PODEX 최종 보고서

(실내 배송 RC-Car)

B반 3조

김동영 김에녹 문한빈

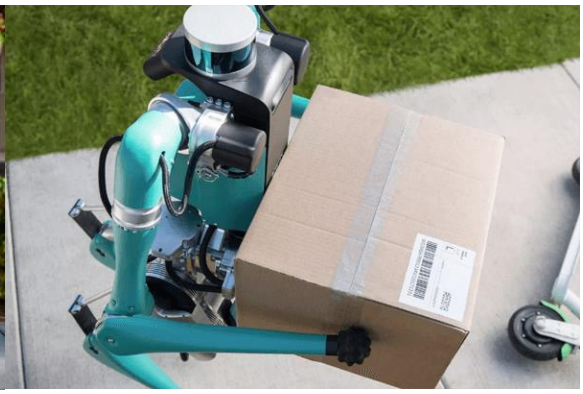
백관진 전지윤 최효진

목차

1. 주제 선정

1.1 배경 및 동기

2019년 1월, 아마존이 자율주행 택배 배달로봇 '스카우트(Scout)'를 공개했다. 소형 냉장고 크기로 제작된 스카우트는 사람이 걷는 속도로 인도를 따라 주행하며, 보행자 또는 장애물을 피해 목적지까지 안전하게 도착하도록 설계됐다. 스타십 테크놀로지는 스카우트보다 더 작은 자율주행 택배 로봇을 개발해 시범 서비스를 선보이기도 했다. 아마존을 비롯한 전 세계 IT 기업들이 다양한 인공지능 로봇과 드론을 자율주행차와 결합해 배송서비스 테스트를 선보이고 있는 만큼, 세계의 자율주행택배에 대한 관심과 현실성은 더욱더 커지고 있다. 세계 각국에서 이렇게 연구와 개발을 하는 이유는 이것의 미래에 엄청난 부가가치를 가질것으로 예상되기 때문이다. 우리나라는 아직 자율주행택배 서비스에 대해서는 걸음마 단계이다. 이에 자율주행택배 개발이 선택이 아닌 필수가 되었다.



1.2 연구 계획

우리는 택배 로봇으로써 RC car (Remote Control Car), 자율주행 연산을 위한 RPI(Raspberry Pi), 이미지 인식을 위한 RealSense Camera, 2차원 지도 제작을 위한 RP ridar를 사용할 예정이다. Donkey car에 탑재된 RPI를 사용하여 워크 스테이션에서 원격으로 신호를 주고 받는다. RPI에는라즈베리안 대신 Ubuntu를 설치했다. 그 이유는 차량을 주행하기 위해 필요한 소프트웨어 ROS(Robot Operation System)와 호환되는 것이기 때문이다. 그 이후, 차량에 카메라를 설치하여 들어오는 이미지로부터 목표지점인지 유무와 택배 수신자의 얼굴을 확인한다. 자율주행을 위해서는 사전에 주행 지형에 대한 지도가 필요하다. 이 지도를 작성하는 소프트웨어는 SLAM(Simultaneous localization and mapping)이다. RP ridar를 이용하여 지형으로부터 반사되는 빛을 측정하여 실시간으로 2차원 지도를 작성해주는 프로그램이다. 이렇게 사전 지도 작성, 목표지점 입력, 자율주행, 목표지점과 택배 수신자의 확인으로 자율주행택배 과정이 진행된다.

2. 무인택배로봇 현황

2.1 해외 개발 현황

<독일>

콘티넨탈 사가 개발한 배달용 로봇 개와 무인자동차 큐브(사진=콘티넨탈)

무인자동차에서 내린 로봇 개들이 택배를 배달해주는 기술이 공개됐다.

8일(현지시간) 미국 IT 매체 테크크런치는 이날 미국 라스베이거스에서 열린 세계 최대 IT·가전 전시회 'CES 2019'에서 공개된 콘티넨탈 사의 배달용 로봇 개에 대해 소개했다.

콘티넨탈사는 주문자의 집 문 앞까지 사람의 손을 거치지 않으면서도 끊김 없는 로봇 배송 체계를 구축하고자 로봇 개를 개발했다.

무인자동차 '큐브'에서 내린 수 마리의 로봇 개들이 등에 택배를 싣고 일정 반경 안에서 배달을 수행하게 된다.

<영국>

1. 영국 ByBox 사례

영국의 "ByBox Holdings Limited"는 Stuart Miller에 의해 창업되었다.

Stuart는 처음 Logibag SAS를 인수하여 공중전화 부스, ATM, 지하철 등에 물품을 보관하는 박스를 제조 및 설치하였으나, 이들 박스에 우편물, 택배 등을 배송하는 아이디어에 착안하여 ByBox를 설립하여 사업을 확장하였다. 또한, 우편물을 보다 효율적으로 배송하기 위하여 2003년 9월에는 전문 배송업체인 "Hay plc"를 인수하여 배송의 전문성과 안정성을 확보하였다.

〔그림 3〕 ByBox 서비스 프로세스

이러한 Stuart의 생각은 적중하여, ByBox 2002년 매출액이 약 1천만 유로(한화 약 1억 5천만원)에 불과하였으나, 창립 10년만에 2010년 매출액은 약 42백만 유로(한화 약 620억)으로 15배 고성장하였으며, 영국, 프랑스, 아일랜드, 베네룩스 등으로 사업영역을 확장하고 있다

<중국>

중국 징동상청(이하 징동)에서 6월 18일자로 무인택배로봇 서비스를 시작했다.징동은 베이징 해정구(北京市海淀区)에서 서비스를 발표하였다.

그리고 이 서비스는 중국 공안으로부터 도로주행에 대한 감독을 받고 있다. 도로를 시속

15Km/h로 달릴 수 있다.

로봇에는 360도 카메라와 각종 센서가 장착되어 도로 노선을 인식 할 수 있다고 합니다. 그래서 당연히, 도로의 노선을 지키며 정차 할 수 있습니다.

이러한 택배지 배송은 택배의 연결과 사무실 및 집 그리고 공원 등 다양한 공간에서 물건을 전달 할 수 있는 환경을 제시 할 것이다.

또한, 물건 전달 방법은 사용자 얼굴인식 및 피킹코드 입력 방법, APP 링크 확인의 3가지 방법이 존재한다.

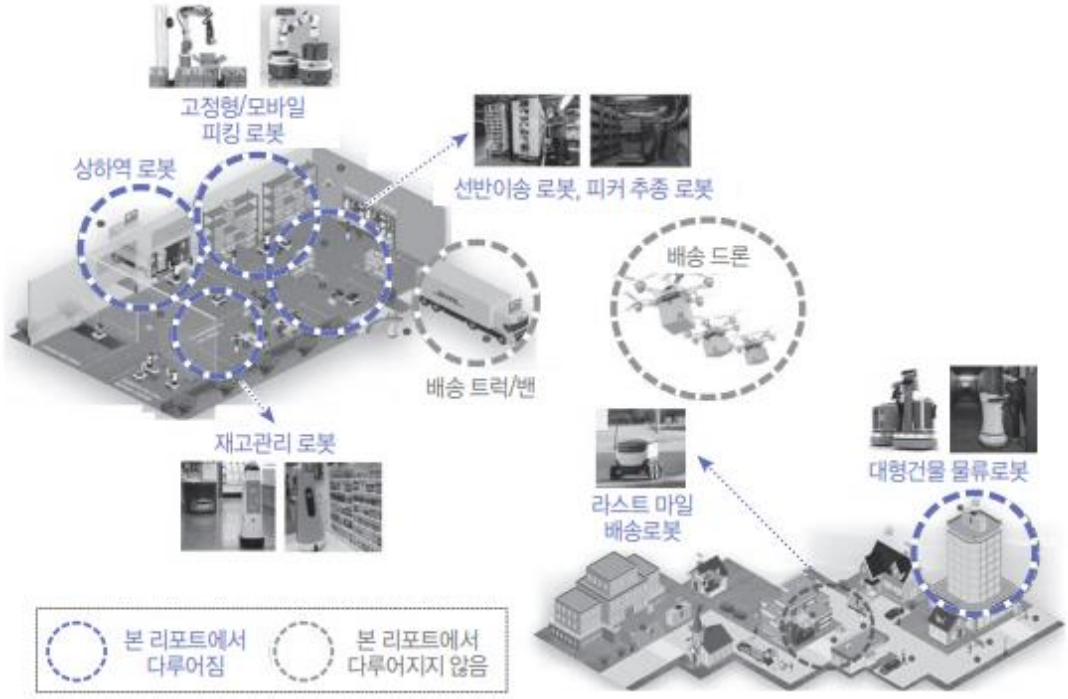
로봇의 옆면의 작은 칸을 통하여 물건을 수취하는 장면

2.2 <국내 현황>

해외에서 무인 택배 로봇이 개발되는 것과 달리 국내에서는 그 움직임이 더디다.

