

Lab 2: Inside AWS DeepRacer

Introduction:

The goal of this lab is to introduce students to the physical DeepRacer Evo bot and give them a better understanding of its inner workings. This lab will give a brief overview of the DeepRacer core application and its Robot Operating system as well as give students some hands-on experience with programming the DeepRacers motors and servo to achieve movement.

Although the DeepRacer's default functionality is to complete laps around a track, you are also able to create projects, such as Follow the Leader. This lab also aims to give students insight on creating projects for the DeepRacer, besides racing a track, and enable them to create their own project if desired.

DeepRacer Core:

The DeepRacer Core application is comprised of a series of ROS (Robot Operating System) nodes in charge of defining different functionalities of the bot. These nodes are then launched using the DeepRacer Launcher ROS package when starting up the DeepRacer bot. The nodes are split into four categories: Perception, Decision, Navigation, and Application Support:

Perception:

Controls for sensors used to perceive the environment around the DeepRacer:

- camera_node: reads and publishes data from the camera(s)
- sensor_fusion_node: combines camera and LiDAR data

Decision:

Makes performance decisions using the data from the above nodes:

- inference_node

Navigation:

Controls for motors and servo to enable movement:

- DeepRacer_navigation_node: sends signals to motors and servo to complete the movements specified from the inference node
- Servo_node: uses signals sent from above and converts them to raw PWM values for motor and servo control

Application Support:

As the name implies, these nodes are used to make sure the DeepRacer is functioning as intended:

- Ctrl_node:
- Software_update_node:
- Model_loader_node
- Otg_control_node
- Network_monitor_node
- Device_info_node
- Status_led_node
- Usb_monitor_node
- Webservice_publisher_node

These nodes can also be utilized when creating your own project with DeepRacer. More on that later on...

More info on DeepRacer ROS nodes can be found here: http://wiki.ros.org/aws_deepracer

Part 1: Pulse Width Modulation

Pulse Width Modulation (PWM) is a technique used to control the amount of power delivered to electronic devices such as motors, lights, and speakers. The method involves generating a square wave with a fixed frequency and varying the duty cycle of the waveform, which represents the percentage of time that the wave is in the "on" state compared to the "off" state.

By changing the duty cycle, the average voltage or current delivered to the device can be varied, which allows for control of the device's speed, brightness, or volume. PWM is commonly used in electronic speed controllers for motors, LED lighting systems, and audio amplifiers.

The frequency of the PWM waveform is typically in the range of a few hundred Hz to several kHz, and the duty cycle can be adjusted by changing the pulse width, either manually or through an electronic circuit or microcontroller. We will see more on this in the activity below:

Activity: Explore pwm.sh file








What sensors, motors, etc are being controlled within this file?

Which GPIO pin is used for pwm?

How is PWM enabled/disabled?

Which pwm module is used for which device? (ex: pwm0)

Below is what is within each pwm module folder:

 power		File folder
 capture	4,096	File
 duty_cycle	4,096	File
 enable	4,096	File
 period	4,096	File
 polarity	4,096	File
 uevent	4,096	File

What does the duty cycle control?

If I wanted to turn the servo a little less right than defined in the `servo_ctrl()` function, what would I assign the duty cycle to?

If I wanted to turn the motor so that it went back just a little bit faster than the “Normal” speed that is defined in the `motor_ctrl()` function, what would I assign the duty cycle to?

Test out the different functions defined in the `pwm.sh` file. Be sure to have to DeepRacer elevated so that the wheels are not touching the ground and will be able to turn without the bot actually moving. Please do not go over the max speeds defined in the file and be careful as the wheels may turn fast. (Suggestion: start with the slow commands for the motors and go slower or faster from there with whatever you are comfortable with)

```
Init motor and servo power and duty cycle:
sudo ./pwm.sh enable

Disable motor and servo power
sudo ./pwm.sh disable

Motor control Commands:
sudo ./pwm.sh motor fw           //Forward
sudo ./pwm.sh motor fw slow     //Slow Forward
sudo ./pwm.sh motor fw max      //Max Forward
sudo ./pwm.sh motor stop        //Stop
sudo ./pwm.sh motor bw          //Backward
sudo ./pwm.sh motor bw slow     //Slow Backward
sudo ./pwm.sh motor bw max      //Max Backward
sudo ./pwm.sh motor [duty_cycle] //Custom motor duty cycle

Servo control Commands:
sudo ./pwm.sh servo left        //Turn Left
sudo ./pwm.sh servo mid         //Middle
sudo ./pwm.sh servo right       //Turn Right
sudo ./pwm.sh servo [duty_cycle] //Custom ESC duty cycle
```

Steps to gain access to the DeepRacer bot:

1. SSH into the DeepRacer
 - a. `ssh deepracer@AMSS-449K`
 - i. Password: qomjoz-qasxa0-paHtig
2. Gain root privileges (needed in order to view and use file)
 - a. Sudo su
 - i. Same password again
 - b. PLEASE DON'T DO ANYTHING TO MESS WITH THE BOT

Is there anything interesting you noticed or discovered while playing with the pwm.sh functions?

Task: Create a script using the pwm.sh functions to make the DeepRacer complete a square.

