Polaris GEM e2 & Simulator

The Center of Autonomy at University of Illinois at Urbana-Champaign

User Manual

Version 1.0

Create: 07/01/2020

Last Update: 08/19/2021

Contact: Hang Cui

Email: hangcui3@illinois.edu

Table of Contents

1. Polaris GEM e2 Hardware

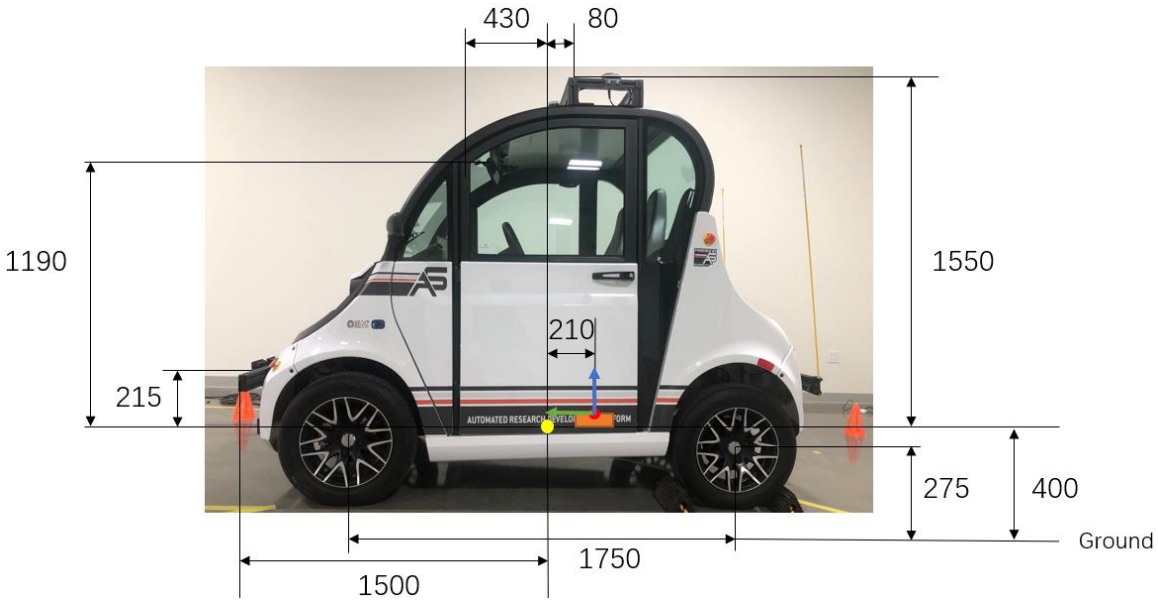
1.1 Polaris GEM e2 Vehicle

Software interfaces to the controls: steering, braking, acceleration

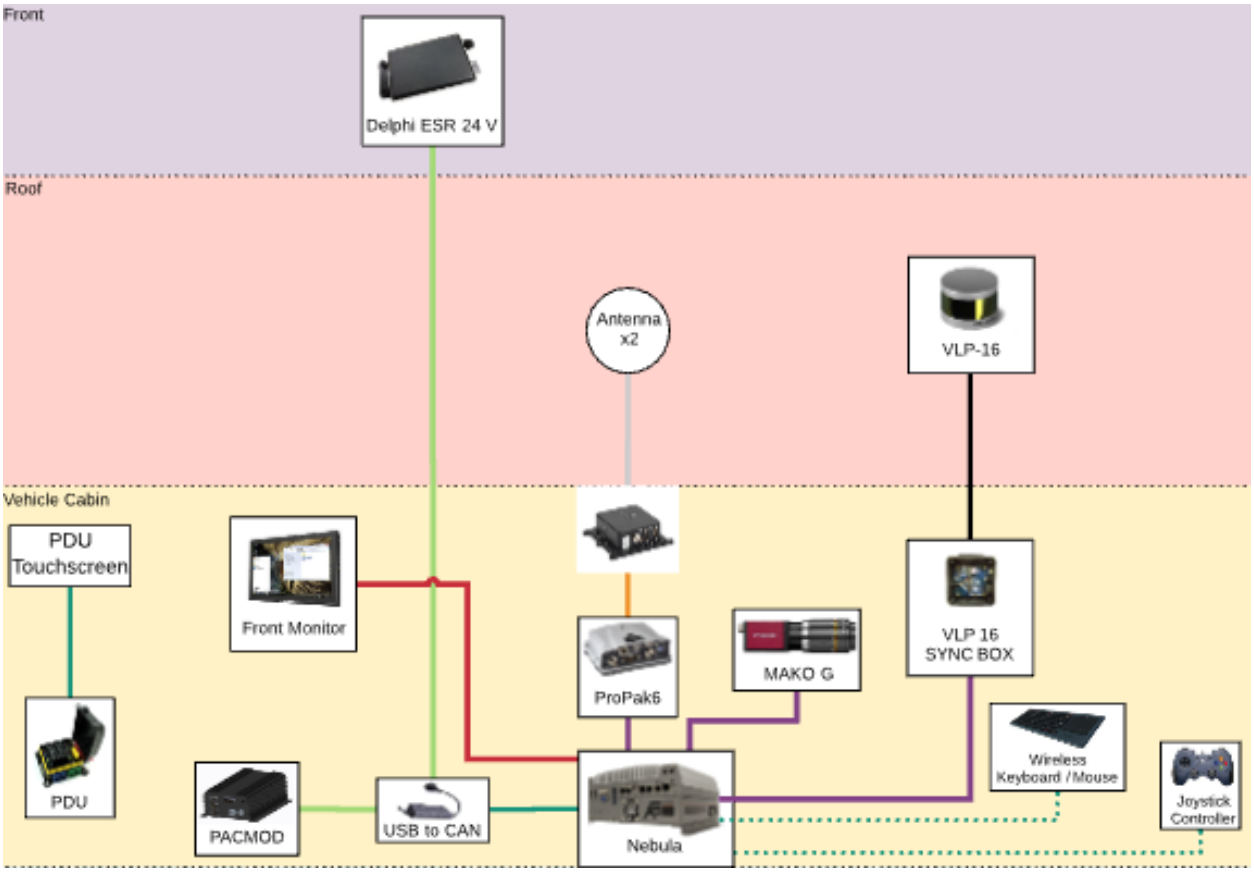
Software access: left and right blinkers, reverse and drive gear selection, speed feedback

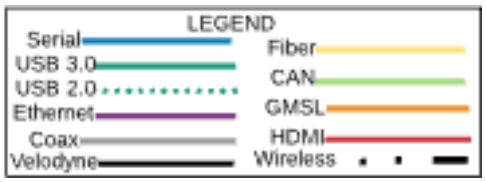
Convenience features: Dash mounted display screen, Power distribution terminals

Dimensions: mm



1.2 Hardware Overview





1.3 Master Power Switch

Switch will allow operator to cut power to power distribution system

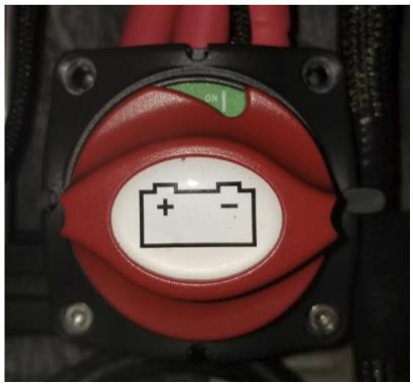
ON will supply power to power distribution system from vehicle battery

OFF will remove power to the power distribution system

Located under the driver's seat

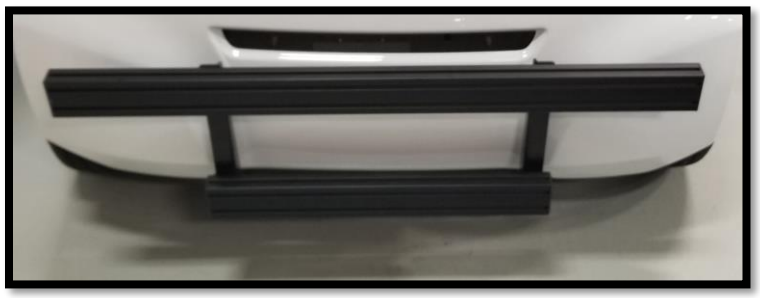
OFF

ON



1.4 Automated Research Development Platform

All front and rear racks are made with 3 inch x 1.5 inch 15 series 80/20

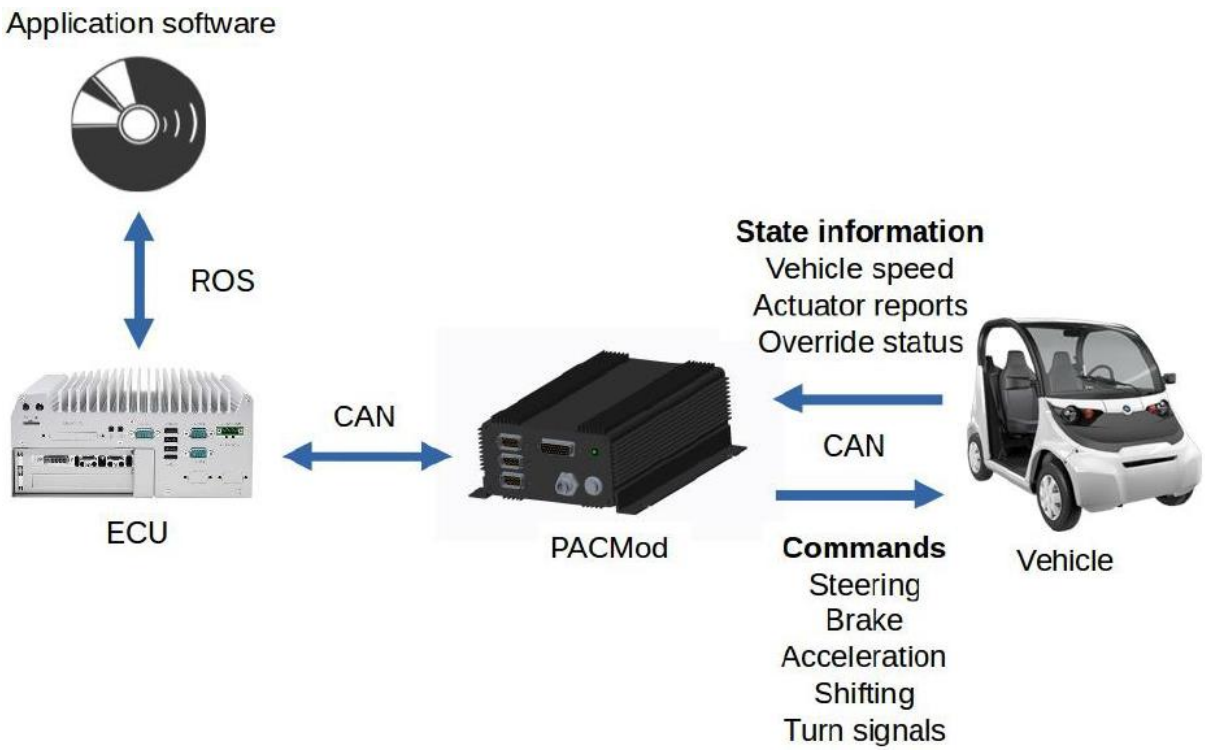


1.5 AStuff Spectra 2

The AutonomouStuff Spectra 2 is the world's first dual GPU edge ai platform with industrial-grade design and in-vehicle features. Designed specifically to support two high-end 250W NVIDIA® graphics cards, it offers tremendous GPU power up to 28 TFLOPS in FP32 for emerging GPU-accelerated edge computing, such as autonomous driving, vision inspection and surveillance/security.