무인 사물함이 공고히 자리잡혀 있는 데다가 연구하는 정도로 그치고 있다.

물류센터, 공장 등에서 IoT 기술과 자율주행 등 로봇 기술 및 학습을 통한 환경·상황인식, 스케줄링 등 인공지능 기술의 융합을 통한 물류 효율 향상을 목적으로 하는 로봇시스템으로 물품의 포장·분류·적재 및 이송과정에 주로 활용된다.



※ 그림 배경 출처: "Robotics in Logistics: A DPDHL perspective on implications and use cases for the logistics industry", 2016, DHL Trend Research

| 그림 1. 물류로봇의 유형 및 본 리포트 포함 범위 |

택배 분류 로봇까진 개발이 되었으나 배송까지 무인화하는 로봇은 개발중이다.

국토교통과학기술진흥원의 자료에 따르면 두 가지 연구과제를 추진하고 있다.

연구과제1 - 무인 로봇을 활용한 택배산업 응용기술 연구
ㅇ한국형 무인 로봇 활용 택배 모델 및 응용기술 개발 전략 수립

연구과제2 - 중소 택배기업 공용물류 시스템 비즈니스 모델 개발
ㅇ공동물류 시스템 구성을 위해 집화/배송의 운영 네트워크, 新 허브/서브 터미널 설계(분류를 위한 Layout 구성, 물류장비(Unit Load) 구성 등) 진행 및 중소 택배기업를 대상으로 실행 시뮬레이션을 수행
ㅇ공용물류 시스템 개발 사업화를 위한 정부지원 방안 및 이에 따른 법·제도 개선방안 도출

현재 시험운영 중인 회사로는 택배 배달이 아닌 음식배달 로봇이 개발중이다. 추후 배달 전문 로봇으로 발전할 가능성이 엿보인다.

지난해 3월 인공지능(AI) 투자 프로젝트 '배민데이빗'을 공개, 100억원 투자 계획을 밝혔던 우아한형제들은 같은해 하반기부터 로봇 개발에 발을 담갔다.

음식배달 수요는 폭발적으로 증가하는 반면, 수요를 공급할 수 있는 배달기사 수는 한정된 만큼 이 문제를 배달 전문 로봇으로 풀어낸다는 구상이다.

우아한형제들의 배달로봇 프로젝트는 총 3단계로 나뉜다.

1단계로 이번에 공개된 딜리처럼 실내 공간에서 안정적으로 배달을 해주는 로봇을 개발한 뒤,

2단계로 아파트 단지와 같은 실내·외가 복합된 공간을 위한 로봇을 연구, 개발할 계획이다.

마지막 3단계는 인도를 이용해 음식점부터 고객이 있는 위치까지 음식을 배달해주는 로봇을 만들어 날씨가 궂은 날이나 배달이 밀리는 시간대에 로봇을 활용하는 방안이다. 배달기사들이 기피하는 지역이나 날씨에 로봇이 투입돼 빠른 배송을 실현한다는 것이 최종 목표다.

이처럼 아직 국내에서의 개발 및 연구 상황은 이제 막 시작하는 단계이다. 무인택배의 목표보다는 무인사물함과 무인 음식배달이 수익을 가져다 주기 때문이다.

참고문헌)

기사:

https://itreport.tistory.com/555 [IT REPORT WORLD]

논문:

[국토교통과학기술진흥원]http://www.korearobot.or.kr/wp/wp-content/uploads/2017/08/KEIT-PD17-7-%EC%9D%B4%EC%8A%882-%EB%AC%BC%EB%A5%98%EB%A1%9C%EB%B4%87-%EA%B8%B0%EC%88%A0%EB%8F%99%ED%96%A5-%EB%B0%8F-%ED%96%A5%ED%9B%84%EC%A0%84%EB%A7%9D.pdf

[물류로봇] https://www.kisdi.re.kr/kisdi/common/premium?file=1%7C12711

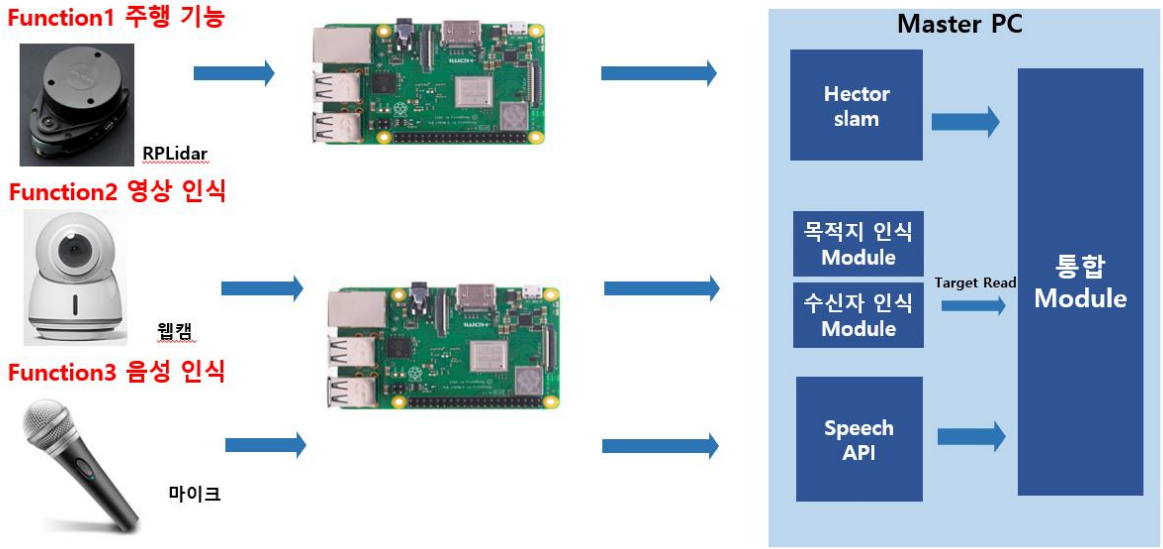
[무인택배함] http://xn--jj0bv9dsyh2ot6thnkb01s.com/down/mu.pdf

[배달로봇 달리] https://www.zdnet.co.kr/view/?no=20180315111625

3. 하드웨어 및 소프트웨어

3.1 프로젝트 구조도

- 1번 라즈베리 파이는 Donkey Car와 Lidar를 연결해 차량 제어 및 SLAM 기능 구현
- 2번 라즈베리 파이는 카메라와 마이크와 연결해 영상 인식과 음성 인식 기능 구현
- 1개의 Master PC에서 1번과 2번 라즈베리 파이를 제어



