 을 조작하는 방법 활용