Datasheet link: https://autonomoustuff.com/-/media/Images/Hexagon/Hexagon%20Core/autonomousstuff/pdf/as-spectra-2-datasheet.ashx?la=en&hash=3FBD8D8C48469BBC65773BA4752AACAD

1.6 PACMod Vehicle Interface



PACMod Override

Steering

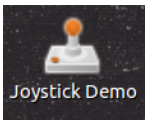
Brake / Throttle

1.7 Joystick Controller

Launching the Demo

There are two methods to launch the demonstration on a typical vehicle. First, an ECU configured by AutonomouStuff for use on a PACMod enabled vehicle will have a desktop icon named "Joystick Demo" for launching the demo graphically. Second, the launch through the terminal.

(1) On the Ubuntu Desktop, locate the icon shaped like a joystick with the name "Joystick Demo"



Set **LED OFF** using Mode button

Set **X mode** on the back of the controller

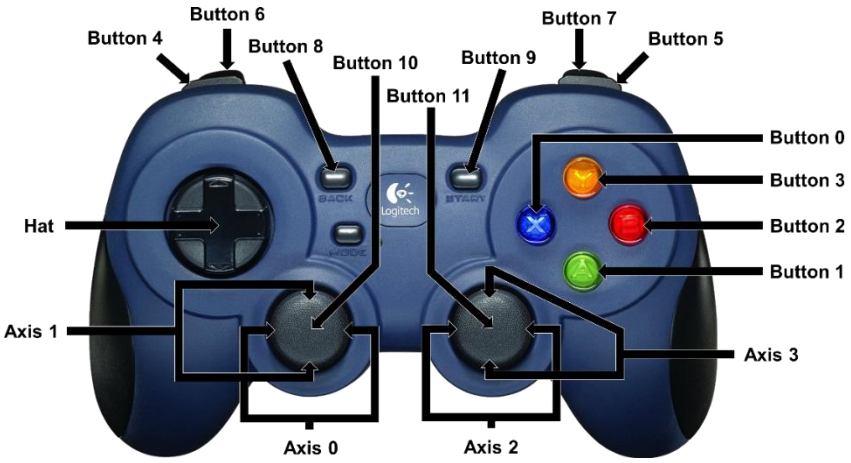
(2) Double click the icon and the demonstration will start

```
NODES
  /game_control/
    joy (joy/joy_node)
    pacmod_game_control (pacmod_game_control/pacmod_game_control_node)
  /
    kvaser_can_bridge (kvaser_interface/kvaser_can_bridge)
    pacmod (pacmod/pacmod)

auto-starting new master
process[master]: started with pid [22772]
ROS_MASTER_URI=http://joe-Oryx-Pro:11311/

setting /run_id to 9c8d62c0-dad5-11e7-b304-80fa5b395f4a
process[rosout-1]: started with pid [22785]
started core service [/rosout]
process[kvaser_can_bridge-2]: started with pid [22788]
process[pacmod-3]: started with pid [22791]
process[game_control/joy-4]: started with pid [22815]
process[game_control/pacmod_game_control-5]: started with pid [22835]
```

(3) At this point the demonstration has started and you can control the vehicle with the game controller



Axis 0	Left Joystick X
Axis 1	Left Joystick Y (inv)
Axis 2	Right Joystick X
Axis 3	Right Joystick Y (inv)
Button 0	X
Button 1	A
Button 2	B
Button 3	Y
Button 4	Left Bumper
Button 5	Right Bumper
Button 6	Left Trigger
Button 7	Right Trigger
Button 8	Back
Button 9	Start
Button 10	Left Joystick (click)
Button 11	Right Joystick (click)
Hat	Directional Pad (X, Y)

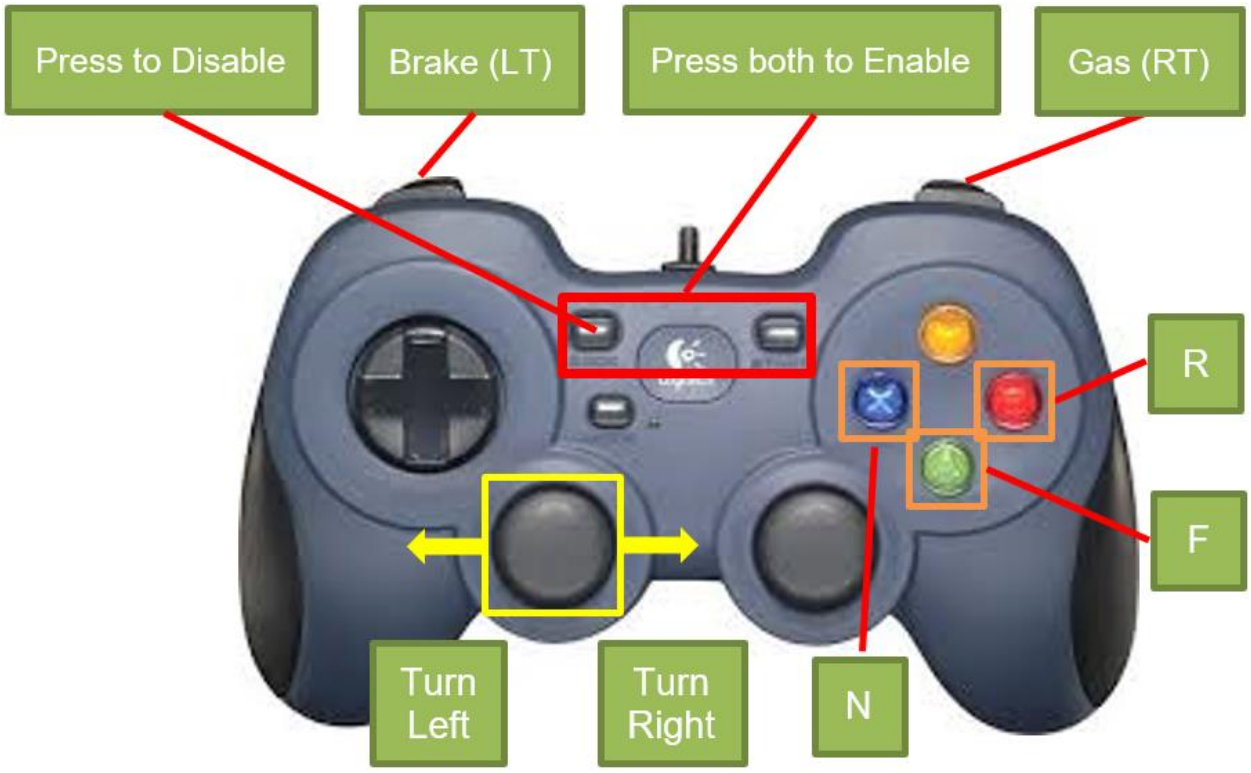
(4) **Joystick Demo**

basic_launch/launch/dbw_joystick.launch

$ **roslaunch basic_launch dbw_joystick.launch**

```
<launch>
  <include file="$(find pacmod_game_control)/launch/pacmod_game_control.launch">
    <arg name="launch_pacmod" value="false" />
    <arg name="is_pacmod_3" value="false" />
    <arg name="pacmod_vehicle_type" value="POLARIS_GEM" />
  </include>

  <include file="$(find platform_launch)/launch/$(env platform_name)/platform.launch">
    <arg name="use_dbw" value="true" />
  </include>
</launch>
```



1.8 Mako G-319C Camera

Mako G

G-319

- Sony IMX265 sensor
- Power over Ethernet
- Ultra-compact design
- Affordable

Mako G	G-319
Interface	IEEE 802.3 1000BASE-T, IEEE 802.3af (PoE)
Resolution	2064 (H) × 1544 (V)
Sensor	Sony IMX265
Sensor type	CMOS
Pixel size	3.45 µm x 3.45 µm

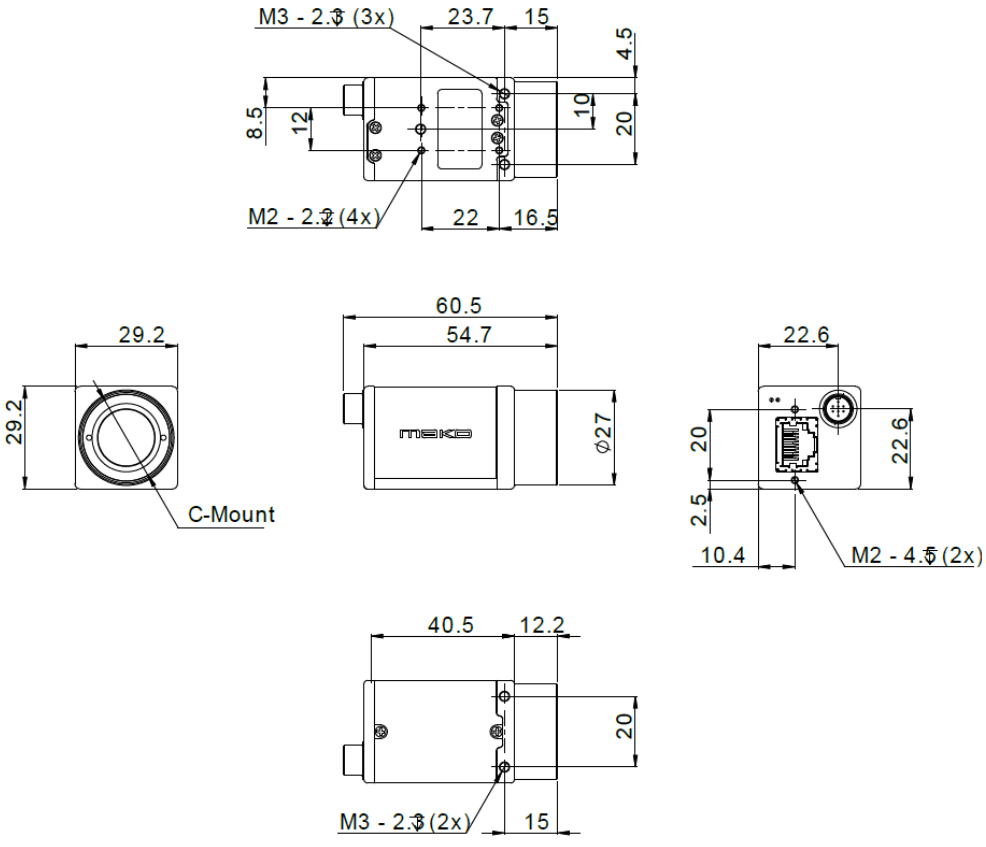
Mako G	G-319
Lens mount (default)	C-Mount
Max. frame rate at full resolution	37.5 fps
ADC	12 bit
Image buffer (RAM)	64
Output	
Bit depth	8/12 bit
Monochrome pixel formats	Mono8, Mono12, Mono12Packed
YUV color pixel formats	YUV411Packed, YUV422Packed, YUV444Packed
RGB color pixel formats	RGB8Packed, BGR8Packed
Raw pixel formats	BayerRG8, BayerRG12, BayerRG12Packed
General purpose inputs/outputs (GPIOs)	
Opto-isolated I/Os	1 input, 3 outputs
Operating conditions/dimensions	
Operating temperature	+5 °C to +45 °C housing temperature
Power requirements (DC)	12 to 24 VDC; PoE
Power consumption	2.3 W @ 12 VDC; 2.6 W PoE
Mass	80 g
Body dimensions (L × W × H in mm)	60.5 × 29.2 × 29.2 (including connectors)
Regulations	CE: 2014/30/EU (EMC), 2011/65/EU (RoHS); FCC Class B; CAN ICES-003