3.2 하드웨어

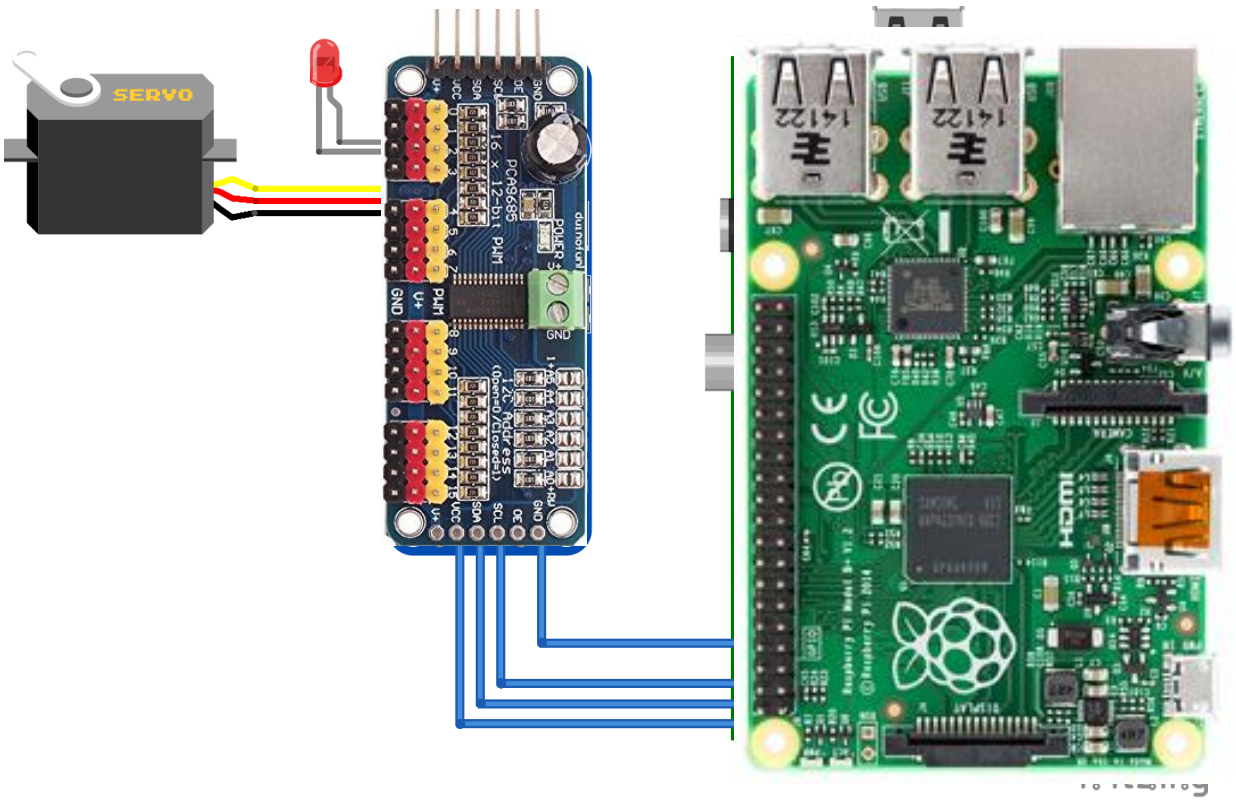
1) Donkey car

해외에서 자율주행 RC카를 구현할 때, 가장 많이 사용하는 모델이다. Donkey car는 다양한 소프트웨어 Package를 제공하고 있으며, 해외에서 많이 쓰는 모델이기 때문에 정보가 많다. 크기는 1:16이고 배터리 용량은 1100mAh로 다소 낮다. 배터리는 RC카, 라즈베리파이, 카메라, LiDAR, 전력공급를 하기 때문에 배터리 용량이 매우 부족하였다.

2) Servo motor (PCA9685)

서보 모터를 조절하기 위해 PCA 보드를 사용하였다. 모터는 동키카 안에 있는 것이고 PCA9685는 모터와 램프등을 구동하는 드라이버이다. I2c 통신을 사용하며 라즈베리파이를 통해 동키카를 제어하는 기능을 수행한다. 방향과 속도를 변경할 수 있다. PWM(Pulse Width Moduation) 방식을 하며 가변정항갑스이 변화 입력에 따라 듀티비가 조절되어 출력을 조절한다. 판넬 및 소형케이스에 장착하기 용이하도록 크기를 최소화 했으며 운전 상태를 LED를 통하여 확인 가능하다 . 부하는 12V~24V, 5A 까지 가능하나 3A 이상 사용시 부하가 걸려 다이오드1N5408를 점퍼 와이퍼로 쇼트 시켜서 사용해야 한다. 또한 외부에서 원격으로 동작을 제어할 수 있는 제어단자가 마련되어 있다.

3) Raspberry Pi(RPI)



영국의 RPI 재단이 학교에서의 기초 컴퓨터 과학 교육용 프로젝트의 목적으로 개발한 신용

카드 크기의 초소형/초저가 PC로 2006년에 개념이 형성되고 재단이 만들어져 2012년 초기의 라즈베리가 세상에 나오게되었다. 라즈베리파이의 특징은 아두이노와 달리 키보드, 마우스, 모니터만 연결하면 PC가 될 수 있다. LinuxOS (Raspbian Strecth)를 기반으로 하고 아두이노의 보다 더욱 다중 처리에 적합해서 RPI을 사용하였다.

참고문헌)

Dongkey 카 소프트웨어 구동 영상 & ROS

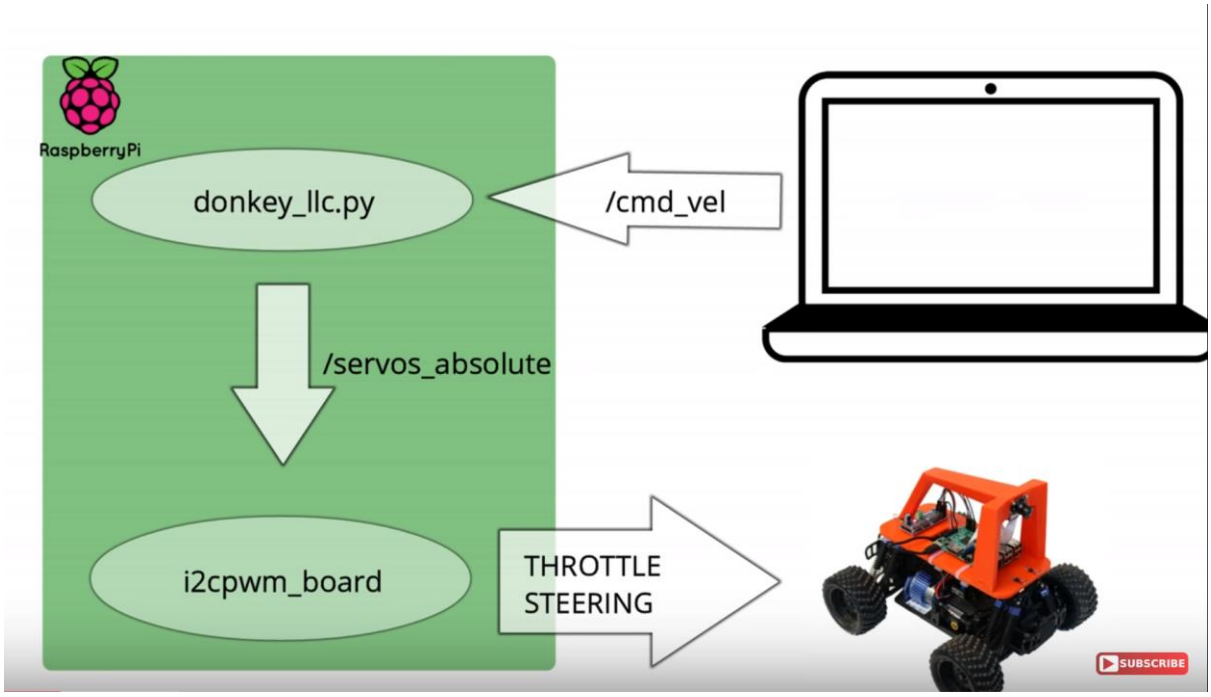
https://www.youtube.com/watch?time_continue=4&v=iLiI_IRedhI

Dongkey 카 배선 연결 : 43분부터 배선 연결은 56분

https://www.youtube.com/watch?v=byRLYkZkJZE

3.3 Ubuntu에 ROS 설치하기

노트북 또는 워크 스테이션에 ROS(Robot Operating System)을 설치하여 donkey car를 움직일 계획입니다. 이를 위해서는, donkey car를 구동하는 RPI(Raspberry Pi)에 ubiquity robotics(ROS + Ubuntu)가 설치되어 있어야 합니다. 설치 과정은 Youtube의 "ROS and Raspberry Pi for Beginners | Tutorial #0 - Topics Packages RosMaster"를 참고하여 작성되었습니다.



