

## DU-도전학기 결과보고서

성 명		학 번	
단과대학		학과(전공)	
도전학기 과제명	(국문) LIDAR 센서를 사용한 서빙로봇 (1) (영문) Serving Robot using LIDAR Sensor (1)		
지도교수 의견	<p style="font-size: 1.2em;">상기 학생은 ARS 등아리 학회장을 거치며 실무적인 능력이 대단히 높을 뿐만 아니라 영성도 넘치므로 본 프로젝트를 무반히 수행할 것으로 생각되어 추천함이다.</p> <p style="text-align: right;">(성명) (소속)</p>		

### 1. 도전 과제의 목표

LiDAR 센서를 사용하여 주변 공간을 스캔하여 맵 데이터를 저장하고, 자율적으로 주행하는 서빙로봇을 제작하는 것이 이번 도전 과제의 목표이다.

### 2. 도전 과제 내용

[붙임 1] 참고

### 3. 도전 과제의 성과

LiDAR 센서를 사용하여 SLAM을 실행시켰고, 이를 통해 지정된 좌표로 이동하는 서빙로봇을 구현하였다. SLAM을 통해 특정 공간에 대하여 지도를 생성하였고, 작동되는 동안 실시간으로 주변을 스캔하여 자율적으로 경로를 설정하여 목적지까지 이동한다.

프로젝트를 완성시키고 나서 이번학기 중에 진행될 여러 경진대회에 참가할 계획이었다. 하지만 코로나바이러스로 인하여 이번학기에 진행 예정이었던 경진대회들이 일정을 미루어 빠르면 여름방학과 혹은 2학기시기에 진행될 예정으로 미뤄진 상태이다. 따라서 현재까지 진행된 상태에서 기능들을 추가하여 추후 여러 경진대회에 참가할 예정이다.

