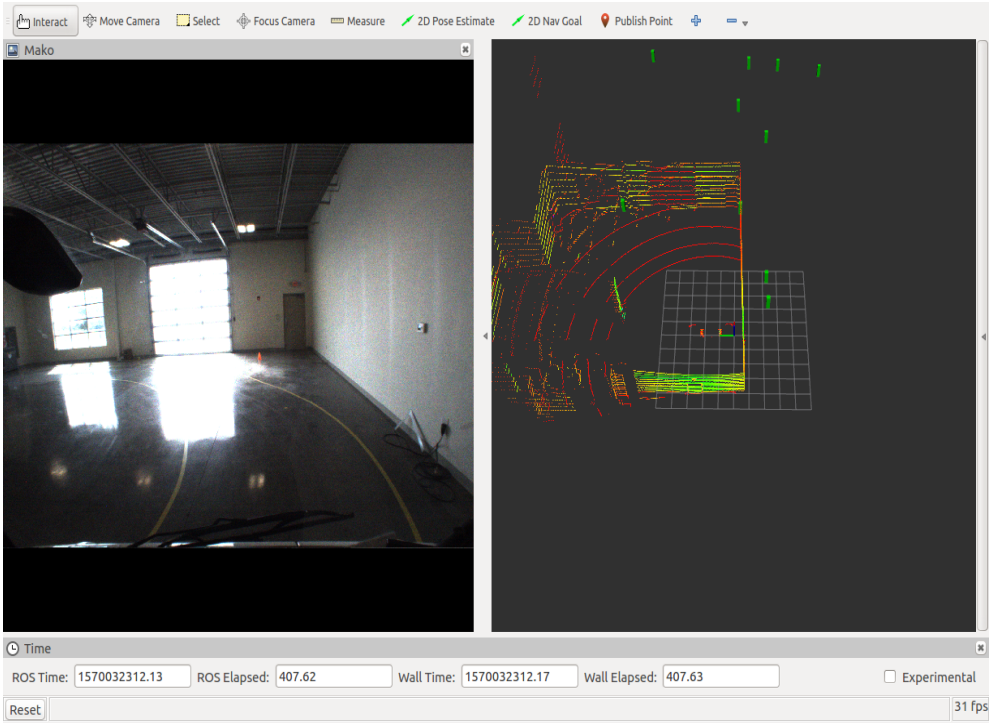
Polaris GEM e2 Camera & Driving ROS Topic Message

Camera Message:

To launch camera and other sensors. Double click the icon “Sensor Visualization” on the desktop. Alternatively, you can also use command:

$ **roslaunch basic_launch visualization.launch**



The ROS image that published to the system is stored in topic /mako_1/mako_1/image_raw. To do image processing, you need to write a ROS subscriber that subscribes this topic and process the image in OpenCV. Note OpenCV3 is installed by default when ROS was installed.

To see all the ROS topics:

$ **rostopic list**

To see the message type of a ROS topic:

$ **rostopic type /ros_topic_name**

For example, the message type of /mako_1/mako_1/image_raw

$ **rostopic type /mako_1/mako_1/image_raw**

sensor_msgs/Image

Reference: http://wiki.ros.org/cv_bridge/Tutorials

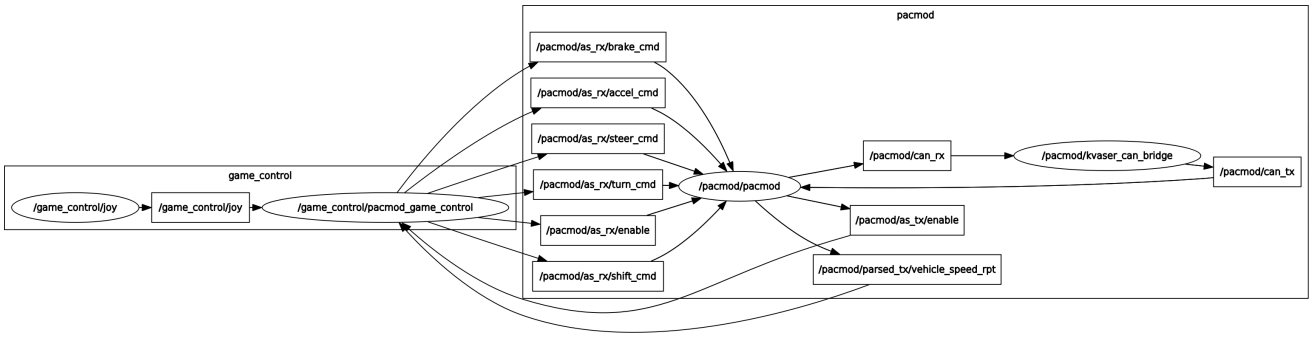
Driving Message:

To launch the control system of the vehicle. Double click the icon “DBW Joystick Demo” on the desktop. Alternatively, you can also use command:

$ **roslaunch basic_launch dbw_joystick.launch**

To see structure of the ROS system:

$ **rosrun rqt_graph rqt_graph**



To control the vehicle, you can publish data to ROS topics listed as below:

/pacmod/as_rx/accel_cmd
/pacmod/as_rx/brake_cmd
/pacmod/as_rx/enable
/pacmod/as_rx/shift_cmd
/pacmod/as_rx/steer_cmd
/pacmod/as_rx/turn_cmd

You don’t have to use all of them to control the vehicle. To make your control command valid to the system, you need to enable the control of steering, braking, etc. In other word, send true to topic /pacmod/as_rx/enable.

To control the vehicle, for example, steering, you need to know the message type of /pacmod/as_rx/steer_cmd

$ **rostopic type /pacmod/as_rx/steer_cmd**
pacmod_msgs/PositionWithSpeed

To get the definition of message pacmod_msgs/PositionWithSpeed

$ **rosmsg show pacmod_msgs/PositionWithSpeed**
std_msgs/Header header
float64 angular_position # The desired rotational position of the motor shaft about it's z axis in Radians
float64 angular_velocity_limit # The desired speed limit to acheive the desired position in Radians/second (z axis)

In order to control the steering, you just need to send values to angular_position and angular_velocity_limit individually.

Reference: http://wiki.ros.org/pacmod