1) 컴퓨터에 ROS 설치

본 내용은 wiki.ros.org/kinetic/installation을 참고하여 작성되었습니다. ROS Kinetic은 Ubuntu 16.04, 15.04만을 지원합니다.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

sudo apt-get update

sudo apt-get install ros-kinetic-desktop-full

apt-cache search ros-kinetic

sudo rosdep init

rosdep update

echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

source ~/.bashrc

source /opt/ros/kinetic/setup.bash

echo "source /opt/ros/kinetic/setup.zsh" >> ~/.zshrc

source ~/.zshrc

sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential

source /opt/ros/kinetic/setup.bash

mkdir -p ~/catkin_ws/src

cd catkin_ws/

catkin_make

source devel/setup.bash

sudo nano ~/.bashrc #파일의 끝에 소스 source ~/catkin_ws/devel/setup.bash 입력
```

2) 라즈베리파이에 ubiquityrobotics 설치

sd카드에 미리 구워진 Ubiquity robotics를 다운로드 하십시오. Ubiquity robotics에는 ROS와 Ubuntu가 설치되어 있습니다. SSH를 이용하여 donkey car에 접속할 예정입니다.

Ssh ubuntu@ubiquityrobot.local

초기 Password는 ubuntu입니다.

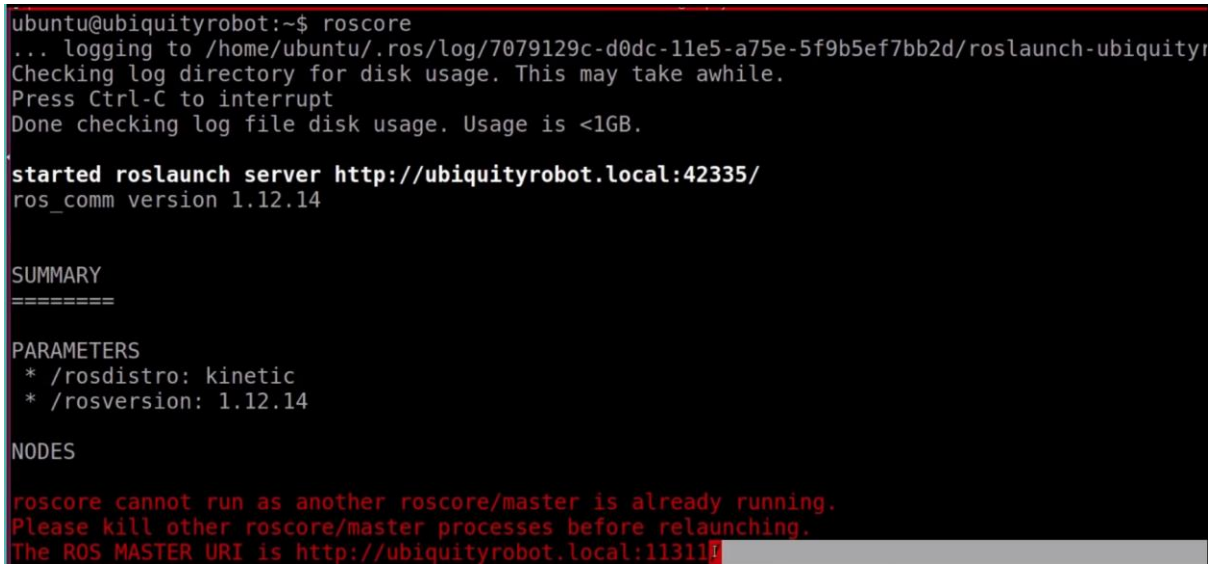
```
Sudo apt install samba

Sudo nano /etc/samba/smb.conf
```

스크롤을 아래로 내려 path = /home/ubuntu/catkin_ws/src/를 확인합니다.

```
Sudo smbpasswd -a ubuntu
Sudo reboot
Roscore
```

아래와 같은 화면이 실행되면 ros가 잘 설치된 것입니다.



3) 서보 제어 보드 노드 설치하기

Donkey car를 제어하기 위해서는 제어 프로그램 서보 제어 보드 노드를 설치해야 합니다.

```
roscd
cd..
cd src/

git clone https://gitlab.com/bradanlane/ros-i2cpwmboard
cd..

catkin_make



source devel/setup.bash

cd src/ros-i2cpwmboard

cd launch/
```

```
sudo gedit i2cpwm_node.launch

sudo nano i2cpwm_node.launch
#  sudo -E 라는 부분을 제거한다.

cd ../msg/
nano ServoArray.msg
nano Servo.msg
nano package.xml



cd ..
nano CMakeLists.txt
```

4) donkey car 제어하기

```
rosrun i2cpwm_board i2cpwm_board # 서보모터 패키지 명령어 인식 안



#new terminal

rostopic list

#/servos_absolute
#/servos_drive
#/servos_proportional 을 확인한다
rostopic info /servos_absolute




rosmsg show i2cpwm_board/ServoArray

rostopic pub /servos_absolute i2cpwm_board/ServoArray
# - servo: 0
# - value: 0.0" 를 확인한다.
# 이 두가지 값을 변경함 으로써 donkey car 를 조종할 수 있다.
```

```
# servo : 1
# value: 333.0" -> 정지상태

# servo : 1
# value: 363.0" -> 이동상태

# servo : 2
# value: 300.0" -> 왼쪽으로 방향 전환
```

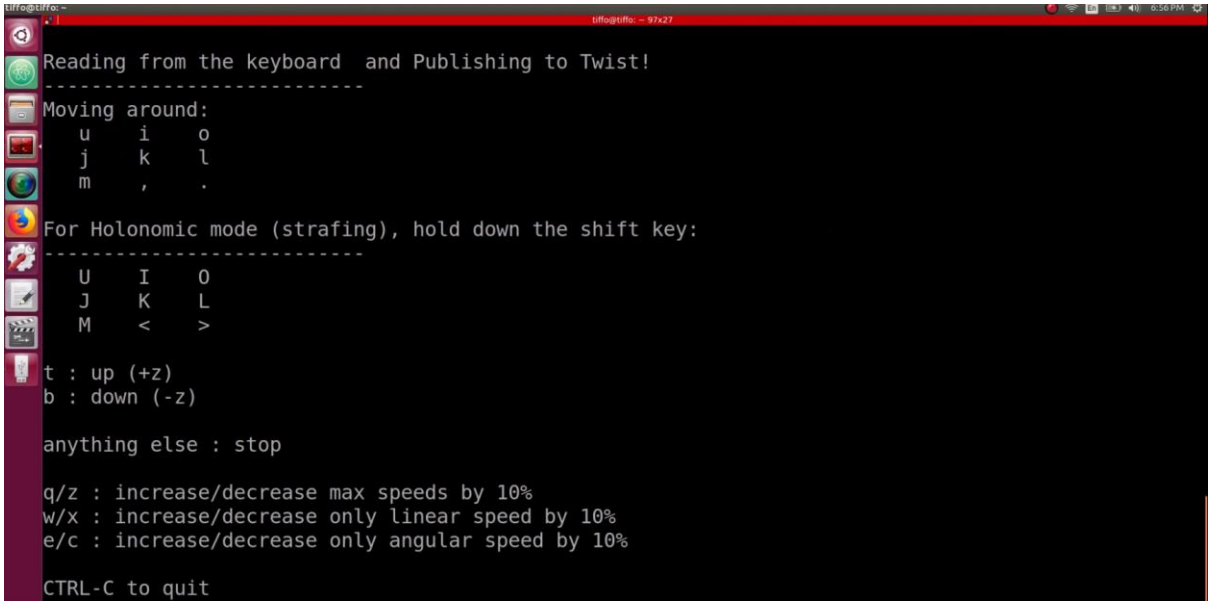
5) 패키지 만들기

```
~/catkin_ws/src$ catkin_create_pkg donkey_llc rospy
Nano CMakeLists.txt
~/catkin_ws/src/donkey_llc$ cd ..
Cd ..
Catkin_make
Rosrun i2cpwm_board i2cpwm_board
```

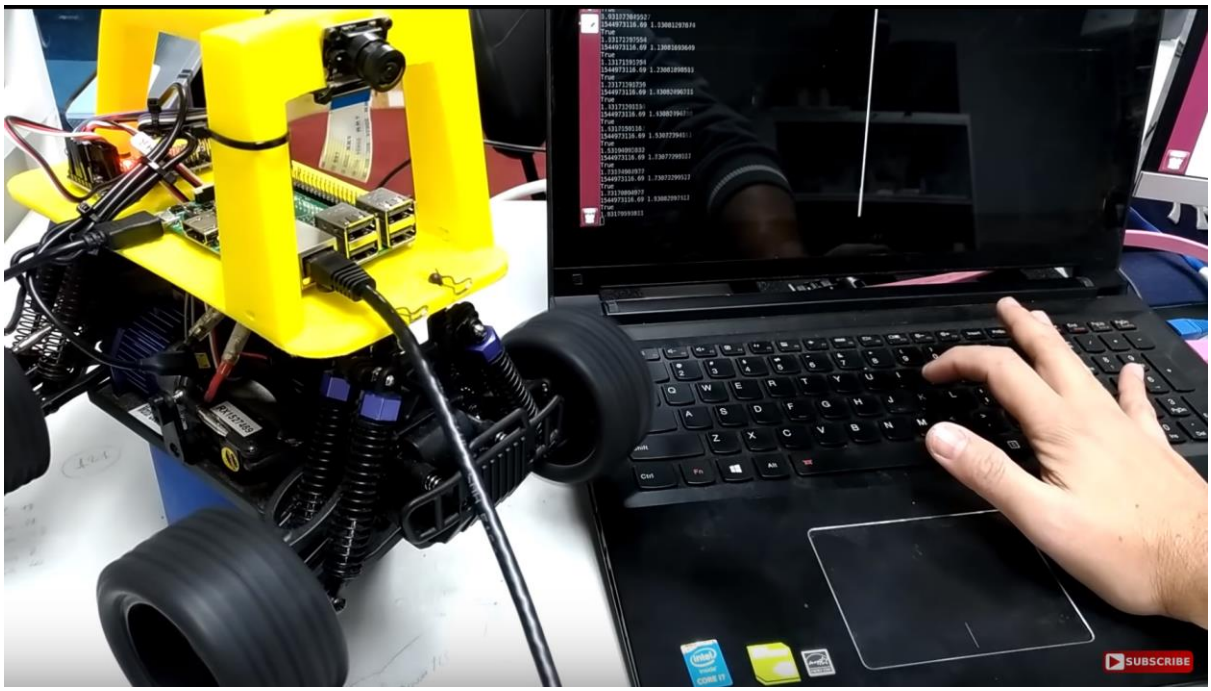
컴퓨터에서 아래 코드를 입력합니다.