Image optimization features:

- Auto gain (manual gain control: 0 to 40 dB; 0.1 dB increments)
- Auto exposure (exposure time control varies by pixel format)
- Auto white balance (G-319C only)
- Binning
- Color correction, hue, saturation (G-319C only)
- Decimation
- Gamma correction
- One look-up table (LUT)
- Region of interest (ROI), separate ROI for auto features

Camera control features:

- Event channel
- Image chunk data
- Global shutter mode
- Storable user sets
- StreamBytesPerSecond (bandwidth control)
- Stream hold
- Sync out modes: Trigger ready, input, exposing, readout, imaging, strobe, GPO
- Temperature monitoring (main board only)



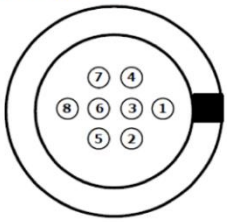
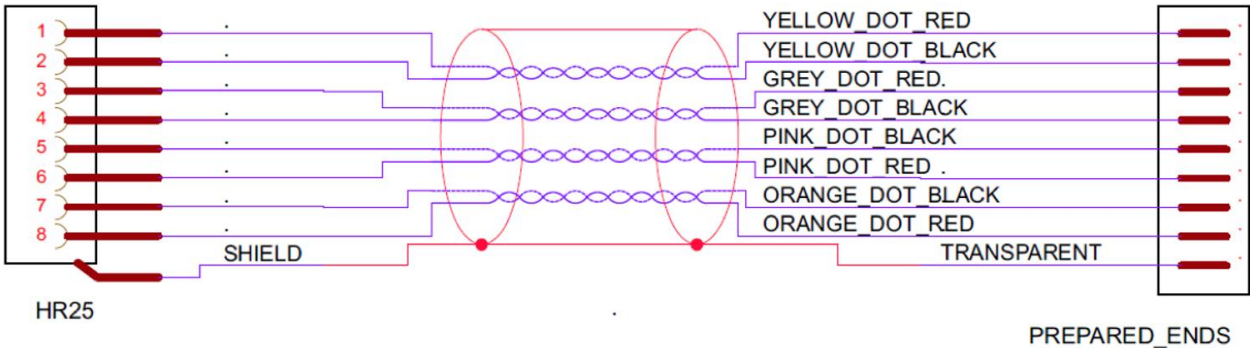
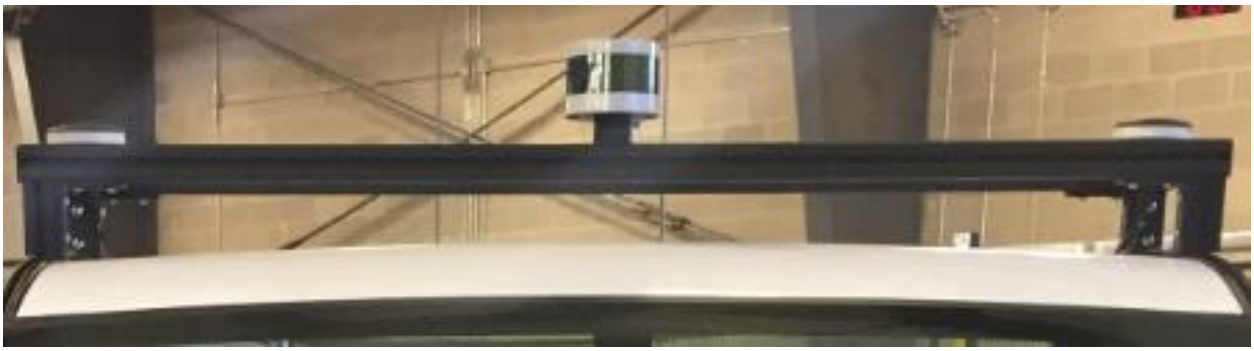
Drawing	Pin	Cable color	Signal	Direction	Level	Description
HR25A-7TP-8S 	1	Yellow dot Red	CameraOut1	Out	Open emitter max. 20 mA	Camera Output 1 (SyncOut1) opto-isolated
	2	Yellow dot Black	CameraOut2	Out	Open emitter max. 20 mA	Camera Output 2 (SyncOut2) opto-isolated
	3	Grey dot Red	CameraOut3	Out	Open emitter max. 20 mA	Camera Output 3 (SyncOut3) opto-isolated
	4	Grey dot Black	CameraIn	In	Uin(high) = 3 V...24 V Uin(low) = 0 V...1.0 V	Camera Input (SyncIn) opto-isolated
	5	Pink dot Black	CameraIn GND	In	Common GND for inputs	Camera Common Input Ground (In GND)
	6	Pink dot Red	CameraOut Power	In	Common VCC for outputs max. 30 V DC	Camera Output Power for digital outputs (OutVCC)
	7	Orange dot Black	ExtPower	---	12 V DC... 24 V DC +/- 10 %	Power Supply
	8	Orange dot Red	GND	---	GND for ext. Power	External Ground for external power

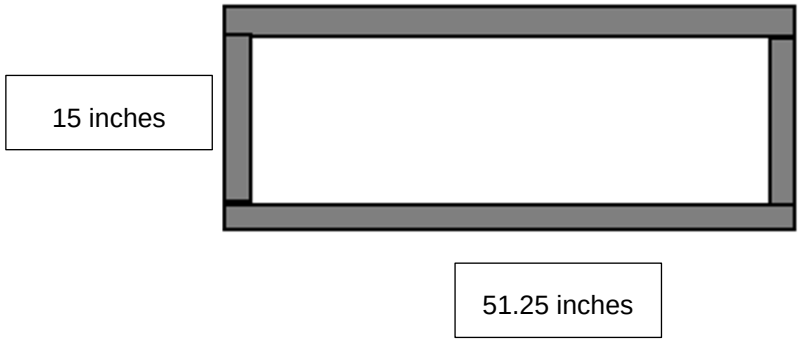
Table 5: Mako-GI/O definition



1.9 Velodyne VLP-16 LiDAR

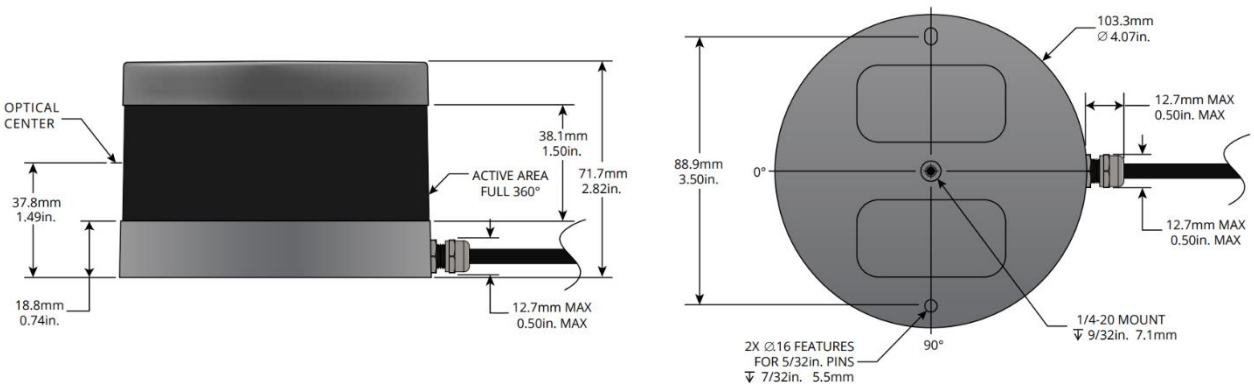
Roof Rack



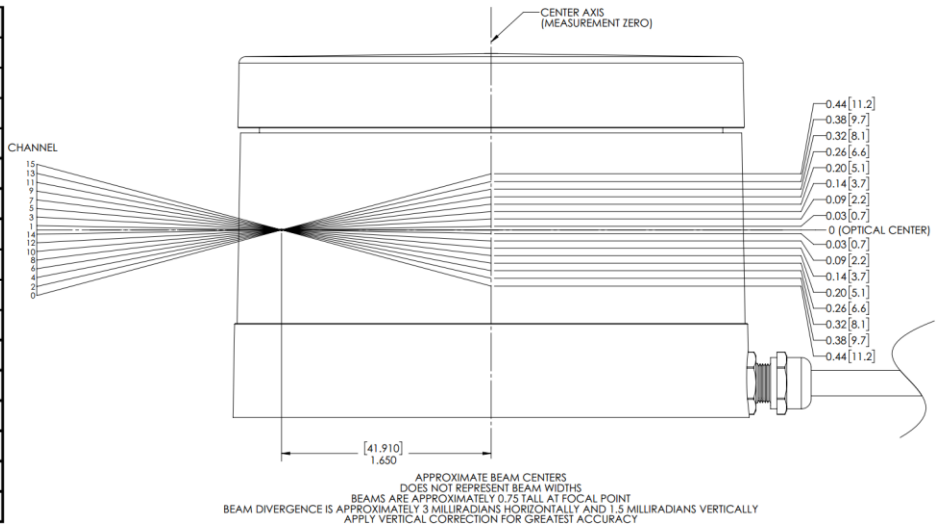


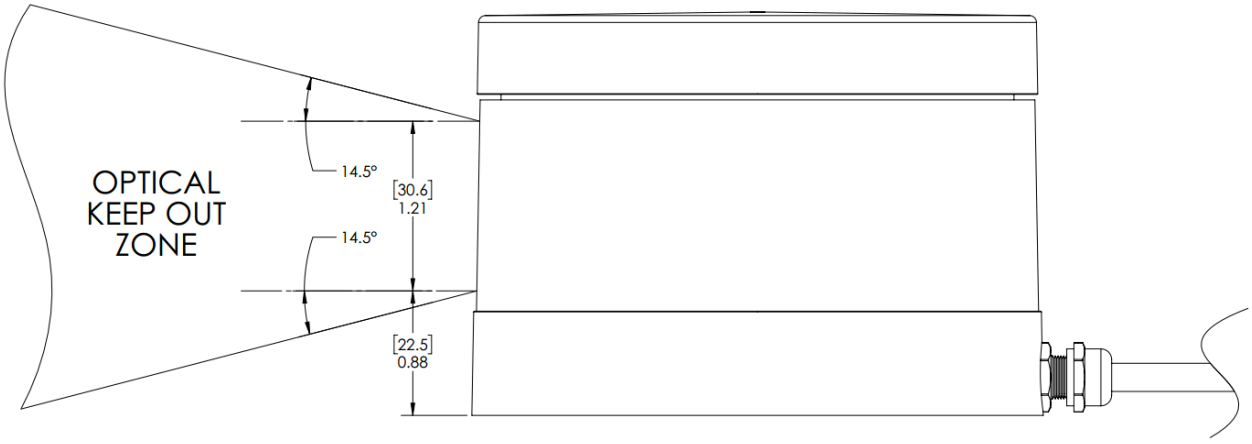
All roof racks are made with 1.5 by 1.5 inches 15 series 80/20

Dimensions



Laser ID	Vertical Angle
0	-15°
1	1°
2	-13°
3	-3°
4	-11°
5	5°
6	-9°
7	7°
8	-7°
9	9°
10	-5°
11	11°
12	-3°
13	13°
14	-1°
15	15°





Sensor

- 16 Channels
- Measurement Range: 100 m – 120 m
- Range Accuracy: Up to ±3 cm (Typical)1
- Field of View (Vertical): +15.0° to -15.0° (30°)
- Angular Resolution (Vertical): 2.0°
- Field of View (Horizontal): 360°
- Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4°
- Rotation Rate: 5 Hz – 20 Hz
- Integrated Web Server for Easy Monitoring and Configuration