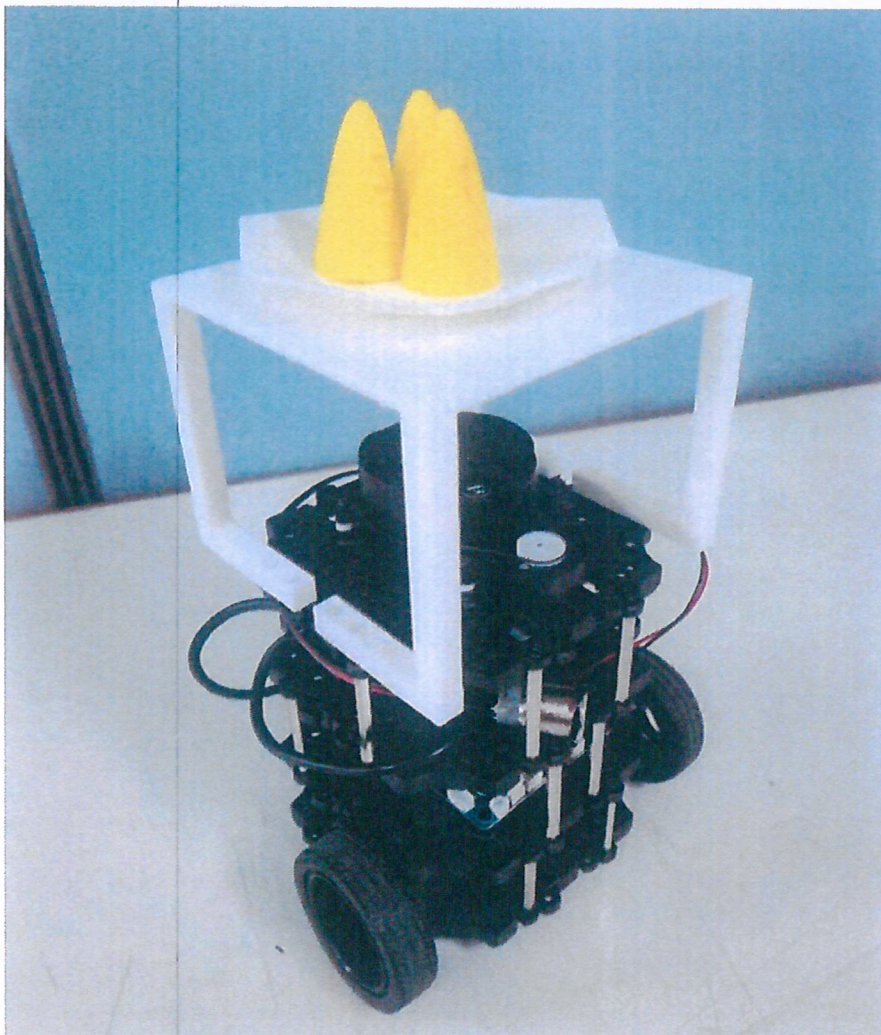
### 4. 자기평가

현재 꾸준히 성장하고 있는 자율주행 기술과 서빙로봇과 같은 제품들이 계속 출시되고 있다. 실무에서 많이 쓰이는 ROS를 다루며 실제 서빙로봇을 직접 구현을 하고나니, 이때까지 진행하였던 프로젝트보다 훨씬 더 어렵고 복

잡한 프로젝트라는 것을 실감하였다. 그만큼 졸업이후 취업을 하여 실무에 투입되면 분명 큰 도움이 될 거라 생각한다.

기존 계획에서 추가하였던 'Intel Depth Camera'를 사용하여 3D 형태로 SLAM을 사용하고, 서빙로봇을 구동할 수 있도록 계획을 추가하였었다. 하지만 필요한 패키지를 설치하고 실행하였음에도 불구하고 해결하지 못한 오류가 계속 발생하였다. 오류를 해결하지 못하여 결국 'Intel Depth Camera'를 사용한 3D SLAM을 실행시키지 못하였다. 도전학기 진행기간은 끝이 났지만, 방학기간을 활용하여 구현하지 못한 기능들을 보완 및 다른 기능들을 추가할 것이다. 또한 3D Printing을 통해 필요한 기물을 더욱 보완하고 외형 프레임을 제작할 것이다.

## 5. 최종 결과물



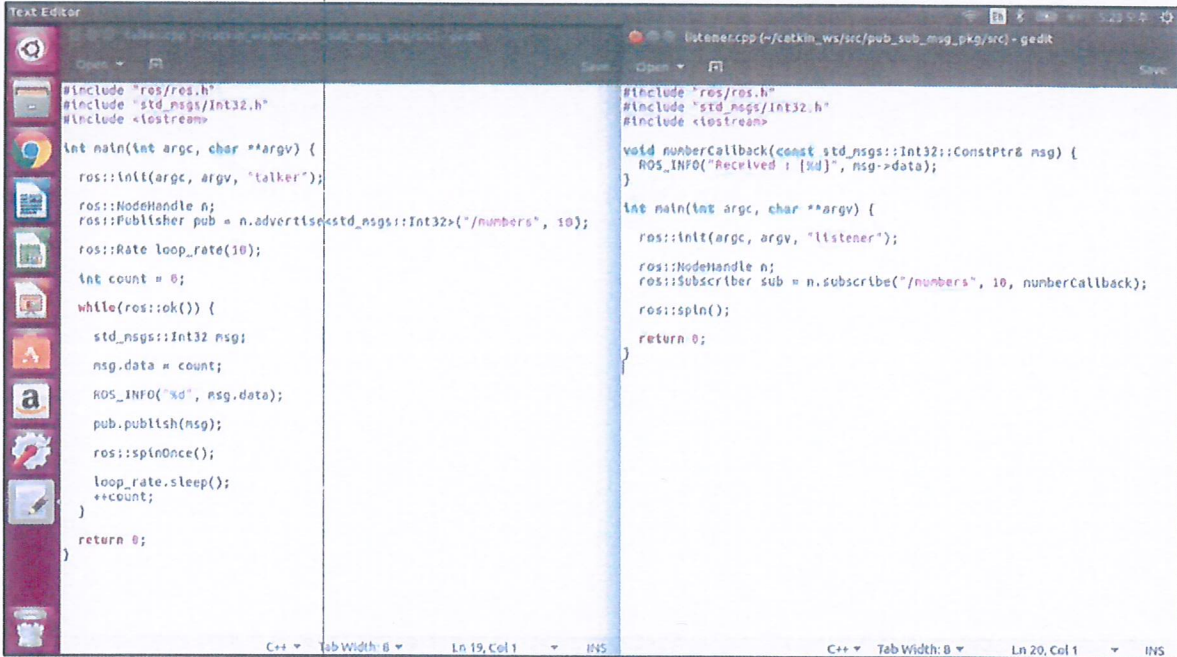
[붙임 1]

