Operation Manual

Project Overview

Our project consists of a system of components to autonomously drive a 1/5th scale car from the front of the COE Building to the back of the AME building. This system relies on an intel RealSense camera to map the environment, which is then processed through the NVIDIA Jetson processor to design an optimal path to reach the goal destination.

Component/Module Description

- NVIDIA Jetson
  - The NVIDIA Jetson serves as the main processor to compute and receive signals from the intel RealSense camera. It houses all the software required to autonomously move the vehicle while detecting unexpected obstacles. The NVIDIA Jetson uses two main packages to achieve this which are the Visual SLAM and NVBLOX packages. These packages used in correlation with the SBMPO allow for the independent autonomous control of the vehicle.

- Electronics Bay
  - The Electronics bay is used to house all electronic components required for autonomous control. It holds the NVIDIA Jetson, a Power Distribution Block, 2 Motor Controllers, and a 12V Voltage Regulator. The Electronics bay serves as a secure way to hold multiple components in the vehicle while preventing any unexpected damage that may occur due to a collision or bump in the road.

- Voltage Regulator
  - The voltage regulator is used to control excess voltages coming through from the main battery which will ensure that the jetson does not receive any excess voltage.

- NVIDIA Mount
  - Holds and secures the NVIDIA Jetson in place during testing and operation in case severe failure occurs. It is built with plates and 4 screws to prevent the NVIDIA from overheating.

- Intel RealSense Camera

- The intel real sense camera Provides Live video feed for NVIDIA Jetson. It is used to build a software map of the environment and for detection of obstacles.

- Telescoping Camera Mount
  - The function of the telescoping camera mount is to hold and secure the intel RealSense camera in place. It also functions such that when the vehicle is powered on, the camera will automatically lift from under the front hood of the vehicle and begin surveying and mapping the environment upon start up.

- Arduino Teensy
  - The Arduino Teensy allows for use of multiple PWM pins since the NVIDIA Jetson only provides 2 pins. This will allow for the control of multiple PWM cycles which is essential in controlling the appropriate wheel angle turn, acceleration of the vehicle. The Arduino Teensy is essential in ensuring that changes in acceleration and wheel angle turn are done smoothly instead of quick rapid changes.

Integration

- NVIDIA
  - Place the mounting tray in the frunk and put the NVIDIA on the tray. Place the mounting cover on the NVIDIA and secure it with 4 screws.
- Electrical Wiring
  - Multiple components are wired together. The Input of data flows from the Intel RealSense Camera, which is connected to the NVIDIA Jetson via USB-C Cable. The software of the NVIDIA Jetson will depend upon the Arduino Teensy that is powered via a micro-USB cable in the NVIDIA Jetson. The Arduino Teensy will output 8 PWM signals that are wired throughout the vehicle to the appropriate motor controllers to move and turn the car. The point of integration with team 503 begins with the connection of these PWM signals into the motor controllers of the vehicle.
- Motor Drivers
  - Attach the motor drivers in the mounting tray inside the frunk of the trunk

Operation

- Using the NVIDIA
  - Powering
    - The NVIDIA can be powered at 7V-19V, and has 3 wattage settings of 15W, 30W, and 60W. It is very important not to exceed these voltages or power levels because the processor can be damaged.
  - Logging in
    - Username: nvidia
    - Password: nvidia

        - You should update the system prior to using any packages using the command: $ sudo apt update && sudo apt upgrade -y
    - Docker / ROS2
        - $ ~/run_project.sh
        - $ ros2 launch sd504 sd504.launch.py
- Powering the electronics bay
    - The electronics bay and all its components are powered by a 16-pin connector to the inner car. 2 pins are dedicated as V+ and V- from the battery, 8 pins are dedicated for the PWM signals to the motors, and 4 pins are dedicated for the steering motor encoder signals.
- Software
    - https://github.com/JTylerBoylan/Senior-Design-2023
    - If changes are needed in the software of the car, you can fork this repository and clone your fork into the NVIDIA's workspace. Then you'll need to build the workspace, and update the ros2 launch file to include your package.

Troubleshooting

- NVBLOX
    - https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nvblox
    - https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nvblox/blob/main/docs/tutorial-nvblox-vslam-realsense.md
    - https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nvblox/blob/main/docs/troubleshooting-nvblox-vslam-realsense.md
- SBMPO path planning
    - https://github.com/JTylerBoylan/SBMPO

Appendix