Mechanical / Electrical / Operational

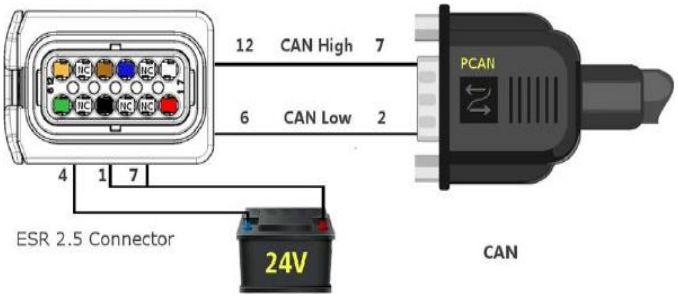
- Power Consumption: 8 W (Typical)2
- Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply)
- Weight: ~590 g (without Cabling and Interface Box)
- Dimensions: See diagram on previous page
- Environmental Protection: IP67
- Operating Temperature: -10°C to +60°C3
- Storage Temperature: -40°C to +105°C

Outputs

- 3D LiDAR Data Points Generated:
  - Single Return Mode: ~300,000 points per second
  - Dual Return Mode: ~600,000 points per second
- 100 Mbps Ethernet Connection
- UDP Packets Contain:
  - Time of Flight Distance Measurement
  - Calibrated Reflectivity Measurement
  - Rotation Angles
  - Synchronized Time Stamps (µs resolution)
- GPS: $GPRMC and $GPGGA NMEA Sentences from GPS Receiver (GPS not included)

1.10 Delphi ESR 2.5 Radar (24V)

CAN / USB Connection Wiring



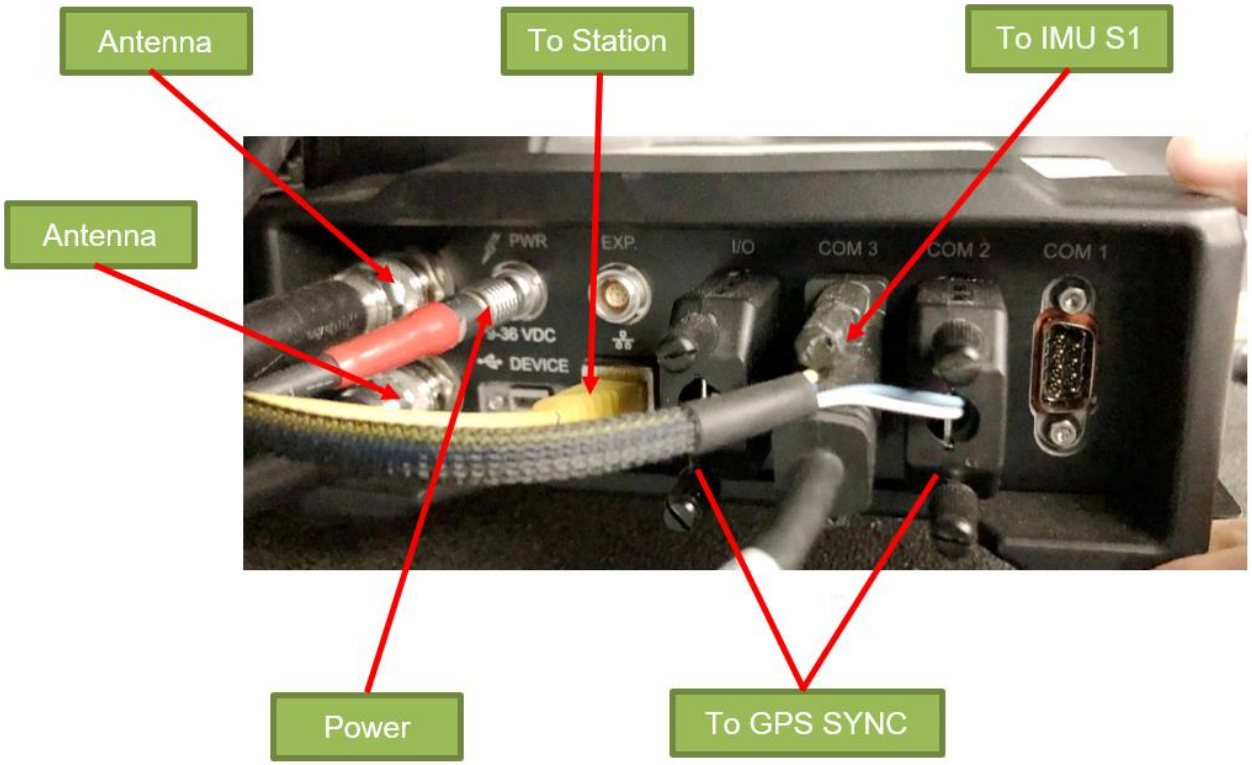
Pin #	Signal	Color
1	Battery (+24V)	Red
2	USB D+ (green wire)	Green (USB)
3	USB D- (white wire)	White (USB)
4	Ground	Black
5	USB Ground (black wire)	Black (USB)
6	PRVCANL	Green
7	Ignition (+24V)	White
8	USB +5V (red wire)	Red (USB)
9	VEHCANL	Blue
10	VEHCANH	Brown
11	VEHCAN Shield	
12	PRVCANH	Orange

USB-to-CAN (Kvaser Hybrid 2xCAN/LIN)

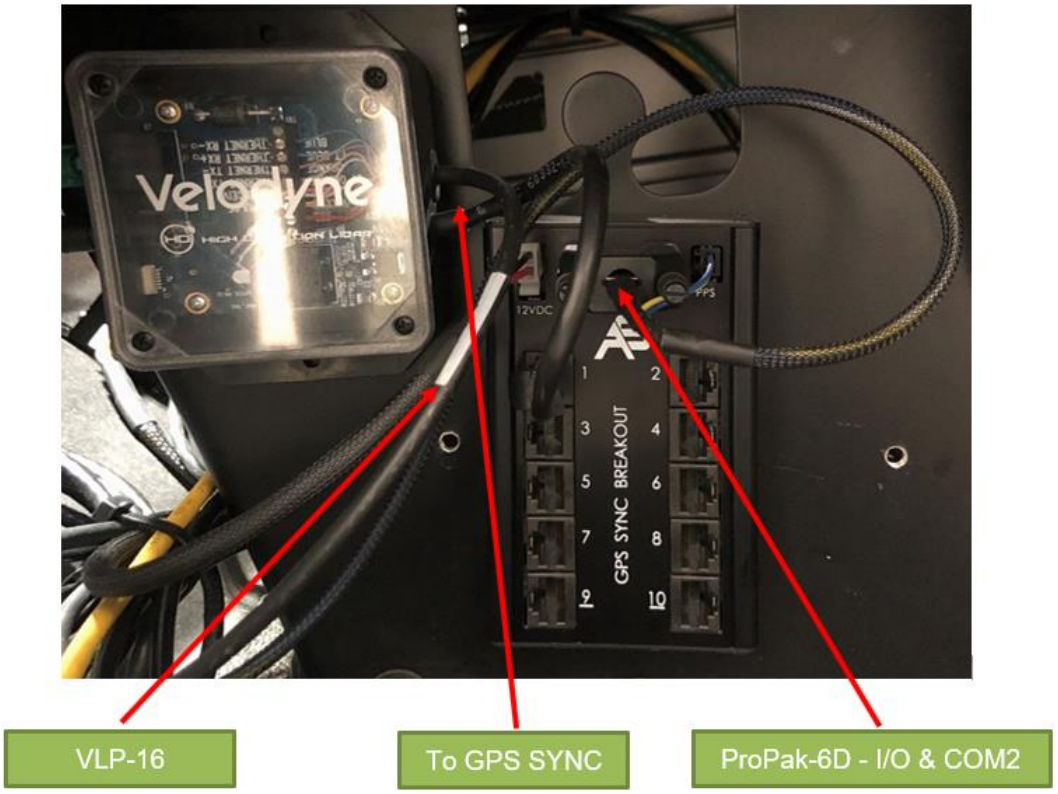
1.11 ProPak 6 & SPAN-IGM-S1

ProPak-6D1

Dual Antenna Support
Cellular
L1/L2 GPS+GLONASS
L-Band TerraStar-C PPP Corrections
-3 Grade IMUs
20 Hz Positions and Measurements
4GB Internal Memory



Connector Type	Connector Label	Description
GNSS Antenna External Oscillator	ANT 1 ANT 2 or ANT1 OSC	GNSS GPS1 and GPS2 antennas (TNC) (model dependant) or GNSS GPS1 antenna (TNC) and external oscillator (BNC) (model dependant)
Power	PWR	4-pin LEMO power connector
Expansion	EXP.	9-pin LEMO expansion port for CAN1 and CAN2
USB	DEVICE	USB Device (Type micro B) connector (high speed only) 480 Mbps
Ethernet		Ethernet RJ45 connector
I/O	I/O	4 Event Input/3 Event Output (DB9 female connector) I/O port is configurable
Serial Communication Ports	COM1 COM2 COM3/IMU	COM1, COM2, COM3/IMU DB9 male communications port RS-232 (RS-422 selectable via software)



SPAN-IGM-S1

200Hz/125 Hz Inertial Measurements
Direct Wheel Sensor Support
Commercially Exportable
Small and lightweight design

G5Ant-3AMT4

Matte black finish without branding
Various mounting options and connectors
Size: 89 mm dia. x 25 mm hgt
Weight: 368 g

2. Polaris GEM e2 ROS Software

2.1 Software Setup

Setup .bashrc

source /opt/ros/noetic/setup.bash

Setup AutonomouStuff drivers

$ sudo apt update && sudo apt install apt-transport-https

$ sudo sh -c 'echo "deb [trusted=yes] https://s3.amazonaws.com/autonomoustuff-repo/ $(lsb_release -sc) main" > /etc/apt/sources.list.d/autonomoustuff-public.list'

Install Kvaser linuxcan SDK:

https://autonomoustuff.atlassian.net/wiki/spaces/RW/pages/17475947/Driver+Pack+Installation+or+Upgrade+Instructions

https://www.kvaser.com/download/

$ sudo apt install ros-$ROS_DISTRO-kvaser-interface ros-$ROS_DISTRO-delphi-esr ros-$ROS_DISTRO-astuff-sensor-msgs ros-$ROS_DISTRO-pacmod ros-$ROS_DISTRO-pacmod-game-control

Extra Software

$ sudo apt install solaar

$ sudo apt install preload

$ sudo apt install meld

$ sudo apt-get install indicator-multiload

2.2 Frame Setup

platform_launch/launch/white_e2/platform.launch

platform_launch/launch/core/all_supported_drivers.launch

veh_frame (default=base_link)

front_radar_frame (default=front_radar)

lidar1_frame (default=lidar1)