1. ROS 시작하기 - Publisher & Subscriber 이해

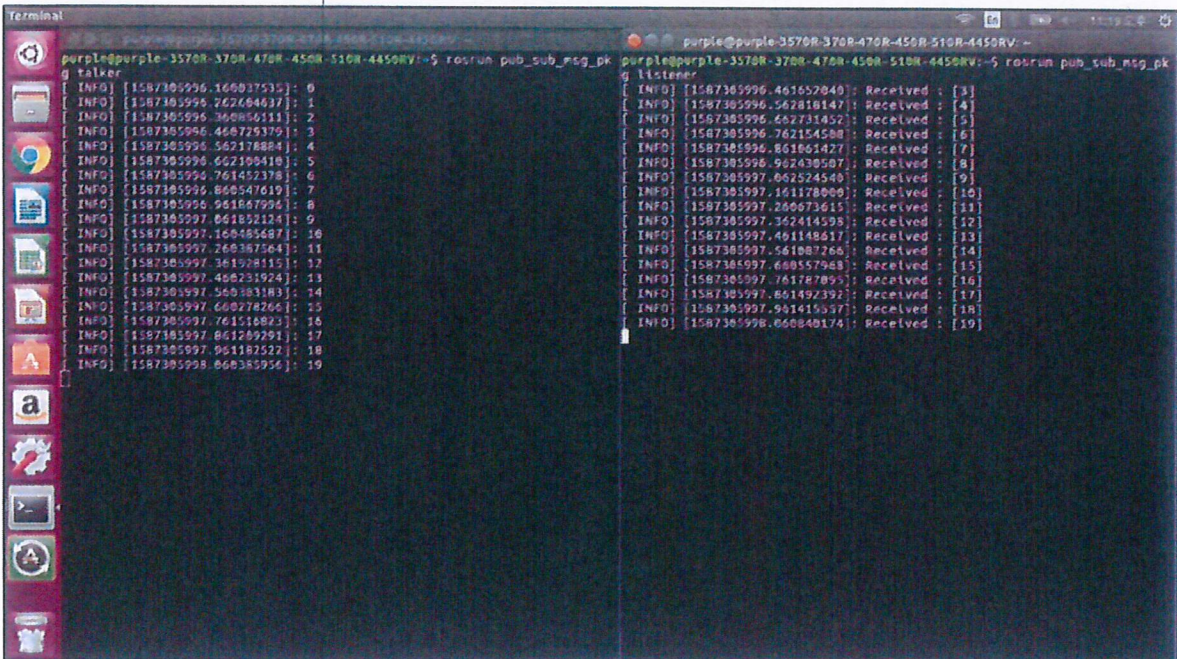
ROS를 처음 다뤄보기 때문에 기본구조중 하나인 Publisher와 Subscriber를 이해하는 것부터 시작하였다. 새로운 패키지를 생성하고 코드파일을 추가한다.

```
$ cd ~catkin_ws/src
$ catkin_create_pkg pub_sub_msg_pkg roscpp std_msgs actionlib actionlib_msgs
```

위와 같은 명령어를 통해 새로운 패키지를 생성하고, 패키지 안에 'src' 폴더에 Publisher 기능이 들어간 코드 파일과 Subscriber 기능이 들어간 코드파일을 추가한다.



이후 'catkin\_make' 명령어를 통해 위의 패키지를 빌드하고 터미널을 열어 각각의 파일들을 실행한다.