```
Export ROS_MASTER_URI=http://ubiquityrobot.local:11311
Export ROS_IP=’hostname -I’
Hostname -I

Rostopic list
Rosrun teleop_twist_keyboard.py
```



키보드를 사용하여 donkey car를 작동할 수 있습니다.



참고문헌)

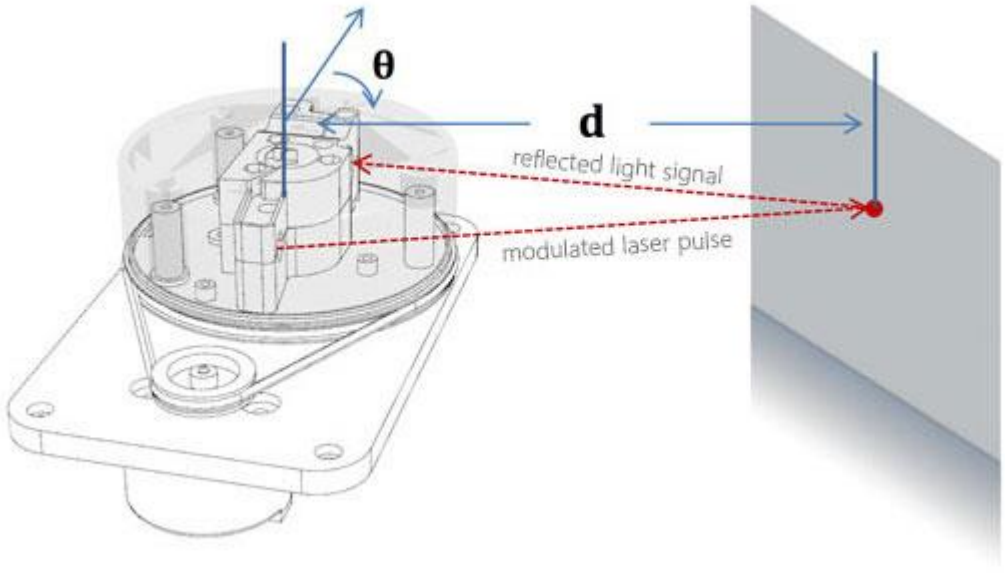
WIKI.ROS.ORG/CATKIN 참고

www.donkeycar.com 참고

https://youtu.be/iLiI_IRedhI (ROS and Raspberry Pi for Beginners | Tutorial #0 - Topics Packages RosMaster)

3.4 RPridar1

Robopeak社에서 출시한 LIDAR(Light Detection and Ranging) Sensor인 RPLIDAR는 SLAM을 구현하기 위한 저렴한 솔루션으로 관심받는 제품이다. 원리는 1차원 레이져센서를 회전시키면서 각에 대한 거리를 스캔함으로 2차원 데이터를 생성한다.



기능은 :2D 레이저 스캐너로 360도에 6미터 이내 범위를 레이저로 스캔할 수 있다. 스캔 자체로는 RVIZ 프로그램에서 붉은 점으로만 나타난다. 종합적인 맵핑을 할 수 있는 SLAM 이 필요하다.

RVIZ 프로그램은 아래 Ridar를 설치하다 보면 자연스럽게 설치가 진행된다.

우선 Ridar를 Workstation에 설치하여 작동하는지를 살펴보았다. 관련 코드는 아래와 같다.

1) RP ridar 소스관리 툴 git 설치

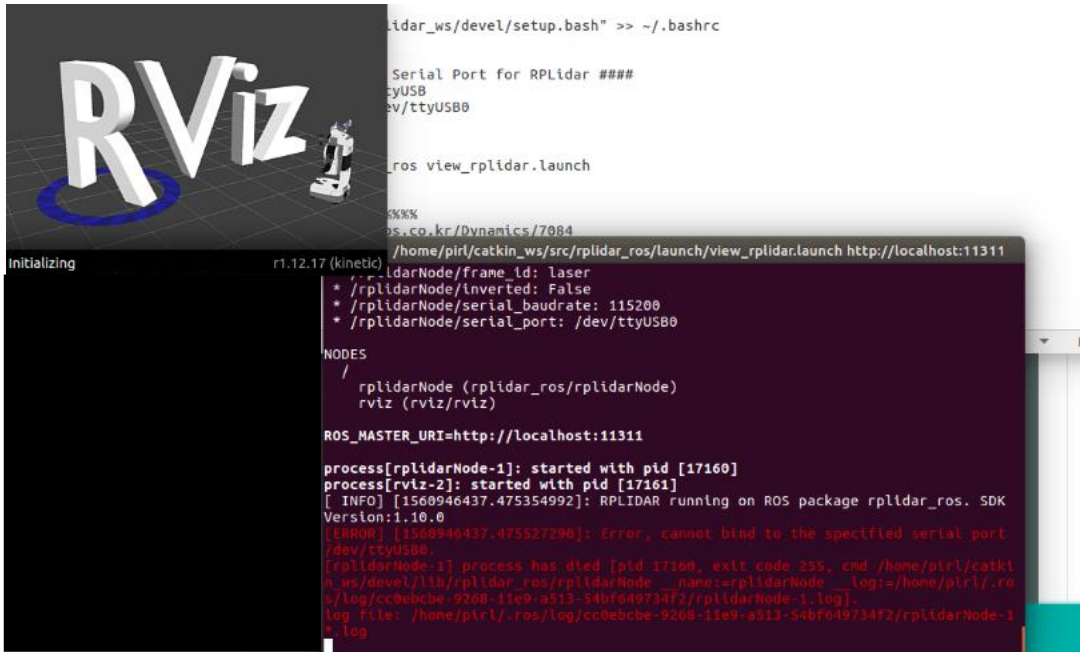
```
sudo apt-get install git

### Clone the ROS node for the Lidar in the catkin workspace src dir
git clone https://github.com/robopeak/rplidar_ros.git

### Build with catkin
cd ~/catkin_ws/
catkin_make

### Set environment when build is complete
source devel/setup.bash

### Launch demo with rviz
roslaunch rplidar_ros view_rplidar.launch
```



위와 같은 화면이 뜨면 제대로 설치되었으므로 확인 후 RPI에 설치한다.

라이다가 donkey car내에 장착된 RPI를 통해 작동하기 때문이다.

2) RPI에 설치하기

```
cd ~
mkdir -p rplidar_ws/src
sudo git clone https://github.com/Slamtec/rplidar_ros.git
cd rplidar_ros
sudo mv * ~/rplidar_ws/src
cd ~/rplidar_ws/
catkin_make

#### 환경설정 ####
echo "source ~/rplidar_ws/devel/setup.bash" >> ~/.bashrc
source ~/.bashrc

#### Activate the Serial Port for RPLidar ####
ls -l /dev|grep ttyUSB
sudo chmod 666 /dev/ttyUSB0

#### 실행하기 ####
roslaunch rplidar_ros view_rplidar.launch
```

3.5 SLAM(Simultaneous localization and mapping)

동시적 위치추정 및 지도작성은 로봇공학 등에서 사용하는 개념으로, 임의 공간에서 이동하면서 주변을 탐색할 수 있는 로봇에 대해, 그 공간의 지도 및 현재 위치를 추정하는 문제이다. SLAM 기술에는 수십가지가 있으며 구글의 Cartographer가 유명하다. 5기의 방식을 이어받아 odometry 기능이 없는 Hector_slam을 사용하였다. Odometry는 이동하면서 자신의 위치를 측정하여 표시해주는 기능이다.

설치 방법은 ROS에 나와있으나 너무나 간단하여 다른 글을 참조하였다. http://www.modulabs.co.kr/Dynamics/7084 해당글은 라이다가 아닌 다른 장비(depth camera)를 사용한 것이므로 리눅스 명령어를 바꾸어 주어야 한다.

1) Hector SLAM