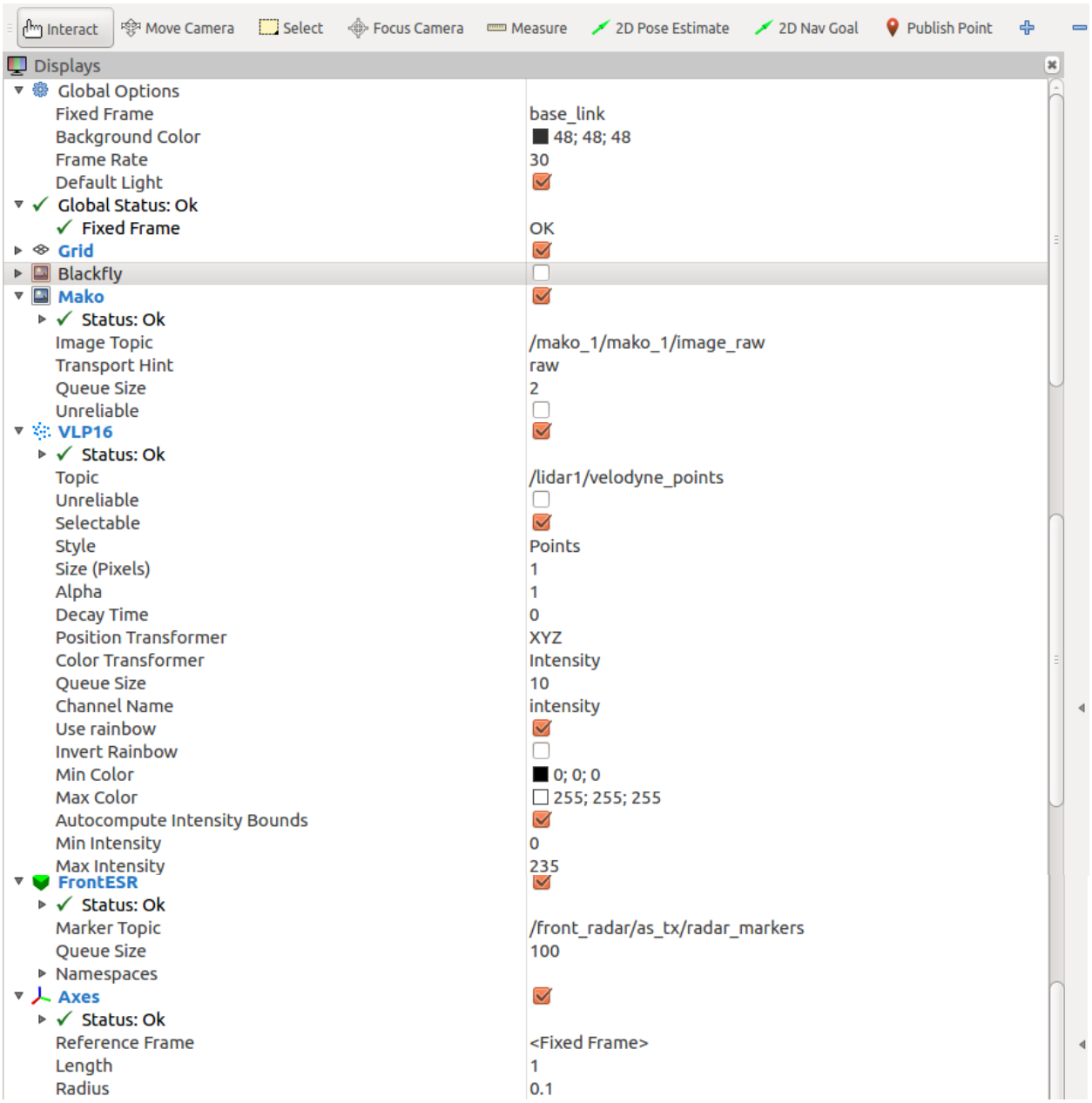
novatel_frame (default=novatel)

novatel_imu_frame (default=imu)

mako_1_frame (default="")

Usage: static_transform_publisher x y z yaw pitch roll frame_id child_frame_id period (ms)

2.3 AStuff Spectra 2 Station (Rviz)



2.4 PACMod Software Vehicle Interface

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

Source: https://github.com/astuff/pacmod.git (branch: release)

Supported Hardware

- Polaris GEM Series (e2/e4/e6/eLXD)
- Polaris Ranger X900
- International Prostar+ 122
- Lexus RX-450h

can_msgs/Frame.msg

Header header
uint32 id
bool is_rtr
bool is_extended
bool is_error
uint8 dlc
uint8[8] data

CAN Device List

```
dev@dev-gem:/usr/src/linuxcan/canlib/examples$ ./listChannels
CANlib version 5.28
Found 2 channel(s).
ch  0: Kvaser USBcan Light 2xHS 73-30130-00714-7, s/n 11783, v4.1.844  (leaf v8.28.846)
ch  1: Kvaser USBcan Light 2xHS 73-30130-00714-7, s/n 11783, v4.1.844  (leaf v8.28.846)
dev@dev-gem:/usr/src/linuxcan/canlib/examples$
```

Published Topics

Topic	Message Type	Description
can_rx	can_msgs/Frame	All data published on this topic is intended to be sent to the PACMod system via a CAN interface.
parsed_tx/global_rpt	pacmod_msgs/GlobalRpt	High-level data about the entire PACMod system.
parsed_tx/accel_rpt	pacmod_msgs/SystemRptFloat	Status and parsed values [pct] of the throttle subsystem.
parsed_tx/brake_rpt	pacmod_msgs/SystemRptFloat	Status and parsed values [pct] of the steering susbsystem.
parsed_tx/steer_rpt	pacmod_msgs/SystemRptFloat	Status and parsed values [rad] of the steering susbsystem.
parsed_tx/turn_rpt	pacmod_msgs/SystemRptInt	Status and parsed values [enum] of the turn signal subsystem.
parsed_tx/shift_rpt	pacmod_msgs/SystemRptInt	Status and parsed values [enum] of the gear/transmission subsystem.
parsed_tx/vehicle_speed_rpt	pacmod_msgs/VehicleSpeedRpt	The vehicle's current speed [mph], the validity of the speed message [bool], and the raw CAN message from the vehicle CAN.
parsed_tx/vin_rpt	pacmod_msgs/VinRpt	The configured vehicle's VIN, make, model, manufacturer, and model year.
as_tx/vehicle_speed	std_msgs/Float64	The vehicle's current speed [m/s].
as_tx/enable	std_msgs/Bool	The current status of the PACMod's control of the vehicle. If the PACMod is enabled, this value will be true. If it is disabled or overridden, this value will be false.

Subscribed Topics

Topic	Message Type	Description
can_tx	can_msgs/Frame	All data published to this topic will be parsed by the PACMod driver. This should be connected to a CAN interface.
as_rx/accel_cmd	pacmod_msgs/PacmodCmd	Commands the throttle subsystem to seek a specific pedal position [pct - 0.0 to 1.0].
as_rx/brake_cmd	pacmod_msgs/PacmodCmd	Commands the brake subsystem to seek a specific pedal position [pct - 0.0 to 1.0].
as_rx/shift_cmd	pacmod_msgs/PacmodCmd	Commands the gear/transmission subsystem to shift to a different gear [enum].
as_rx/turn_cmd	pacmod_msgs/PacmodCmd	Commands the turn signal subsystem to transition to a given state [enum].
as_rx/steer_cmd	pacmod_msgs/PositionWIthSpeed	Commands the steering subsystem to seek a specific steering wheel angle [rad] at a given rotation velocity [rad/s].
as_rx/enable	std_msgs/Bool	Enables [true] or disables [false] PACMod's control of the vehicle.

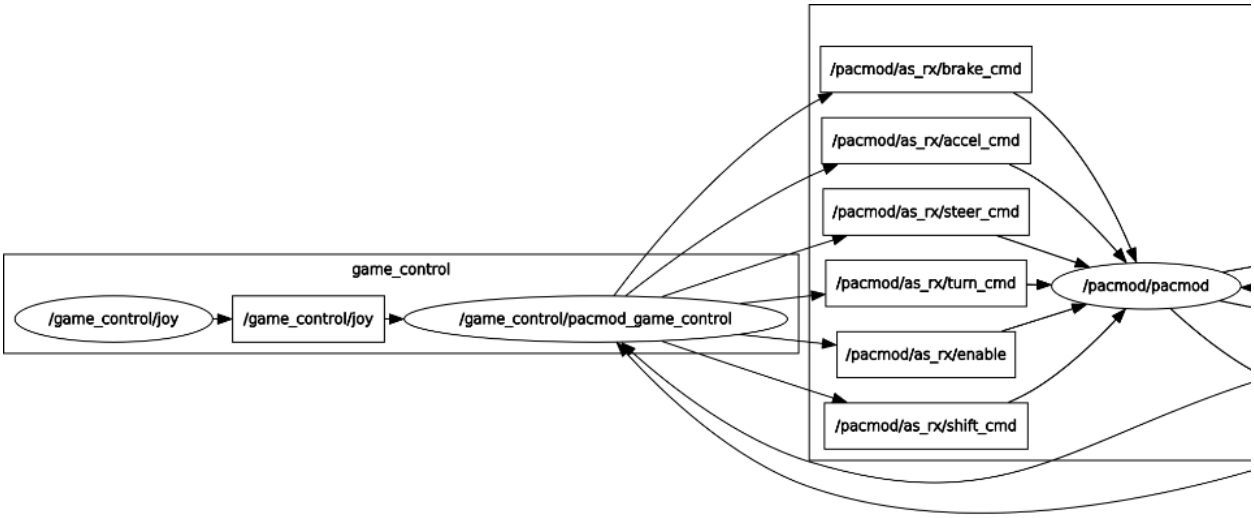
Parameters

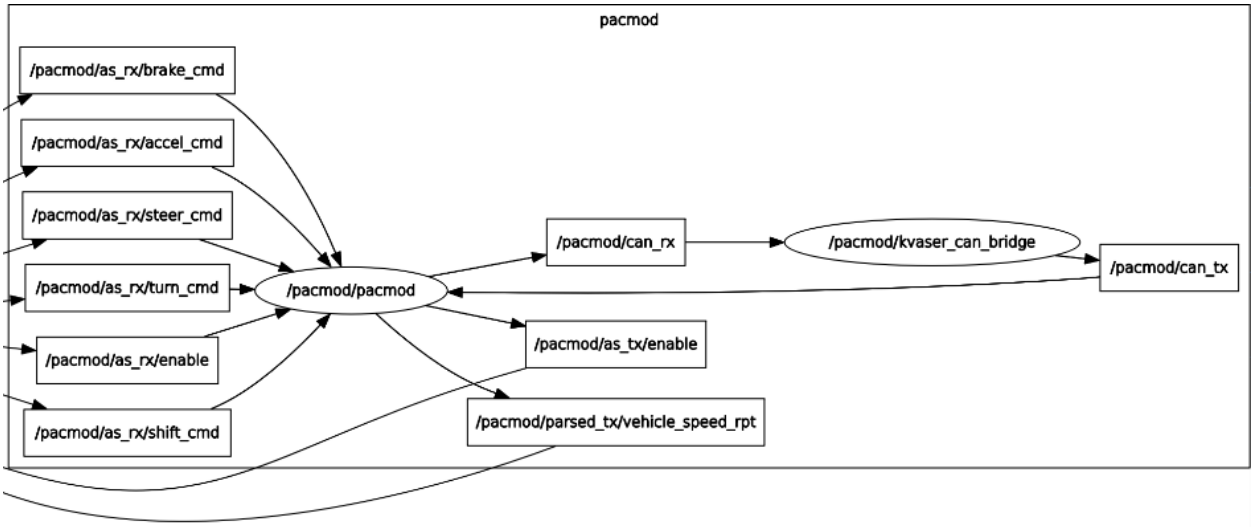
~vehicle_type: a string value indicating the type of vehicle to which the PACMod is connected.