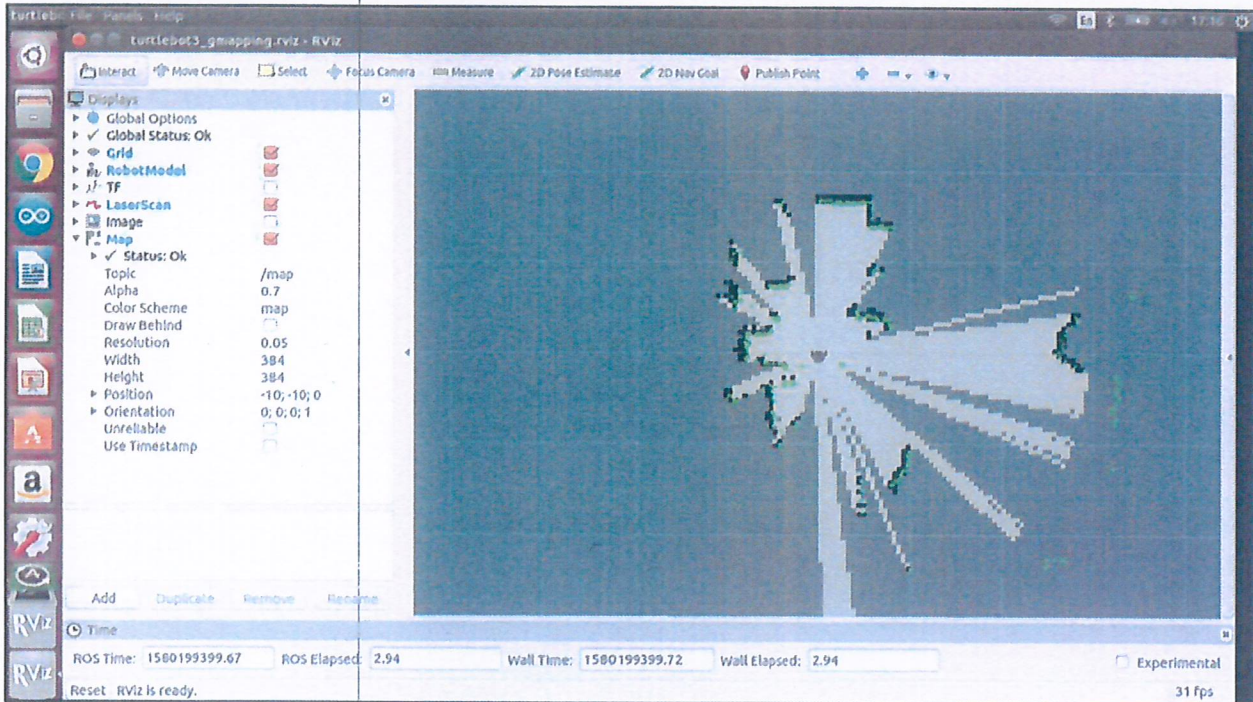


## 2. Turtlebot3 - SLAM & Navigation

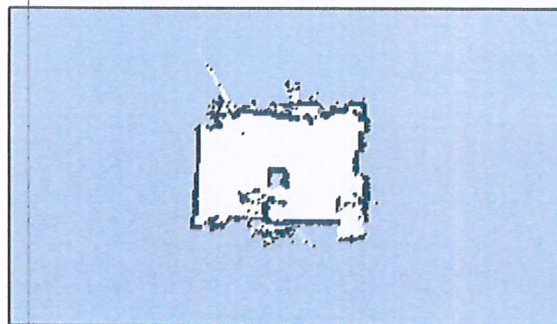
터틀봇3을 작동시켜 SLAM을 실행한다. 아래와 같이 명령어를 입력하면 SLAM이 실행되고 지도를 작성할 수 있다.

```
$ roscore
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
$ roslaunch turtlebot3_slam turtlebot3_slam.launch slam_method:=gmapping
```

실행하게 되면 아래와 같이 Rviz가 실행된다.



터틀봇3을 이동시켜 공간을 다 스캔하고, 데이터를 지도로 저장한다.



< SLAM을 통하여 생성된 지도 >

위와 같이 새롭게 생성된 지도를 가지고 Navigation 기능을 실행한다.

```
$ roscore
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/map.yaml
```

