Report Date: 11/18/2022
To: ematson@purdue.edu, ahsmith@purdue.edu and lee3450@purdue.edu
From: C.C

- Eunmin Kim (32200928@dankook.ac.kr)
- Booyong Lee (201810909@sangmyung.kr)
- Hanbyeol Lee (yhb1834@cau.ac.kr)
- Jeeyoung Oh (jeeyoung9907@cau.ac.kr)
- Seoyeong Lee (lsyoung66@cu.ac.kr)

Summary

Communication between two RPIs was implemented through ROS2. Every node was implemented too.

What C.C completed this week:

- Wrote methodology and implementation in paper
- Drew the circuit of the driving part
- Made an elements specification table
- Tried to run multiple proximity sensors simultaneously
- Wrote obstacles_detection node using RPLiDAR and proximity sensors to subscribe /scan topic and publish /Stop topic
- Trained a YOLO v7 model with COCO dataset and TACO dataset
- Succeeded in implementing motor publisher and subscriber
- Implemented ROS2 communication with two other RPi
- Fixed the dependency problem between modules by adjusting python environment path
- Built an Android application that shows current location using Google Map API

Things to do by next week

- Will complete the detection node and training YOLO v7 model

Problems or challenges:

- It was hard to run 5 proximity sensors simultaneously due to the method we connected pins and Raspberry Pi.
- Setting the path of ROS had some troubles.
- Making workspaces is bad for running executable files so we deleted all the packages and workspaces for several times.
- Spend lots of time on setting environment variable and apply it.
- It is hard to combine various sensors' codes due to dependency and QoS problems.

References