Valid values are:

- POLARIS_GEM
- POLARIS_RANGER
- INTERNATIONAL_PROSTAR_122
- LEXUS_RX_450H

PACMod Graph





/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

2.5 Joystick Controller

ROS wiki: http://wiki.ros.org/pacmod_game_control

Source: https://github.com/astuff/pacmod_game_control.git (branch: release)

ROS wiki: http://wiki.ros.org/joy

Source: https://github.com/ros-drivers/joystick_drivers.git (branch: master)

Parameters

~steering_stick: sets whether the steering command should be controlled by the left or right joystick on a two-stick controller. Valid values are LEFT or RIGHT.

~pacmod_vehicle_type: sets the type of vehicle which is being controlled. This manages vehicle-specific values like the available features and maximum steering angle. Valid values are:

- POLARIS_GEM
- POLARIS_RANGER
- LEXUS_RX_450H
- INTERNATIONAL_PROSTAR_122
- VEHICLE_4
- VEHICLE_5
- VEHICLE_6

~controller_type: sets type of controller being used and associated button mappings. Valid values are:

- LOGITECH_F310
- HRI_SAFE_REMOTE
- LOGITECH_G29
- NINTENDO_SWITCH_WIRED_PLUS
- XBOX_ONE

~steering_max_speed: the maximum rotational speed for the steering wheel in rad/s.

~max_veh_speed: the vehicle speed is used to scale the rotation rate of the steering wheel. This value is the speed, in m/s, at which the most restriction is placed on rotation rate. This helps controllability as speed increases.

~accel_scale_val: a scaling value (0.0 - 1.0) for the accelerator. 1.0 = full throttle range. 0.0 = no throttle control.

~brake_scale_val: a scaling value (0.0 - 1.0) for the brake. 1.0 = full braking range. 0.0 = no brake control.

2.6 ROS Topics of Polaris GEM e2

To get the message definition:

$ **rostopic type </rostopic_name>**

Joystick:

/game_control/joy
/game_control/joy/set_feedback

Front RADAR:

/front_radar/as_rx/vehicle_motion
/front_radar/as_tx/objects
/front_radar/as_tx/radar_error_status
/front_radar/as_tx/radar_markers
/front_radar/as_tx/radar_markers_array
/front_radar/as_tx/radar_status
/front_radar/as_tx/radar_tracks
/front_radar/can_rx
/front_radar/can_tx
/front_radar/parsed_rx/vehicle1_msgs
/front_radar/parsed_rx/vehicle2_msgs
/front_radar/parsed_rx/vehicle3_msgs
/front_radar/parsed_rx/vehicle4_msgs
/front_radar/parsed_rx/vehicle5_msgs
/front_radar/parsed_tx/radarstatus1
/front_radar/parsed_tx/radarstatus2
/front_radar/parsed_tx/radarstatus3
/front_radar/parsed_tx/radarstatus4
/front_radar/parsed_tx/radarstatus5
/front_radar/parsed_tx/radarstatus6
/front_radar/parsed_tx/radarstatus7
/front_radar/parsed_tx/radarstatus8
/front_radar/parsed_tx/radarstatus9
/front_radar/parsed_tx/radartrack
/front_radar/parsed_tx/radarvalid1
/front_radar/parsed_tx/radarvalid2
/front_radar/parsed_tx/trackmotionpower

LiDAR:

/lidar1/lidar1_nodelet_manager/bond
/lidar1/lidar1_nodelet_manager_cloud/parameter_descriptions
/lidar1/lidar1_nodelet_manager_cloud/parameter_updates
/lidar1/lidar1_nodelet_manager_driver/parameter_descriptions
/lidar1/lidar1_nodelet_manager_driver/parameter_updates
/lidar1/lidar1_nodelet_manager_laserscan/parameter_descriptions
/lidar1/lidar1_nodelet_manager_laserscan/parameter_updates
/lidar1/scan
/lidar1/velodyne_packets
/lidar1/velodyne_points

Front Camera:

/mako_1/mako_1/camera_info
/mako_1/mako_1/image_raw
/mako_1/mako_1/image_raw/compressed
/mako_1/mako_1/image_raw/compressed/parameter_descriptions
/mako_1/mako_1/image_raw/compressed/parameter_updates
/mako_1/mako_1/image_raw/compressedDepth
/mako_1/mako_1/image_raw/compressedDepth/parameter_descriptions
/mako_1/mako_1/image_raw/compressedDepth/parameter_updates
/mako_1/mako_1/image_raw/theora
/mako_1/mako_1/image_raw/theora/parameter_descriptions
/mako_1/mako_1/image_raw/theora/parameter_updates
/mako_1/mako_1/parameter_descriptions
/mako_1/mako_1/parameter_updates

GNSS & INS:

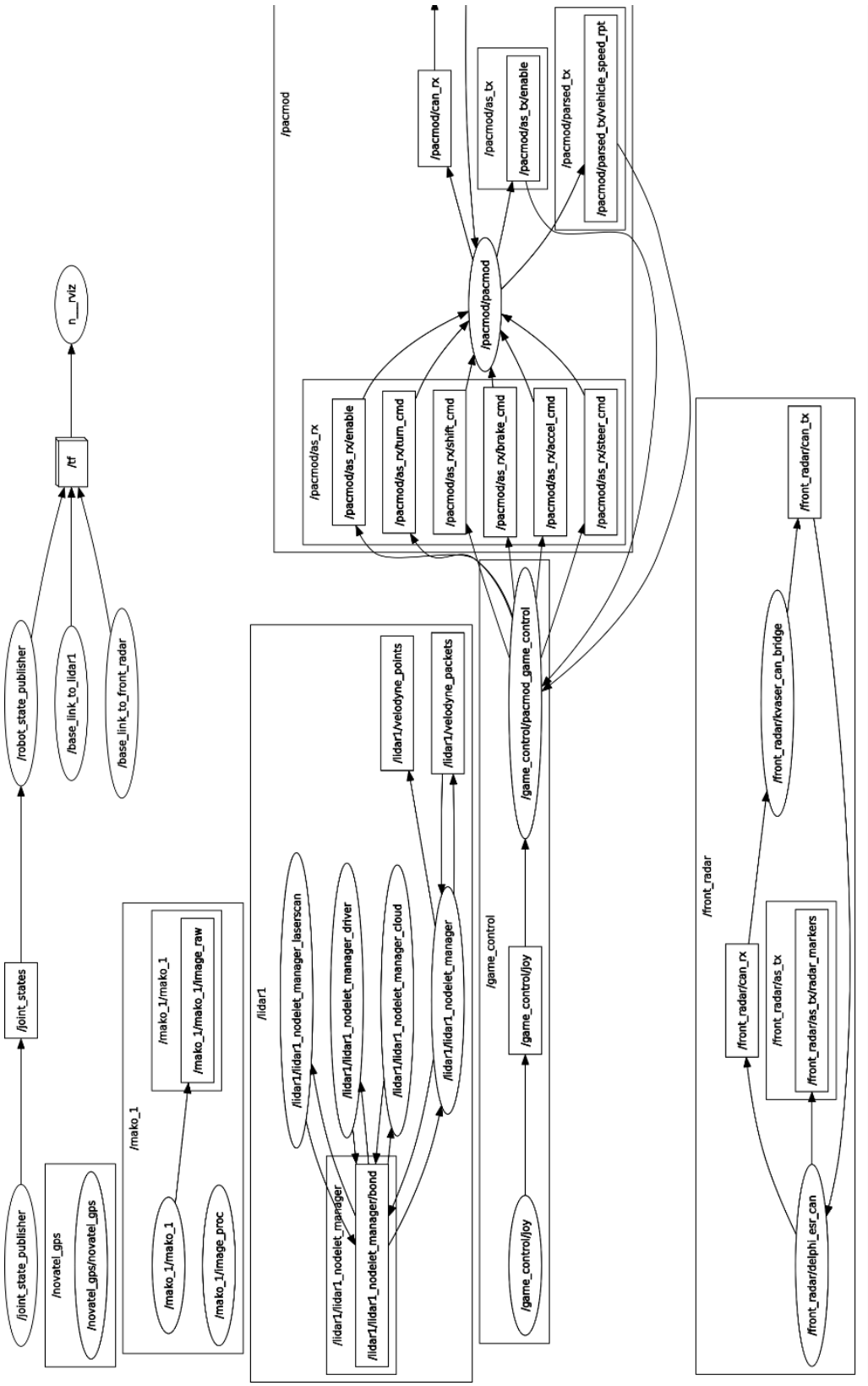
/novatel_gps/bestpos
/novatel_gps/corrimudata
/novatel_gps/fix
/novatel_gps/gpgga
/novatel_gps/gprmc
/novatel_gps/gps
/novatel_gps/gps_sync
/novatel_gps/imu
/novatel_gps/inscov
/novatel_gps/inspva

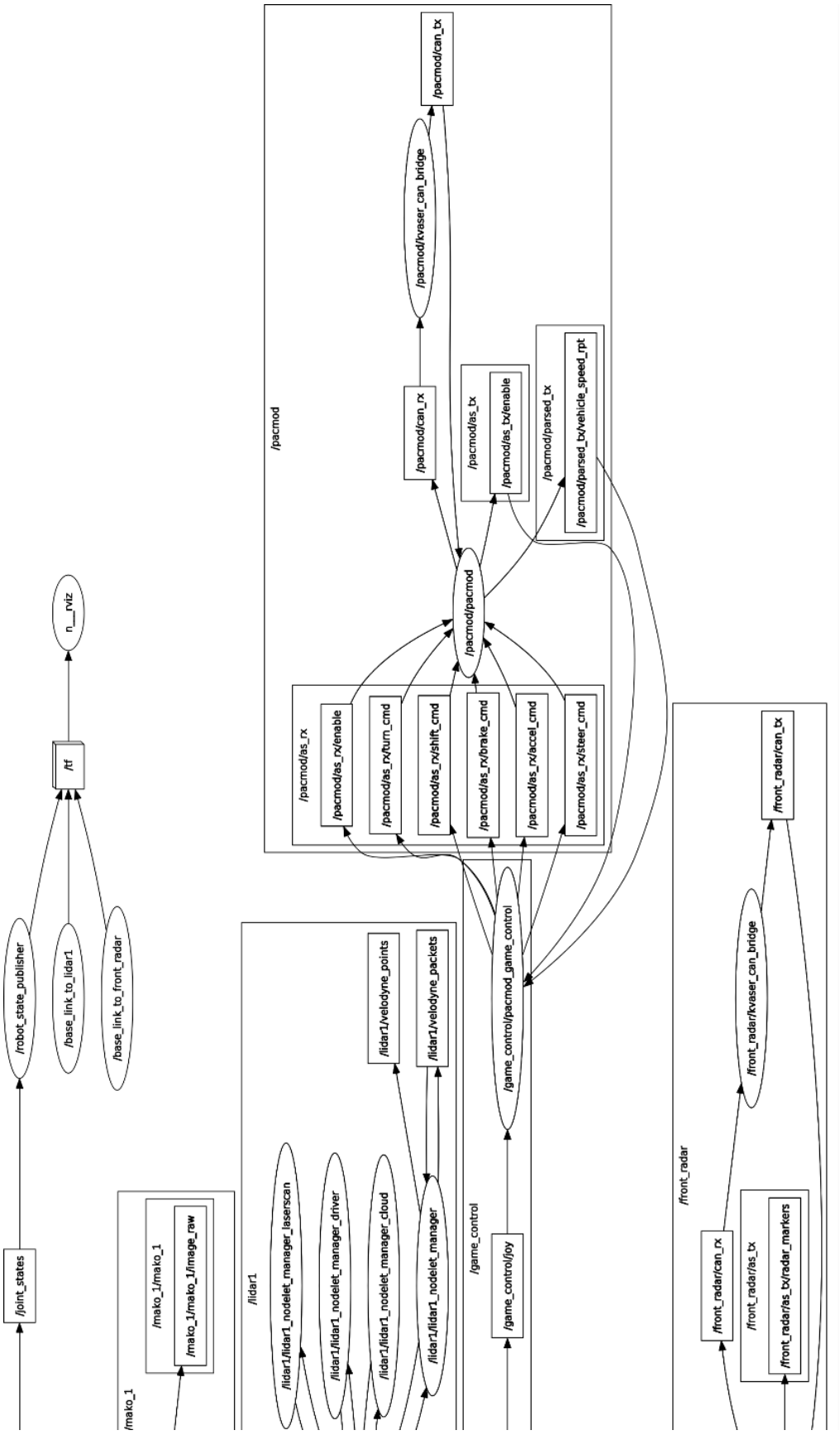
/novatel_gps/inspvax
/novatel_gps/insstdev