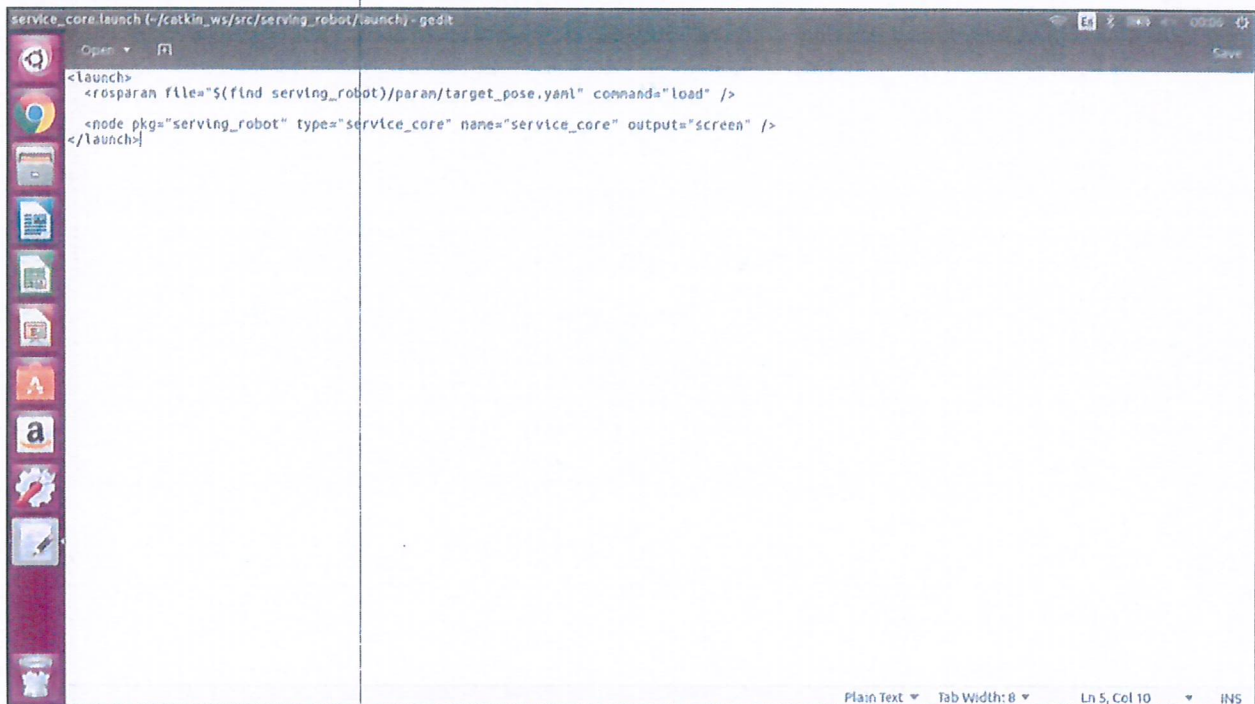
Navigation기능을 실행하면 SLAM과는 다르지만 마찬가지로 Rviz가 실행된다.



위와 같이 'position'의 x,y,z 값과 'orientation'의 x,y,z,w 값이 출력이 된다. 이제 Rviz에서 원하는 위치로 터틀봇3을 이동시켜주고 이동이 완료되었을 때, 터미널 창에서 'position'값과 'orientation'값을 확인한다. 이후, 추출한 좌표 값을 파라미터 파일에 저장한다. 실제 서빙로봇을 작동시키기 위해 터틀봇3을 지도상 특정위치 두, 세 군데로 이동하여 해당위치의 좌표 값을 위와 같이 확인하고 파라미터 파일에 저장한다.

### 3. ROS 노드 파일 + 파라미터 파일 - launch 파일 생성

서빙로봇을 동작시키기 위해서 launch 파일을 생성해야한다. 'roslaunch' 명령어로 실행하면 좌표 값이 저장된 파라미터 파일을 반환을 하지 못하여 launch 파일로 묶어주고 'roslaunch' 명령어로 서빙로봇을 동작시킨다. launch 파일 생성 후, 파일 안에 위치 좌표 값이 저장된 파라미터 파일과 좌표 값을 반환하여 서빙로봇을 작동시키는 소스코드를 추가하여 묶어준다.



