This Repo can be found at https://github.com/benjamine07/Raspberry_Pi_Group

Assignment Overview

Homework 6 involves the creation of Robotic Operating System (ROS) nodes with the end goal of controlling an RC racecar and recording data from the vehicle. The key tasks are:

- Create three ROS nodes to communicate with each other of the same LAN

1. R Pi 1 (standalone): connect a USB gamepad/controller and publish button/joystick data with a ROS node
2. R Pi 2 (attatched to racecar): create a second ROS node that subscribes to button/joystick data and creates a corresponding PWM signals to control racecar throttle and steering. This should also *publish* data from the attached camera.
3. Ubuntu VM: the third ROS node subscribes to all published data from the car and controller nodes, acting as a data logger.
4. throttle and steering should be saved as a CSV via pandas library (python)
5. include sample data collected in a CSV file (will be in sample_data folder)

- Convert this README.md into a PDF to be submitted
- Document the assignment in a GitHub Repo

How to Create a ROS package

Linux

adapted from https://www.ros.org

Once ROS is installed on your system (see here for instructions) you need to create a ROS package:

Create a workspace in any location of your choosing and move your working directory to it. your home directory is a good place for most cases.

```
mkdir ~/catkin_ws
cd ~/catkin_ws
```

Now initialize the ROS package workspace:

```
catkin_make
source devel/setup.bash
```

Create a new ROS package with name "my*ROS*pkg" and re-init the workspace

```
catkin_create_pkg my_ROS_pkg std_msgs rospy roscpp
catkin_make
```

Notes About the Racecar Setup

The Racecar is a Traxxas 4WD RC car. The motor ESC and steering servo are controlled by an on-board Raspberry Pi 3b (R Pi). The R Pi generates the PWM signals needed by the ESC and servo via a hardware module that communicates with the R Pi via I2C. The software is several ROS nodes used for development of the vehicle control software.