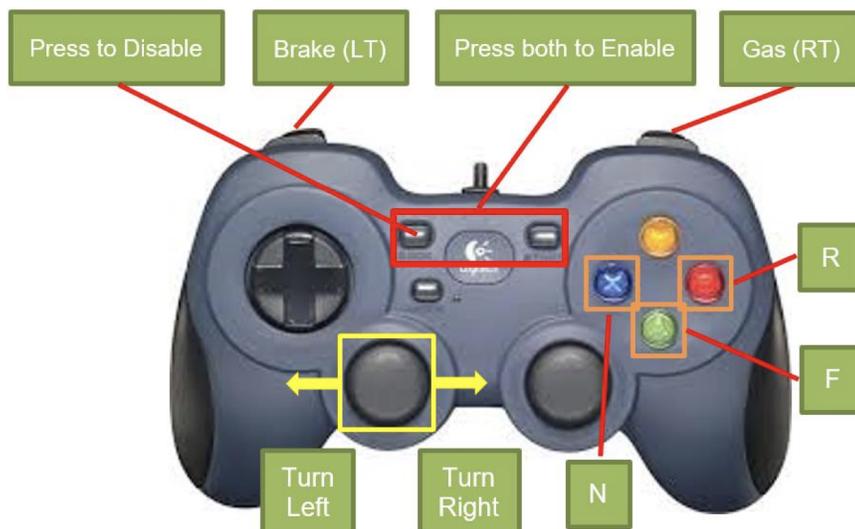


## IMPORTANT:

- When the vehicle is not in use, ensure chocks are put down and the parking brake is on!
- Ensure none of the emergency stop buttons both inside and outside of the vehicle are pressed down, otherwise Pacmod will not work. Also, the emergency stop buttons do not stop the vehicle, but disconnect the computer from the car, which is an important distinction.

## Starting up:

1. Click on the Visualization icon on the desktop.  
That should launch the Rviz software and you will see a window pop up with the camera feed on the left, and lidar feed on the right.  
**Note:** sometimes when starting up, the camera feed will be extremely dark, or have inaccurate colors. Killing and restarting the program usually can fix this.
2. Click on the dbw\_joystick icon on the desktop.  
You will see a terminal pop up. You can ignore the warning about the joystick force feedback. Pick up the joystick (make sure it is plugged in) and then hold down select and start at the same time. If the vehicle judders a little or the steering wheel turns that is normal. Pacmod should now be enabled. You can test this by clicking left or right on the directional pad to turn on the blinkers, or try turning the steering wheel by moving the left joystick left or right. The picture below should help with what the joystick buttons do and how to control the vehicle.  
**Note:** The gas pedal is not able to be controlled with the parking brake still on.  
**Note:** Safety driver can take control away from the computer/joystick at any time by turning the steering wheel or pressing the pedals. In order for computer/joystick to regain control of the vehicle, just press select and start again.



## Starting off with ROS:

Open up a terminal and cd into the demo\_ws directory. Then type:

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

Do both of these commands for every terminal you will open for running ROS/Pacmod commands in!

Type:

```
rostopic list
```

for a list of all available ROS topics to publish to

Type:

```
rostopic type rostopic_name_here
```

to see the message type for a certain ROS topic

Type:

```
rosmmsg show message_type_here
```

to see the syntax of a given type of message

To publish to a topic, the format for a command is:

```
rostopic pub [-l|-r|-1] [type] [message_type] args
```

-l to latch the message unless terminated using ^C, -r to send the message at a rate (default 10 Hz), -1 to latch the message once.

## EXAMPLES:

### 1. Controlling the turn signals

Type:

```
rostopic type /pacmod/as_rx/turn_cmd
```

It should say that the message type for the turn\_cmd topic is pacmod\_msgs/PacmodCmd.

Type:

```
rosmmsg show pacmod_msgs/PacmodCmd
```

and it should show (probably along with other items):

```
std_msgs/Header header
```

```
uint16 ui16_cmd
```

Putting it together, we get:

```
rostopic pub -l /pacmod/as_rx/turn_cmd pacmod_msgs/PacmodCmd "{header: auto, ui16_cmd: 0}"
```

This turns on the right indicator. Set ui16\_cmd to 1 to turn off the indicator, 2 to turn on the left indicator, and 3 for alternating/hazards.

### 2. Turning the steering wheel

Type:

```
rostopic type /pacmod/as_rx/steer_cmd
```

It should say that the message type for the turn\_cmd topic is pacmod\_msgs/PositionWithSpeed.

Type:

```
rosmmsg show pacmod_msgs/PositionWithSpeed
```

and it should show :

```
std_msgs/Header header
```

```
float64 angular_position
```

```
float64 angular_velocity_limit
```

Angular\_position is to set the steering wheel angle [rad] and Angular\_velocity\_limit is the rotation velocity [rad/s] to get to that steering wheel angle.

Putting it together, we can type:

```
rostopic pub -l /pacmod/as_rx/steer_cmd pacmod_msgs/PositionWithSpeed "{header: auto, angular_position: -10, angular_velocity_limit: 1}"
```

Which turns the steering wheel clockwise at 1 rad/s until it reaches the position -10.

### 3. Controlling the brake pedal

Type:

```
rostopic pub -l /pacmod/as_rx/brake_cmd pacmod_msgs/PacmodCmd "{header: auto, f64_cmd: set_number_here}"
```

To control the brake. Set f64\_cmd from 0.0 to 1.0 to set a position for the brake pedal, with 0.0 being at default state and 1.0 being fully pressed down.

### 4. Controlling the gas pedal/Moving Forwards

**Note:** this one requires a safety driver! Do not move the vehicle without a safety driver.

Release the parking brake, ensuring the stick goes all the way down. Set the gear to forwards. It is probably best to try this outside at first, so have the safety driver take the vehicle outside.

```
Type: rostopic pub -l /pacmod/as_rx/accel_cmd pacmod_msgs/PacmodCmd "{header:auto, f64_cmd: 0}"
```

To control the gas pedal. Set f64\_cmd from 0.0 to 1.0 to set a position for the accel pedal, with 0.0 being at default state and 1.0 being fully pressed down.

**Note:** 0.1/0.2 probably won't move the car because the gas pedal hasn't been pushed enough but don't necessarily jump to a higher number unless you are prepared for the car to move.