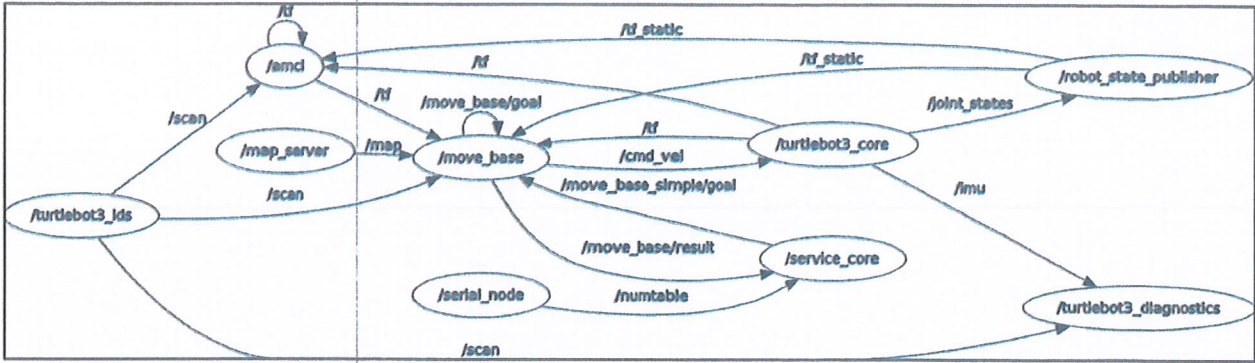
```

service_core.launch (-/catkin_ws/src/serving_robot/launch) - gedit
<launch>
<rosparan file="$(find serving_robot)/param/target_pose.yaml" command="load" />
<node pkg="serving_robot" type="service_core" name="service_core" output="screen" />
</launch>
  
```

### 4. 서빙로봇 구동

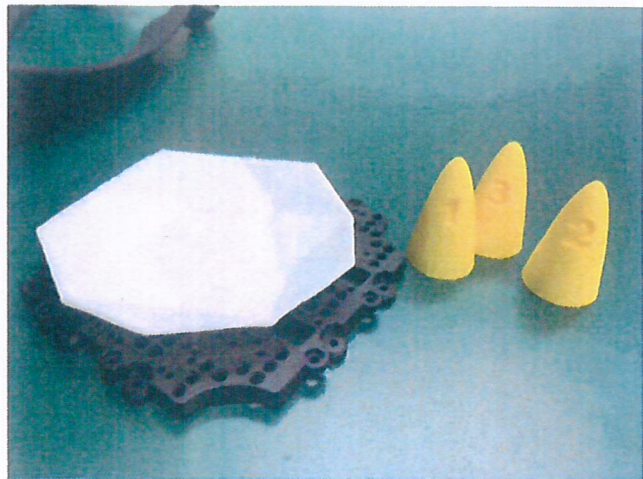
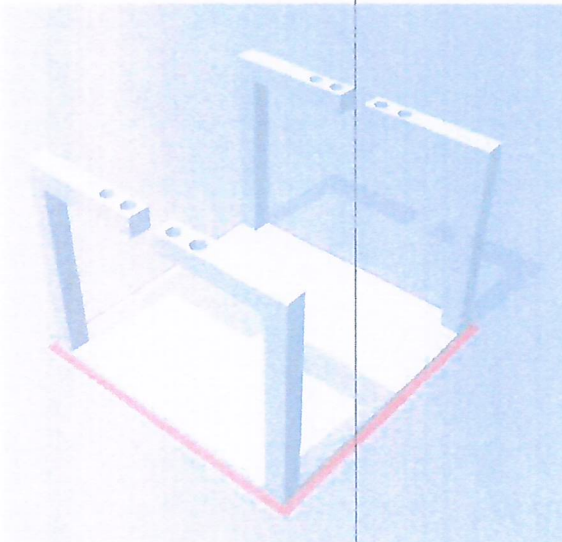
서빙로봇을 구동시키기 위해 Remote PC와 Turtlebot PC를 준비한다. Remote PC에서 roscore를 실행하고 Turtlebot PC에서 터틀봇3 구동 launch 파일을 실행한다. 이후, Remote PC에서 roserial, Navigation, service launch 파일들을 실행한다. 등록된 NFC 카드를 태그 하여 파라미터 파일에 저장된 좌표 값을 NFC 데이터에 따라 반환한다. 이후 단계별로 서빙로봇을 동작시켜 목표지점까지 이동 후, 제자리로 돌아온다.

위와 같은 과정들을 쉽게 정리한 것이 아래의 그림이다. 이 그림은 노드파일 혹은 launch파일을 실행하였을 때, 각각의 노드들이 어떠한 토픽을 가지고 어떤 노드들과 연결이 되어 통신을 하는지 보여주는 그림이다.



## 5. 3D Printing

서빙로봇의 필요한 프레임 제작하기 위해 3D Printing을 진행하기로 하였다. 음식을 얻을 받침대와 접시를 제작하였고 NFC 태그 키와 결합하여 테이블 번호를 알려줄 기물을 제작하였다. Autodesk사의 Inventor 프로그램을 사용하여 도면을 제작하였고 3D Printing을 진행하였다.