PACMOD:

/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/headlight_cmd
/pacmod/as_rx/horn_cmd
/pacmod/as_rx/turn_cmd
/pacmod/as_rx/wiper_cmd
/pacmod/as_tx/enable
/pacmod/as_tx/vehicle_speed
/pacmod/can_rx
/pacmod/can_tx
/pacmod/parsed_tx/accel_rpt
/pacmod/parsed_tx/brake_rpt
/pacmod/parsed_tx/brake_rpt_detail_1
/pacmod/parsed_tx/brake_rpt_detail_2
/pacmod/parsed_tx/brake_rpt_detail_3
/pacmod/parsed_tx/global_rpt
/pacmod/parsed_tx/shift_rpt
/pacmod/parsed_tx/steer_rpt
/pacmod/parsed_tx/steer_rpt_detail_1
/pacmod/parsed_tx/steer_rpt_detail_2
/pacmod/parsed_tx/steer_rpt_detail_3
/pacmod/parsed_tx/turn_rpt
/pacmod/parsed_tx/vehicle_speed_rpt
/pacmod/parsed_tx/vin_rpt

ROS rqt_graph




/joint_states
/robot_state_publisher
/base_link_to_lidar1
/base_link_to_front_radar
/tf
n__rviz
/mako_1
/mako_1/mako_1
/mako_1/mako_1/image_raw
/lidar1
/lidar1/lidar1_nodelet_manager_laserscan
/lidar1/lidar1_nodelet_manager_driver
/lidar1/lidar1_nodelet_manager_cloud
/lidar1/lidar1_nodelet_manager
/lidar1/velodyne_points
/lidar1/velodyne_packets
/game_control
/game_control/joy
/game_control/pacmod_game_control
/pacmod
/pacmod/as_rx
/pacmod/as_rx/enable
/pacmod/as_rx/turn_cmd
/pacmod/as_rx/shift_cmd
/pacmod/as_rx/brake_cmd
/pacmod/as_rx/accel_cmd
/pacmod/as_rx/steer_cmd
/pacmod/pacmod
/pacmod/can_rx
/pacmod/kvaser_can_bridge
/pacmod/can_tx
/pacmod/as_tx
/pacmod/as_tx/enable
/pacmod/parsed_tx
/pacmod/parsed_tx/vehicle_speed_rpt
/front_radar
/front_radar/can_rx
/front_radar/kvaser_can_bridge
/front_radar/can_tx
/front_radar/as_tx
/front_radar/as_tx/radar_markers

2.7 Coming more

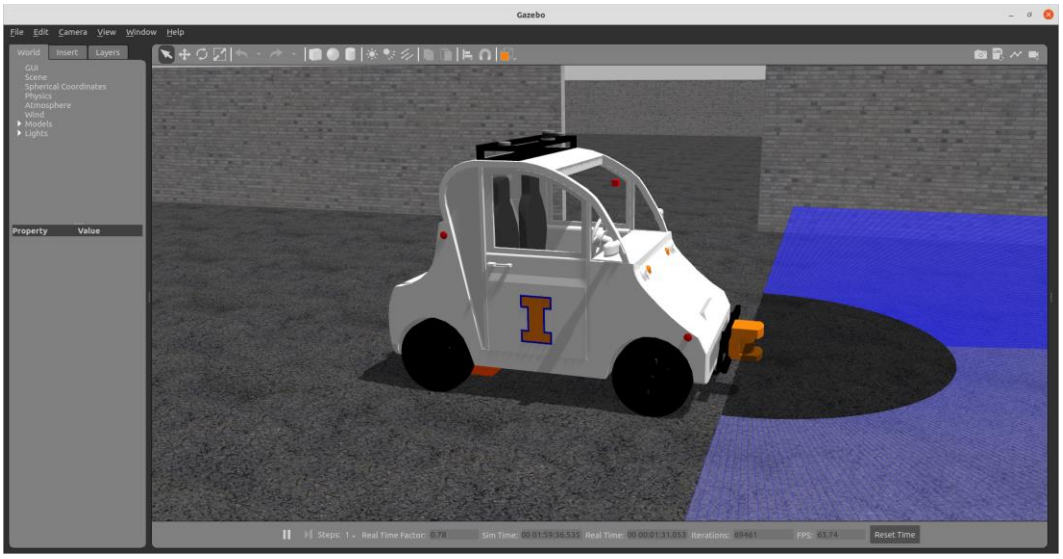
3. Polaris GEM e2 ROS Simulator (Ubuntu 20.04 + ROS Noetic)

3.1 Introduction

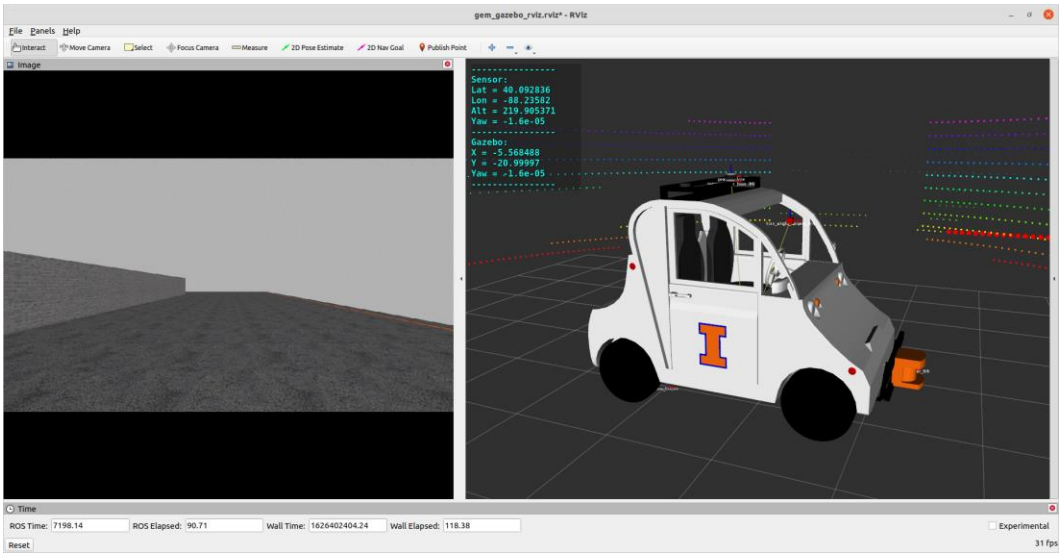
GitHub link: https://github.com/hangcui1201/POLARIS_GEM_e2

The Polaris GEM e2 simulator has been improved and merged into ROS Noetic and Gazebo 11

GEM vehicle with frontal 2D LiDAR



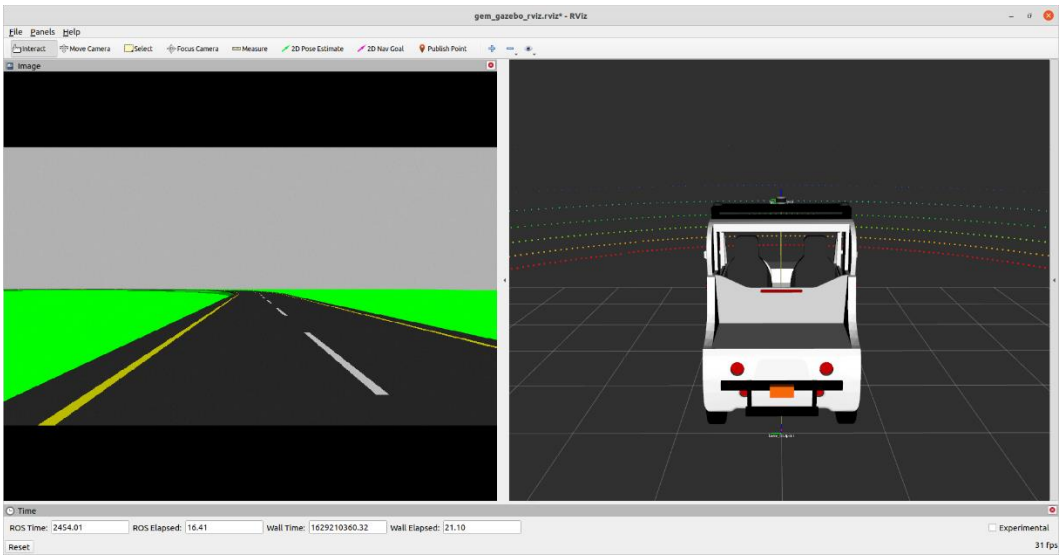
GEM vehicle with top 3D LiDAR and frontal 2D LiDAR

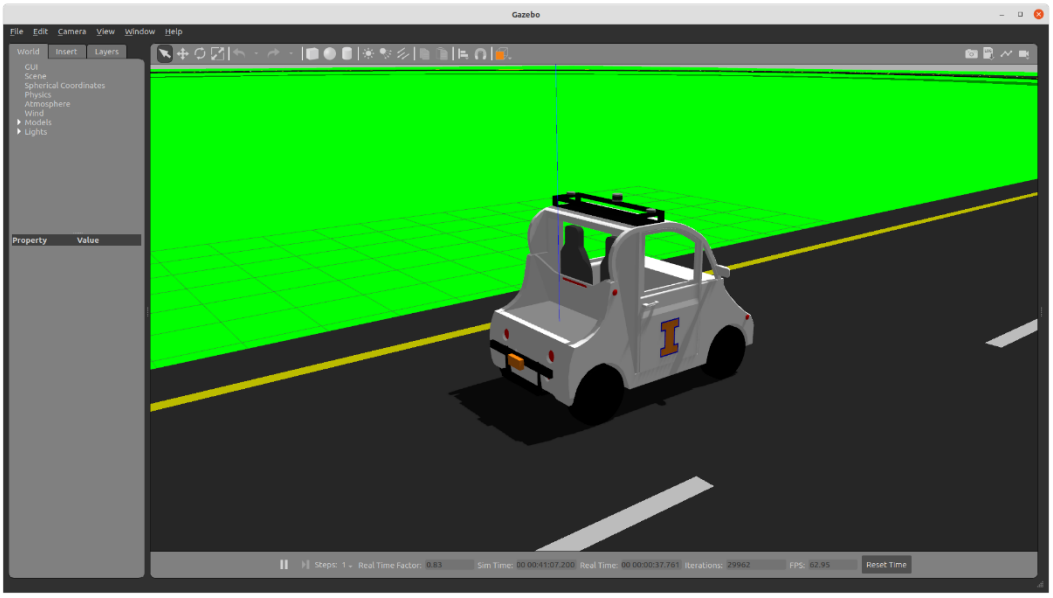


3.2 Launch the Simulator

Simple Track Environment

$ cd ~/demo_ws
$ source devel/setup.bash
$ roslaunch gem_gazebo gem_gazebo_rviz.launch velodyne_points:="true"