```
cd catkin_ws/src
git clone https://github.com/tu-darmstadt-ros-pkg/hector_slam.git
cd ~/catkin_ws && catkin_make
cd catkin_ws/src/hector_slam/hector_mapping/launch
```

```
nono map_default.launch
```

먼저 mapping_default.launch파일의, 내용을 수정해야 하는데 기존 내용을 주석처리하는 방법은 다음과 같다.

%%%%% 구문 변경 %%%%% 기존 구문 주석 처리는 <내용><!--내용 -->

```
두번째 줄을
<node pkg="tf" type="static_transform_publisher"
name="base_to_laser_broadcaster" args="0 0 0 0 0 0 base_link laser 100" />
세번째 줄을
<arg name="base_frame" default="base_link"/>
네번째 줄을
<arg name="odom_frame" default="base_link"/>


   <param name="pub_map_odom_transform" value="true"/>
   <param name="map_frame" value="map" />
   <param name="base_frame" value="base_frame" />
   <param name="odom_frame" value="base_frame" />
```

로 바꾸어 주고 <ctrl+ X><Y><Enter> 순서대로 실행해준다.

다음으로 tutorial.launch파일의 세번째 줄을 아래와 같이 수정해 준다.

```
<param name="/use_sim_time" value="false"/>
```

그 후 맵을 그리는 명령어는 ROS wiki에 나와 있다.

hector slam Building a map###

```
sudo apt-get install ros-indigo-hector-slam
wget http://tu-darmstadt-ros-
pkg.googlecode.com/files/Team_Hector_MappingBox_RoboCup_2011_Rescue_Arena.b
ag
roslaunch hector_slam_launch tutorial.launch
```

3.6 Speech Recognition(PODEX.py)

```
#start
print("₩n₩n============================PODEX============================")
for i in range(NUM_GUESSES):
    for j in range(PROMPT_LIMIT):
        print('전달할 위치를 말하세요 : (Classroom A, Classroom B, Office)')
        target = recognize_speech_from_mic(recognizer, microphone)
        if target["transcription"]:
            break
        if not target["success"]:
            break
        print("이해하기 힘든 단어입니다. 다시 말하세요!₩n")
    target["transcription"]="fail"
#print(target["transcription"])

destination=""
if target["transcription"].lower() in ['classroom a','cluster Bay','resume','crystal bay','classical ballet',
                                       'castlebay','class 1a','ei samay','cholesterol','restaurant','mission of a',
                                       'crescent ave','tresemme','plexaderm','cholesterol in a','collezione',
                                       'cholesterol meds','crystal renay','ristorante','cluster of a','carlos adame','ristorante','cursive a']:
    destination = "ClassRoomA"

elif target["transcription"].lower() in ['classroom b','cluster b','collateral beauty','cursive B',
                                         'class 1b','restaurant b','personal b','cholesterol be','christian p',
                                         'christian be','crest oral-B','christian be','cursive d','cressida b',
                                         'cursive b','chris adobe','crusher p','crested oggy','recipe','christina p',
                                         'price of a b','sobe','classroom','glycerin be','classic r&b','vanessa kirby',
                                         'restaurant near me','cholesterol','crisfield md','glycerin bp']:
    destination = "ClassRoomB"

elif target["transcription"].lower() in ['office','opus','officer']:
    destination = "Office"

print(destination)
```

IDE and Plugin

```
#start
for i in range(NUM_GUESSES):
    for j in range(PROMPT_LIMIT):
        print('수신자를 말하세요 : (Hanbin, Jiyoon, Dongyoung, Gwanjin, Hyojin, Enok)')
        receive = recognize_speech_from_mic(recognizer, microphone)
        #print(receive["transcription"])
        if receive["transcription"]:
            break
        if not receive["success"]:
            break
        print("이해하기 힘든 단어입니다. 다시 말하세요!₩n")
    receive["transcription"]="fail"

receiver=""
if receive["transcription"].lower() in ['hanbin','humping','ambit','ambien','hamby','humble','hyundai','camden','humbling','hamden']:
    receiver = "Hanbin"

elif receive["transcription"].lower() in ['jiyoon','chi you','chilling','chillin','children','june','chia','ji-eun','zhiyun']:
    receiver = "Jiyoon"

elif receive["transcription"].lower() in ['dongyoung','polio','coil','pony o','pocoyo','perennial','daniel','twill','boil','poem','young','tuile']:
    receiver = "Dongyoung"

elif receive["transcription"].lower() in ['gwanjin','function','ogden','panzon','conversion','continment','pungent','kwanzan','anjin']:
    receiver = "Gwanjin"

elif receive["transcription"].lower() in ['hyojin','frozen','fortune','pigeon','putin','busan','who is it','14','who''s it','prison','hilton']:
    receiver = "Hyojin"

elif receive["transcription"].lower() in ['enok','anal','edel','able','girdle','hello','hazel']:
    receiver = "Enok"
```

python Podex.py로 전체 모듈이 실행된다. 사용자가 마이크로 목적지와 수신자를 말하면 들어온 음성을 인식해 destination변수에 목적지(ClassRoomA, ClassRoomB, Office)를 입력, receiver 변수에 수신자 이름(hanbin, jiyoon, dongyoung, gwanjin, hyojin, enok)이 입력된다

```
    print("Destination : "+destination)
    print("Receiver : "+receiver)

    f=open("destination.txt","w")
    f.write(destination)
    f.close()

    f=open("receiver.txt","w")
    f.write(receiver)
    f.close()

os.system("./darknet detector demo data/obj.data yolo-obj.cfg backup/yolo-obj_10000.weights http://192.168.0.10:8090/?action=stream -i 0")
os.system("python Facial_Recognition_Part3.py")
```

해당 destination과 receiver 변수를 text파일로 저장하고 목적지 인식모듈(Yolo)와 수신자 인식 모듈(Face_detection)이 실행된다.

3.7 YOLOv3 – Real Time Object Detection

택배배달에 있어 장소인식을 위하여 3가지(ClassRoomA, ClassRoomB, Office)의 데이터 셋을 각각 300장씩 수집한 후 YOLO mark를 이용하여 이미지에서 트레이닝을 할 부분을 지정 및 학습을 실시.

이미지 분류를 위해 신경망 체계 이미지 분류 프레임워크인 DarkNet을 설치 후 진행.

라즈베리파이(ip: 192.168.0.10)에 설치된 웹캠카메라를 이용하여 목적지 인식 실행.

```
/*insert code*/
char buffer[20];
FILE *fp = fopen("destination.txt","r");
fgets(buffer, sizeof(buffer), fp);
printf("%s\n",buffer);
fclose(fp);
if( !strcmp(names[class], buffer) ){
    bBox=1;
    //printf("start\n");
    if(html_fd = fopen("/home/pirl/darknet/test.txt","w"))
    {
        //printf("file open success3\n");
        printf("left : %d\nright : %d\ntop : %d\nbot : %d\n\n",left,right,top,bot);
        fprintf(html_fd,"name : %s\n",names[class]);
        fprintf(html_fd,"left : %d\nright : %d\ntop : %d\nbot : %d\n\n",left,right,top,bot);
        fclose(html_fd);
    }
    else{
        //printf("file open fail4\n");
    }
    exit(0);

}
/*end code*/
```

그림 1 darknet/src/image.c 파일 수정 : 음성입력하여 인식된 값을 destination.txt.에 저장하고 Yolo 모듈에서 이를 읽어들여 인식된 names[class]와 일치하면 종료exit(0) 한다.

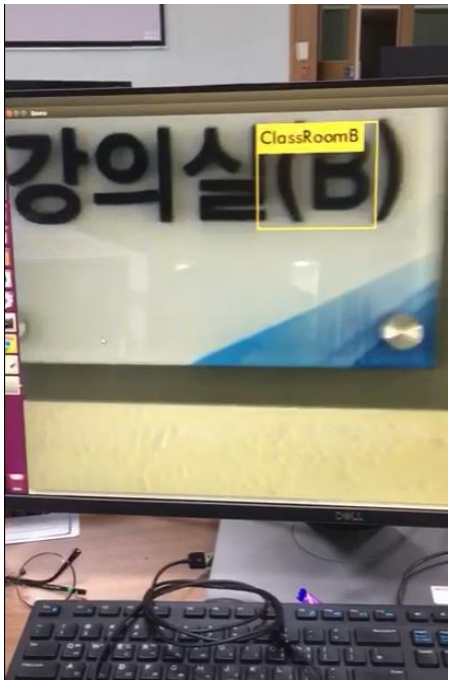


그림 2 이미지 인식모듈로 ClassRoomB 인식

3.8 OpenCV haarcascade_frontalface detection

Harr-like feature는 단순 합 이미지를 이용 특징 값을 표현하는 것이다. 위치, 크기, 모양에 따라 수많은 형태를 가질 수 있다. 이를 AdaBoost 학습 알고리즘을 사용하여 약한 인식자들을 합쳐 강인한 인식자의 그룹으로 만든다. 이때 생성된 특징 값은 얼굴의 특징을 잘 포용하는 장점을 가진다. 계산 방식이 단순 합이기 때문에 동영상에 적용이 매우 용이하며 1초안에 얼굴 이미지 데이터를 추출하는 것이 가능해진다.