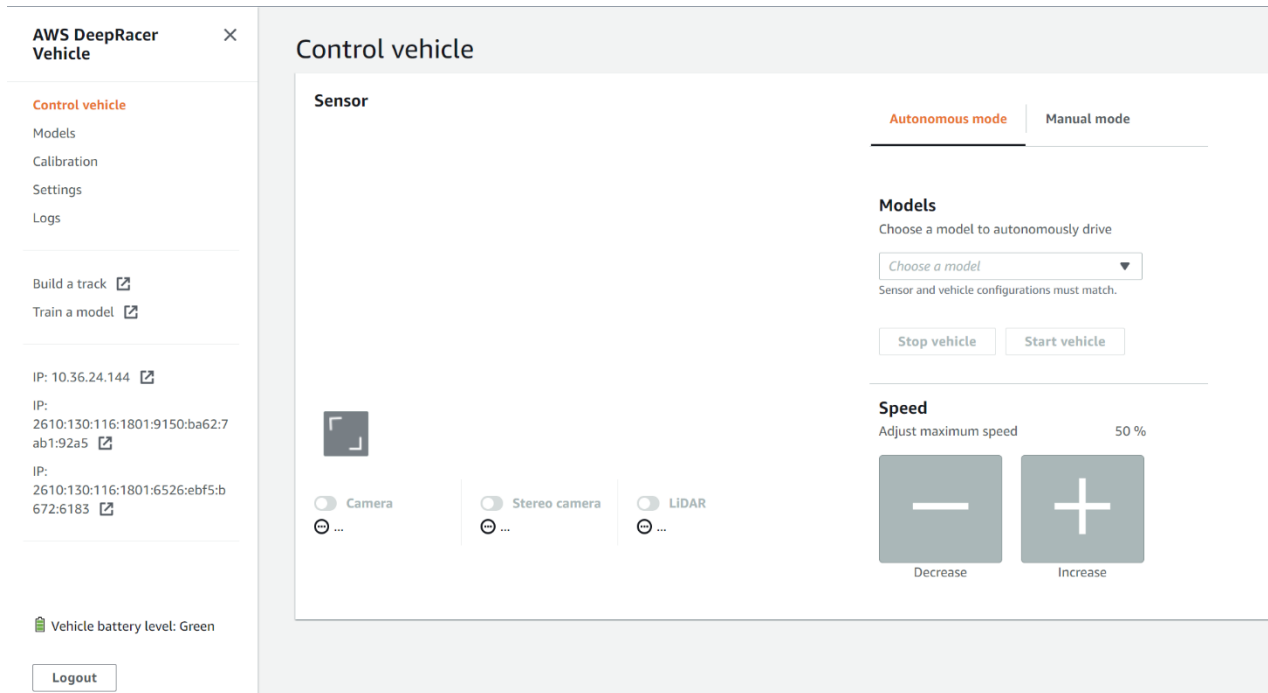
Any challenges in creating your script?

What did you learn while creating and testing your script?

Part 2: Project Experience – Follow the Leader

In order to enable Follow the Leader mode, we first have to disable/stop the deepracer core, or default functionality.

3. SSH into the DeepRacer
 - a. ssh deepracer@AMSS-449K
 - i. Password: qomj0z-qasxa0-paHtig
4. Gain root privileges
 - a. Sudo su
 - i. Same password again
 - b. PLEASE DON'T DO ANYTHING TO MESS WITH THE BOT
5. Disable/stop deepracer-core service
 - a. systemctl stop deepracer-core
6. Once Disabled, we can launch the Follow the Leader application
 - a. Everything should already be downloaded to the DeepRacer
 - b. Run these commands
 - i. Source the ROS 2 Foxy setup bash script:
 1. source /opt/ros/foxy/setup.bash
 - ii. Set the environment variables required to run Intel OpenVino scripts:
 1. source /opt/intel/openvino_2021/bin/setupvars.sh
 - iii. Source the setup script for the installed packages:
 1. source ~/deepracer_ws/aws-deepracer-follow-the-leader-sample-project/deepracer_follow_the_leader_ws/install/setup.bash
 - iv. Launch the nodes required for the FTL sample project:
 1. ros2 launch ftl_launcher ftl_launcher.py
 2. A series of messages should begin printing to your terminal
 3. Keep this terminal open, you will need it later
7. Once the Follow the Leader application is launched, we can now enable the Follow the Leader mode:
 - a. Navigate to <https://10.36.24.144/>
 - i. If there is an error just reset to DeepRacer using the Reset button next to the power button
 - b. Login using password: b21V3m4c
 - c. Once logged in, you will see a page similar to this:



- i. Next to Manual mode, there should be a Follow the Leader mode
 1. Here you should be able to start and stop the vehicle
- ii. If the DeepRacer decides not to move, you can try troubleshooting or you can look at the terminal used to launch the application
 1. Using Terminal:
 - a. Messages should be continuously flowing on the console
 - b. You should be able to see where it is writing the commands it wants to execute (ie. Forward, Reverse, Right, Left)
 - c. With the DeepRacer elevated and wheels off the ground, move your hand or body in front of the cameras
 - i. Are the commands being printed following your movements?
 2. Troubleshooting:
 - a. Possible Issues:
 - i. Motor wires are loose or laid in some way that is messing with the connection (also the battery might just be switched off)
 - ii. Calibration is incorrect