## 6. Intel Depth Camera

이번 DU 도전학기를 진행하면서 'Intel Depth Camera'를 사용하는 것을 추가하였다. Intel Depth Camera를 사용하게 되면 3D 형태의 SLAM을 사용할 수 있게 되는데, 기존 2D LiDAR 센서로 SLAM을 사용하는 방법보다 훨씬 안정적으로 서빙로봇을 구동시킬 수 있다. 그만큼 카메라를 통한 3D SLAM을 적용시키는 것이 LiDAR 센서보다 많이 까다롭다.

LiDAR 센서를 사용하여 서빙로봇의 기능을 구현한 후, Intel Depth Camera를 사용하기 위해 필요한 패키지를 설치하고 나서 매뉴얼대로 실행하였다. 하지만 아래와 같이 에러가 발생하였다.

```

/opt/ros/kinetic/share/realsense2_camera/launch/rs_camera.launch http://192.168.0.7:11311
* /camera/realsense2_camera/intra_width: 640
* /camera/realsense2_camera/initial_reset: False
* /camera/realsense2_camera/jsoe_file_path:
* /camera/realsense2_camera/linear_accel_cov: 0.01
* /camera/realsense2_camera/odom_frame_id: camera_odom_frame
* /camera/realsense2_camera/pointcloud_texture_index: 0
* /camera/realsense2_camera/pointcloud_texture_stream: RS2_STREAM_COLOR
* /camera/realsense2_camera/pose_frame_id: camera_pose_frame
* /camera/realsense2_camera/pose_optical_frame_id: camera_pose_optic...
* /camera/realsense2_camera/publish_odom_tf: True
* /camera/realsense2_camera/publish_tf: True
* /camera/realsense2_camera/rosbag_filename:
* /camera/realsense2_camera/serial_no:
* /camera/realsense2_camera/tf_publish_rate: 0.0
* /camera/realsense2_camera/topic_odom_in: odom_in
* /camera/realsense2_camera/unite_lm_method:
* /camera/realsense2_camera/usb_port_id:
* /roscpp_core: kinetic
* /rosversion: 1.12.14

NODES
  /camera/
    realsense2_camera (nodelet/nodelet)
    realsense2_camera_manager (nodelet/nodelet)

ROS_MASTER_URI=http://192.168.0.7:11311

process[camera/realsense2_camera_manager-1]: started with pid [3800]
process[camera/realsense2_camera-2]: started with pid [3801]
[ INFO ] [1591800222.269144910]: Initializing nodelet with 4 worker threads.
[ INFO ] [1591800222.325835830]: RealSense ROS v2.2.13
[ INFO ] [1591800222.325868779]: Running with LibRealSense v2.33.1
[ INFO ] [1591800222.348512510]:
10/06 23:43:42,360 ERROR [149318635734784] (handle_libusb.h:51) failed to open usb interface: 0, error: RS2_USB_STATUS_ACCESS
10/06 23:43:42,385 WARNING [149318667395500] (rs.cpp:283) null pointer passed for argument "device"
[ WARN ] [1591800222.38513231]: Device 1/1 failed with exception: failed to set power state
[ WARN ] [1591800222.38513231]: The requested device with ID 1 is not found, will try again.
[ INFO ] [1591800228.405995584]:
10/06 23:43:48,413 ERROR [149318644127488] (handle_libusb.h:51) failed to open usb interface: 0, error: RS2_USB_STATUS_ACCESS
10/06 23:43:48,434 WARNING [149318667395500] (rs.cpp:283) null pointer passed for argument "device"
[ WARN ] [1591800228.43487984]: Device 1/2 failed with exception: failed to set power state
[ WARN ] [1591800228.43487984]: The requested device with ID 1 is not found, will try again.

```

이 에러를 해결하기 위해 수많은 방법들을 찾아보았고, ROS를 다루시는 졸업한 선배들에게도 연락을 하여 해결책을 찾으려고 시도하였으나, 결국 Intel Depth Camera를 사용하지 못하였다. 이번 2020년 1학기가 끝나고 여름 방학기간부터 선배님들과 더욱 많은 시간을 교류하며 이 에러를 해결할 수 있도록 진행할 예정이다.