```
cap = cv2.VideoCapture("http://192.168.0.10:8090/?action=stream")
count = 0

while True:
    ret, frame = cap.read()
    if face_extractor(frame) is not None:
        count+=1
        face = cv2.resize(face_extractor(frame),(200,200))
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)

        file_name_path = 'faces/user'+str(count)+'.jpg'
        cv2.imwrite(file_name_path,face)

        cv2.putText(face,str(count),(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
        cv2.imshow('Face Cropper',face)
    else:
        print("Face not Found")
        pass

    if cv2.waitKey(1)==13 or count==300:
        break
```

그림 3 얼굴 인식모듈 (허가되지않은 사용자) : Package Locked

그림 4 얼굴인식 모듈(허가된 사용자) : Package Unlocked

```
while True:
    #카메라로 부터 사진 한장 읽기
    ret, frame = cap.read()
    # 얼굴 검출 시도
    image, face = face_detector(frame)
    try:
        #검출된 사진을 흑백으로 변환
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        #위에서 학습한 모델로 예측시도
        result = model.predict(face)
        #result[1]은 신뢰도이고 0에 가까울수록 자신과 같다는 뜻이다.
        if result[1] < 500:
            confidence = int(100*(1-(result[1])/300))
            # 유사도 화면에 표시
            display_string = str(confidence)+'% Confidence it is user'
#         cv2.putText(image,display_string,(100,120), cv2.FONT_HERSHEY_COMPLEX,1,(250,120,255),2)
        #75 보다 크면 동일 인물로 간주해 UnLocked!
        if confidence > 85:
            cv2.putText(image, "Package Unlocked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
            cv2.imshow('Face Cropper', image)
        else:
            #75 이하면 타인.. Locked!!!
            cv2.putText(image, "Package Locked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
            cv2.imshow('Face Cropper', image)
    except:
        #얼굴 검출 안됨
        cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 2)
        cv2.imshow('Face Cropper', image)
        pass
    if cv2.waitKey(1)==13:
        break
cap.release()
cv2.destroyAllWindows()
```

3.9 PODEX 구동 방법

1) Donkey Car 제어

- Master PC에서 2개의 터미널을 open
- 1번 터미널에서 Donkey Car 라즈베리 파이로 ssh를 이용해 접속
- 1번 터미널에 roslaunch donkey_llc keyboard_demo.launch 입력
- 2번 터미널에 rosrun teleop_twist_keyboard teleop_twist_keyboard.py 입력

2) SLAM 구동

- Master PC에서 3개의 터미널을 open
- 1번 터미널에서 Donkey Car 라즈베리 파이로 ssh를 이용해 접속
- 1번 터미널에 roslaunch rplidar_ros rplidar.launch 입력해 Lidar 작동
- 2번 터미널에 rosrun rplidar_ros rplidarNodeClient 를 입력해 Lidar에서 입력 받은 값을 불러옴
- 3번 터미널에 roslaunch hector_slam_launch tutorial.launch 를 입력해 Lidar에서 입력 받은 값을 rviz로 시각화.

4. 개선방향 및 활용방안

4.1 한계점

- RPLidar와 SLAM을 활용하여 Mapping을 한 후 지도상에서 지정된 위치로 donkey car를 이동시키기 위해 IMU를 사용하고자 하였으나 SLAM을 이용한 Mapping, 사물인식, 얼굴인식에 많은 시간이 허비되어 실현시키지 못하였다.

여기서 IMU는 차량의 세가지 선형가속 구성요소와 세가지 회전속도 구성요소(6-DOF)를 직접 측정하는 장치이다.

- RPLidar, donkey car 배터리, 라즈베리파이 등 많은 장치들을 차체에 설치하다 보니 차량의 무게가 많이 나가고 차체의 높이가 높아 차량운행에 불편함이 있었다.

- ROS의 특성상 Mater PC와 Client PC가 동일한 공유기로 연결되어 있어야 하기 때문에 가동가능범위가 제한적이다.

4.2 개선방향

- IMU 또는 다른 방법을 이용한 자율주행
- 더 큰 RC Car를 사용한 더욱 안정적인 운행 및 물품적재 공간 구성
- 배송물품을 안전하게 배송할 수 있는 Lock기능 설비된 하드웨어
- 전방에 장애물이 나타났을 때 정지 또는 회피하는 기능

4.3 활용방안

자율주행과 네비게이션 기능이 구상한 대로 적용된다면 실내에서 가벼운 물건 또는 서류 등의 물품 배송이 가능 해지고 WiFi가 아닌 실외 통신장치가 있다면 건물과 건물 간에 물품전달을 실현하여 일손 절감에 도움을 줄 것으로 보인다.

전세계 적으로 노령화가 가속화되고있는 가운데 자율주행 택배를 이용한다면 마트 또는 상점에서 물품을 구입 후 자율주행 택배를 이용하여 구매물품을 집까지 배송하는 서비스를 제공할 수 있으며 또한 무인 택배를 운영하며 인건비도 절약할 수 있을 것으로 생각된다.

5. 조원 평가

1) 김동영에게 팀원들이

- 김에녹 : 항상 밝고 긍정적인 모습으로 팀의 분위기를 이끌어가준 고마운 팀원. 팀원들을 다독여주며 '좋게 좋게', '쉽게 쉽게'를 추구하는 '즐겜러'. 끝없이 제시하는 아이디어 덕분에 새로운 인사이트를 발견하고 프로젝트의 방향을 잡을 수 있게 해준 팀원이었다. 그리고 과외 경험이 많아서인지, 무엇이든 물어보면 기초적인 개념부터 차근차근 설명해주던 3조의 선생님.
- 문한빈 : 비전공자임에도 불구하고 빅데이터 프로젝트를 진행하는데 많은 도움을 줬습니다. 회의때마다 창의적인 의견제시로 핵심적인 방향을 제시했고, 특히 빅데이터 프로젝트 암세포 데이터 분석에 많은 기여를 했습니다.
- 백관진 : 이미 취업한 상태라 놀 법도 한데 팀원에게 도움이 되기 위해서 열심히 했다. 시간을 효율적으로 사용하며, 수학적인 부분에서 뛰어난 면모를 보였다. 다른 사람에게 설명하는 것을 매우 잘했다.
