



To: Professor Pisano

Team: 10

Date: 2/27/2025

Subject: Second Prototype Testing Report

---

**Second Prototype Test Plan**

**Draper Convoy**

**CONVOY**

**By**

**Team 10**

**Team Members**

Sadman Kabir: kabirs@bu.edu

Benjamin Hsu: bhsu25@bu.edu

Nikhil Krishna: nikhilk@bu.edu

Hugo Silmberg: hsilmber@bu.edu

Michael Moran: mmoran3@bu.edu

## **Required Materials**

### Hardware:

- Desktop Computer running Ubuntu
- Physical Turtlebot4 Lite:
  - OAK-D-LITE (Camera)
  - RPLIDAR-A1 (Lidar)
  - Raspberry Pi 4B
- Banana Stick

### Software:

- ROS2 installed on the desktop computer
  - Using ROS2 Jazzy distribution
  - Allows data to be accessed through TurtleBot sensors and communicate with other software
- Person detection node
  - YOLO object detection algorithm
    - Pretrained model from Ultralytics
  - Identifies human beings detected within camera view and outputs information about their position relative to the robot
  - Highly accurate, even from long distances and figure partially cut off
- Motor control node
  - Accepts data from the person detection node and outputs information to control the TurtleBot's motors

## **Set-Up**

1. Remove Turtlebot from the dock and make sure it starts up correctly. Restart the robot if it does not start up right the first time.

2. On the desktop, setup a new terminal environment and launch our nodes using the launch\_robot.sh bash script:

```
cd ~/ros2_ws
chmod +x launch_robot.sh
./launch_robot.sh
```

3. Once all nodes are fully started, simply press P in the motor control node terminal to start human tracking.

## **Pre-Testing Set-Up Procedure**

1. Ensure the terminal environment is set up.
2. Ensure Turtlebot4 and all its topics and services start up correctly.
3. Ensure all nodes are launched together.

## **Testing Procedure**

- Place a human model in the simulation and turn the TurtleBot to identify the human.
- The person detection node identifies the human and sends data to the motor control node.
- The TurtleBot begins to move toward the human model.
- Move the human model around the simulation to demonstrate real-time identification and data processing.
- Move the human model outside of the camera's frame and test relocation functionality.
- Controls:
  - P: Autonomous driving mode + relocation
  - U I O; J K L; M , . ; Remote control driving
  - anything else: stop

### **Measurable Criteria**

- I. The person detection node should be able to correctly identify a human being within the camera's frame, display a bounding box around the detected person, and place a centroid in the center of the bounding box.
- II. The person detection node should display the confidence level of the detection near the bounding box, and publish data about the person's distance and angle relative to the center of the camera's frame.
- III. The motor control node should successfully subscribe to the person detection node and accurately retrieve the data published from it. The motor control node should respond to this incoming data and output data to the TurtleBot's motors to move toward the detected human being, with the robot stopping when it gets within a defined distance.
- IV. If the person moves out of the camera's view, a button can be pressed and the TurtleBot should rotate in an attempt to relocate the human being.

### **Score Sheet**

<b>Functionality</b>	<b>Working? (Y/N)</b>
Identifies a banana target	
Bounding box around model w/ confidence level, angle calculation and centroid	
Seek banana target	
Stop when close enough to the banana target	
Auto relocate -> continue seeking banana	