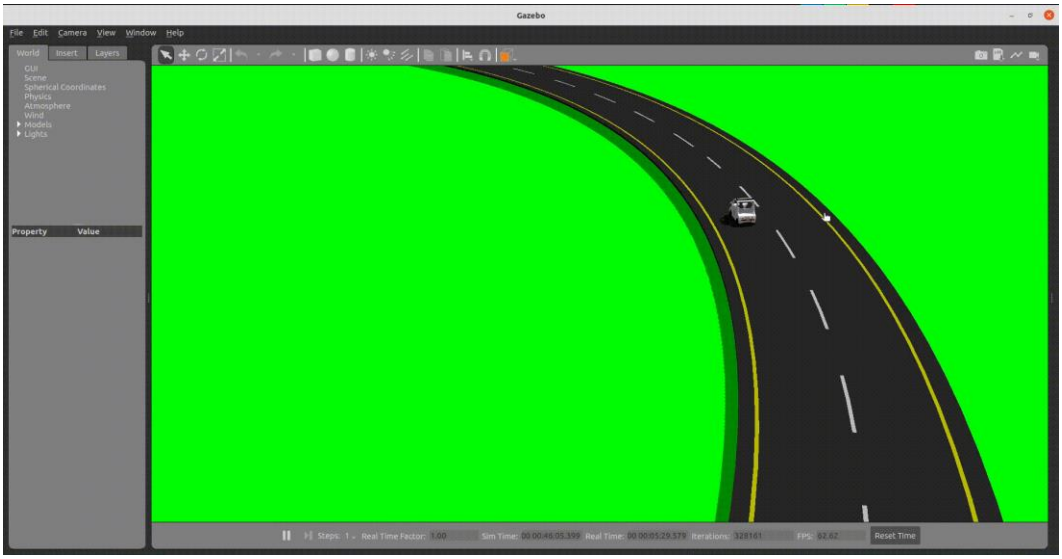


Geometric based Lateral Controller

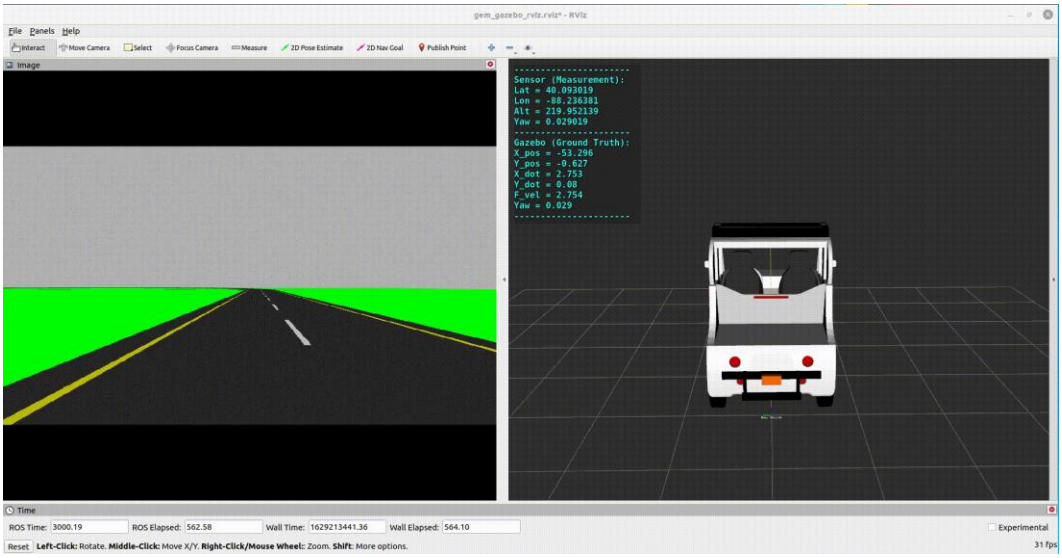
$ source devel/setup.bash
$ roslaunch gem_gazebo gem_gazebo_rviz.launch

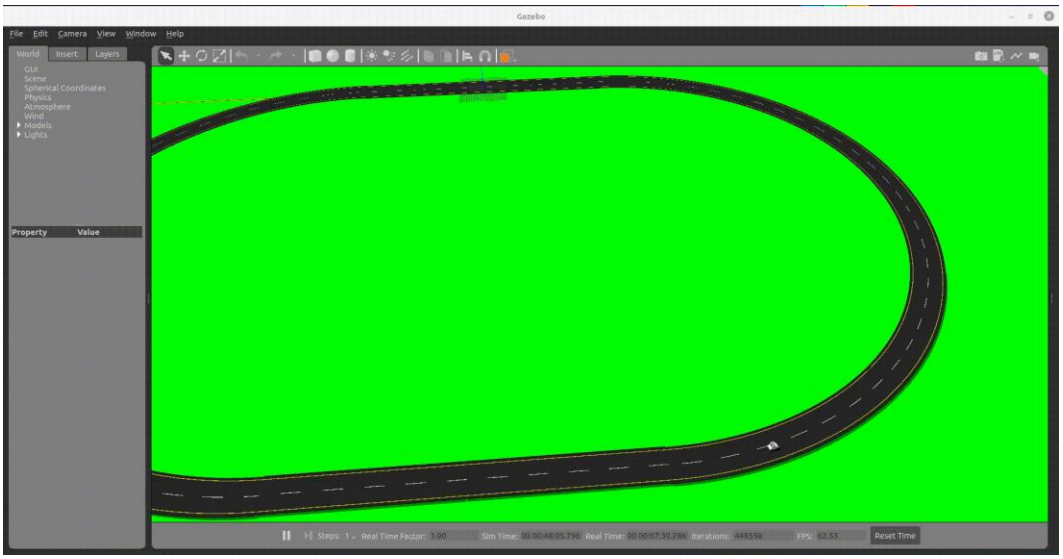
$ source devel/setup.bash
$ roslaunch gem_gazebo gem_sensor_info.launch

$ source devel/setup.bash
$ rosrun gem_pure_pursuit_sim pure_pursuit_sim.py



$ source devel/setup.bash
$ rosrun gem_stanley_sim stanley_sim.py

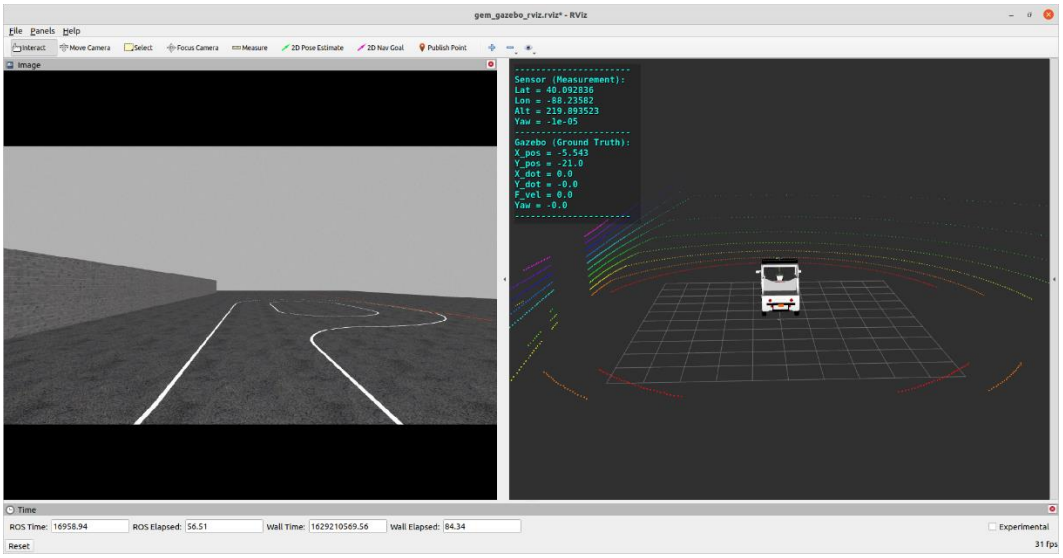


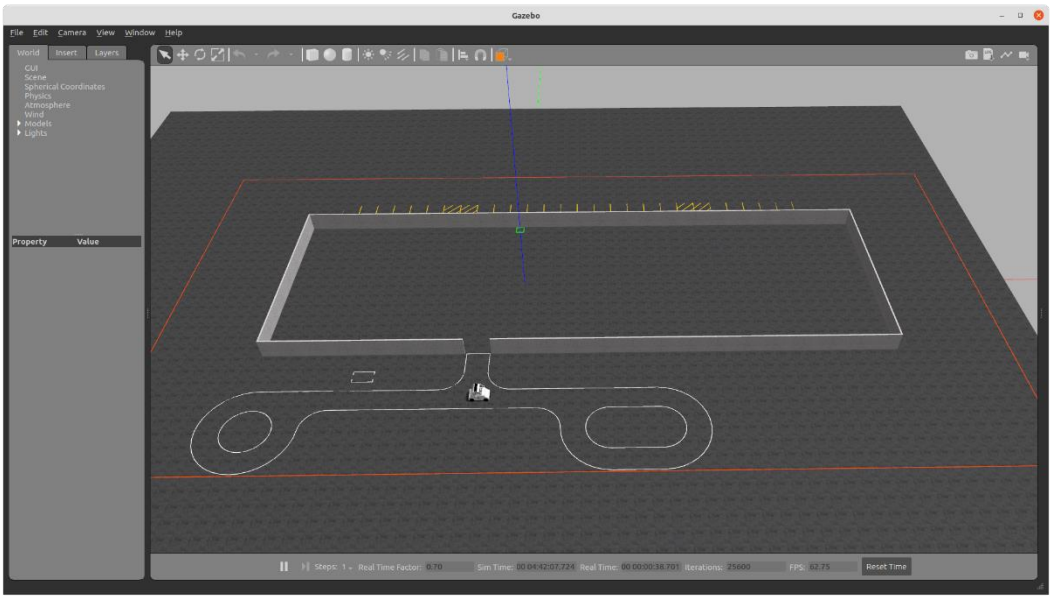

Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Reset Time

Highbay Environment

$ source devel/setup.bash
$ roslaunch gem_gazebo gem_gazebo_rviz.launch world_name:="highbay_track.world" x:=-5.5 y:=-21 velodyne_points:="true"

$ source devel/setup.bash
$ roslaunch gem_gazebo gem_sensor_info.launch





3.3 Coming more

4. Polaris GEM e2 Vehicle (Ubuntu 20.04 + ROS Noetic)

4.1 Introduction

We are using Ubuntu 20.04 and ROS Noetic, the ROS driver for Delphi ESR 2.5 (radar) is not ready yet. At this time, we can not use the radar sensor. Software installed on the SSD drive are listed as below.

Ubuntu 20.04 with ROS Noetic (Python 3)

NVIDIA Driver Version: 460.80

Qt 5.12.11

CUDA 11.0.3

cuDNN 8.1.1

TensorRT 8.0.0.3

OpenCV 4.4.0

pytorch 1.7.1

tensorflow 2.5

4.2 Launch the Demo

Sensor initialization

$ source devel/setup.bash
$ roslaunch basic_launch gnss_sensor_init.launch

Satellite GNSS map visualization

$ source devel/setup.bash
$ roslaunch basic_launch gnss_visualization.launch

Gamepad control initialization

$ source devel/setup.bash
$ roslaunch basic_launch dbw_joystick.launch

GNSS based waypoints tracker using pure pursuit controller

$ source devel/setup.bash
$ rosrun gem_gnss gem_gnss_pp_tracker_pid.py