- 전지윤 : 항상 밝고 긍정적인 마음으로 팀원들과 잘 지내며 자신이 맡았던 일들을 잘 해주었습니다. 특히 AI 프로젝트 발표를 맡아 저희들의 결과물을 잘 표현해준 것 같습니다.

- 최효진 : 비전공자임에도 불구하고 항상 학업에 열중하며 다른 조원들이 모르는 것을 알려주는 모습이 정말 인상 깊었습니다. 많은 아이디어를 내는 아이디어 뱅크로 프로젝트에서 항상 밝은 모습으로 팀 분위기를 만들어줘서 고마웠습니다.

2) 김에녹에게 팀원들이

- 김동영 : 팀이 힘들 때 중심을 잡아준 동료입니다. 힘든 일이 있을 때 발벗고 나서서 스스로 해내며 동료들에게 할 수 있다는 희망을 심어주었습니다. 마지막까지 최선을 다해 최우수 조로 뽑히는데 큰 역할을 해주어 정말 고마움을 느끼는 동료입니다.
- 문한빈 : 교육 과정동안 실질적인 조의 리더로서 책임감을 다하기 위해 노력하는 모습을 보였습니다. 제 역할을 하지 못한 팀장을 대신해 팀 프로젝트를 이끌어 가기 위해 최대한 노력한 사람이었습니다. 맡은 역할보다 더 많은 기여로 팀 활동에 정말 많은 도움이 됐고, 저 또한 이런 모습에 자극 받아 팀 활동에 적극적이게 되었습니다.
- 백관진 : 조에서 뛰어난 인재. 본인은 아니라고 하지만 열정이 넘쳐서 워커홀릭적인 면모를 보였다. 주도적으로 나서 많은 문제를 해결해 존재감이 돋보였다. 언론을 전공해서 그런지 글 쓰는 것이나 과제 이해도가 제일 높았다.
- 전지윤 : 가장 열정적이고 최선을 다했던 조원이라고 생각합니다. 빅데이터 프로젝트에서는 데이터 분석과 PPT작성 및 발표까지 많은 업무들을 도맡아 성실히 잘 수행해 주었고 AI프록젝트에도 대부분의 과정을 혼자 수행하며 조를 이끌어 주어 최우수상이라는 성과를 이루는데 큰 기여를 했다고 생각합니다. 앞으로도 계속 연을 이어가고 싶은 친구입니다.
- 최효진 : 조장을 도와 프로젝트의 방향성을 잡는데 큰 역할을 해주었습니다. 비전공자임에도 불구하고 많은 열정을 쏟아 최선을 다하는 모습이 아주 인상적이며 고마웠습니다. 체력적으로 많이 힘들었을 텐데 끝까지 밝게 임해줘 고맙고 책임감은 꼭 본받고 싶습니다.

3) 문한빈에게 팀원들이

- 김동영 : 컴퓨터공학 전공자로써 프로젝트를 진행할때 많은 노력과 실제 매우 중요한 역할을 해 주었습니다. 많은 일이 주어짐에도 굳은 말 없이 일을 함이 멋있는 동료였습니다.
- 김에녹 : B반의 다른 조들이 탐내는 3조의 개발자. 개발자가 프로젝트에 '갈린다'라는 말이 무엇인지 몸소 보여준 팀원. 하나를 맡으면 2개 3개를 더 해오고, 50을 요구하면 100, 200으로 완성해오는 3조의 노예 '핫산'. 자신이 맡은 부분을 완벽하게 구현하기 위해 밤을 새며 작업하는 모습을 볼 때마다 너무 무리한 파트를 맡긴 것 같아 미안하기도 했다. 그러

나 잠을 깨기 위해 춤을 추고 피아노를 치는 모습을 보면 아직 덜 힘든 것 같아 미안한 마음이 싹 가시게 만드는 재미있는 팀원. 문한빈에게 피아도 한 대와 트와이스의 노래, 약간의 전화시간만 주어진다면 완수하지 못할 프로젝트가 없을 것이다.

- 백관진 : 타고난 프로그래머로 교육 초반, 프로그래밍 교육 때부터 많은 도움을 받았다. 시간이 지나면서 별병도 많아지고 맡은 일도 많아졌다. 프로그래밍 적으로 어려운 문제는 다 해결했고, 웹 개발과 얼굴인식 등 컴퓨터공학 전공자가 할 수 있는 일들을 도맡아 했다.

- 전지윤 : 빅데이터 과정에서는 웹서버 구축을 하며 팀을 많이 도와주었고 AI 프로젝트에서는 컴퓨터공학과 출신 답게 기초지식을 바탕으로 팀원들에게 조언을 주며 막히는 부분들을 해결해주었습니다. 함께 있으면 웃을 일이 많은 친구입니다.

- 최효진 : 컴퓨터공학도로서 팀 내의 다양한 소프트웨어 문제를 해결했습니다. 전공 지식과 경험으로 기술적인 방향을 제시했습니다. 항상 긍정적인 마인드로 프로젝트를 진행하며 많은 도움이 됐고, 큰 역할을 했습니다. 그리고 다른 조원들이 낙관적으로 생각하는 부분에 대해서 냉철한 시각으로 분석하고 방향을 조정할 수 있도록 도와준 팀원이었습니다.

4) 백관진에게 팀원들이

- 김동영 : 팀의 맏형으로서 정신적 기둥같은 역할을 해 주었습니다. 그리고 꾸준히 노력하고 늘 좋은 말들을 해준 동료입니다.

- 김에녹 : 팀의 맏형. 말없이 묵묵하게 자신이 맡은 것을 다해준 고마운 팀원. 다른 팀원들이 집에 갈 때까지 남아서 팀원들을 다독여준 고마운 조장. 보이지 않는 곳에서 우리 조가 필요한 것들을 챙겨준 형님. 비전공자임에도 불구하고 끝까지 함께 달려와 준 형에게 감사하다.

- 문한빈 : 많은 프로젝트에서 팀장으로 많은 기여를 했습니다. 하지만 팀 회의에서 팀원들이 다양한 의견을 제시할 때 갈등중재자로써 역할이 너무 부족했습니다. 교육과정 내에서 한 여러 팀 활동 경험으로 부족했던 부분을 되짚어 보고 리더십을 키웠으면 합니다.

- 전지윤 : 매일 늦게까지 남아 열심하 하는 모습이 인상적이었습니다.

- 최효진 : 조장의 역할로 많은 기여를 해준 것에 감사하며 프로젝트에도 열중하고 학업에도 열중해 많은 도움을 줬습니다. 조장의 역할을 잘 해내 주어서 팀의 분위기를 잘 잡아준 것 같습니다

5) 전지윤에게 팀원들이

- 김동영 : 늘 웃는얼굴로 팀의 분위기를 좋게 만들어준 동료입니다. 덕분에 재미있는 교육 과정을 보낼 수 있었습니다.

- 김에녹 : 주어진 상황에서 맡은 일을 최선을 다해 완수하는 팀원. 비전공자임에도 불구하고 프로젝트에 자신이 맡은 부분을 해결하기 위해 개인적으로 많은 시간을 투자하는 모습이 인상 깊었다. 어려운 프로젝트임에도 불구하고 다양한 시도를 하고 이해하려는 모습, 그리고 자신이 어려운 것을 맡아서 하려는 모습이 멋있었다. 팀원들이 지쳐갈 때면 툭툭 내뱉는 유머러스한 말들로 팀의 사기를 높여주는 것은 물론, 아침저녁으로 차량운행을 해주며 팀원들의 발이 되어준 고마운 친구.

- 문한빈 : 조별활동에 많이 노력한 조원 중 하나로 AI 프로젝트 때는 비전공자임에도 최대한 팀 프로젝트에 기여하려고 노력한 사람이었습니다. 맡은 역할을 충실이 완수하고 자신이 맡은 부분을 끝내도 최대한 팀 활동에 기여할 수 있는 부분을 스스로 찾아보면서 도움을 준 고마운 동료였습니다.

- 백관진 : 호기심이 많은 조원. 이번 과정이 도움이 된다며 열심히 배워 나갔다. 외국대학을 나와 한국의 조별과제 문화를 혹독하게 겪느라 고생이 많았다. 운전하느라 늘 피곤한 모습이 안쓰러웠다. 통계학과를 나와 빅데이터 프로젝트 때 재미있어 하면서 열심히 했다.

- 최효진 : 전공자가 진행하기에도 벅찬 프로젝트임에도 불구하고 팀원에게 도움을 주려고 노력했습니다.

6) 최효진에게 팀원들이

- 김동영 : 늘 팀원들을 도와주고 싶어하는 마음이 있는 동료입니다. 깊은 동료애를 느낄 수 있었습니다.

- 김에녹 : 막내임에도 불구하고, 자신의 전공지식을 살려 다른 팀원들이 부족한 부분을 보완해주는 든든한 서포터. 특유의 미적감각으로 발표자료와 PPT를 도맡아 꾸며주던 디자이너. 그리고 무엇보다도 나이 많고 시커먼 아저씨들 사이에서도 냉철한 판단을 내려주던 3조의 실세.

- 문한빈 : 컴퓨터공학 전공자로 전공지식으로 팀 활동에 많은 기여한 팀원입니다. 웹페이지 개발할 때 디테일한 부분까지 신경 써준 덕분에 완성도 높은 웹 제작이 가능했습니다. 팀 활동에서도 다양한 의견을 종합 정리하는 데 능력을 발휘했으며, AI 프로젝트에서 맡은 역할을 충실히 수행해 팀 활동에 기여한 팀원 중 하나였습니다.

- 백관진 : 이미 다른 포스코 교육 과정을 수료하고 와서 그런지 교육 전반에 대한 이해도가 높았다. 문서작업이나 ppt 제작을 도맡아서 했다. 전반적인 서기 역할로 교수님들이

하신 말씀도 잘 정리해서 알려줬다.

- 전지윤 : 팀의 막내로서 조원들에게 힘이 되어 주었습니다.