PolyNav ROS Driver

PolyExplore, Inc.

Build

Copy ROS driver source code folder polyx_nodea to catkin_ws/src/.

Open the terminal and go under catkin_ws, then type the following commands to build ROS driver:

```
catkin_make
```

Serial Port Output

- To start ROS in background, open a terminal and type "roscore"
- Open second terminal to run the ROS talker and type the following commands:

```
cd catkin_ws
source devel/setup.bash
cd src/polyx_nodea
./polyx_nodea.sh
```

- (Optional) Open third terminal to run the ROS listener and type the following commands:

```
cd catkin_ws
source devel/setup.bash
cd src/polyx_nodea
./polyx_nodea_listener.sh
```

Ethernet Output

Poly ROS driver also provides the Ethernet output. To connect to PolyNav System through Ethernet, refer to Section 3.2 in PolyNav System Setup Guide and Section 3.1 in PolyNav Control Software Manual. If the user wants to output messages from
ROS Ethernet, please configure the system to output these messages through Ethernet. See Section 5 in PolyNav Control Software Manual.

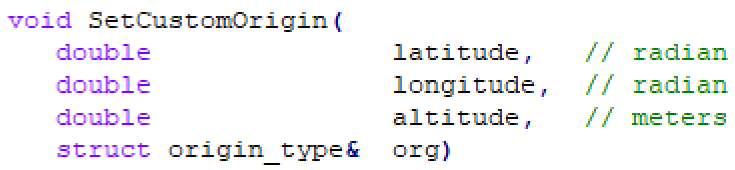
To run ROS driver through Ethernet, open the terminal to run the ROS talker and type the following commands:

```
cd catkin_ws
source devel/setup.bash
cd src/polyx_nodea
./polyx_nodea_eth.sh ipaddress port
```

Where the ipaddress should be 192.168.100.97 or DHCP IP address, port should be 8888.

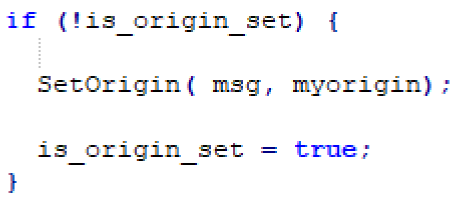
Local Map Origin

By default the ROS driver uses the first navigation solution as the origin. To set a specific position as the origin of your local map use the following function:



```
void SetCustomOrigin(
   double               latitude,   // radian
   double               longitude,  // radian
   double               altitude,   // meters
   struct origin_type&  org)
```

In “polyx_nodea_talker.cpp” file, look for the following part and replace SetOrigin() with SetCustomOrigin().



```
if (!is_origin_set) {

  SetOrigin( msg, myorigin);

  is_origin_set = true;
}
```

Static Heading Event

It may be difficult to initialize the heading by the dual-GNSS antenna system if the system is in a location where the signal is degraded. In this case, the system can be initialized using the static heading event. To generate this event, open a new terminal and follow the steps below:

```
  cd catkin_ws/devel
  source setup.bash
  rosrun polyx_nodea polyx_nodea_heading [heading ZUPT_RMS heading_RMS
duration]
```

where heading, ZUPT_RMS, heading_RMS, and duration are optional parameters and the units are in degrees, m/s, degrees, and seconds, respectively. If the options are not specified the program runs with default message parameters. If you want to edit the default message parameters, just go to

```
 /catkin_ws/src/polyx_nodea/src/
```

and edit "polyx_nodea_heading.cpp" file.

Static Geo-Pose Event

Sometimes, GNSS signals are not available. In this case, the static geo-pose event can be used to initialize or aid the system. Especially, it is possible to hold the position and heading at a specific point. To generate this event, open a new terminal and follow the steps below:

```
cd catkin_ws/devel
source setup.bash
rosrun polyx_nodea polyx_nodea_geopose -p lat lon alt pos_rms -z zupt_rms -h
heading heading_rms -d duration -t roll pitch -g

Options:
-p Latitude Longitude EllipsoidalHeight PositionRMS :(deg, deg, m, m)
-z ZUPTRMS :(m/s)
-h Heading HeadingRMS :(deg, deg)
-d Duration :(seconds)
-t Roll Pitch :(deg, deg)
-g :Turn off GNSS
```

Where Latitude, Longitude, Ellipsoidal Height, PositionRMS, ZUPTRMS, Heading, HeadingRMS, Duration, Roll, Pitch are optional parameters and the units are in degrees, degrees, m, m, m/s, degrees, degrees, seconds, degrees, degrees respectively. Duration is the time duration that we want to send geo-pose messages to system.

```
Valid range of option parameters:
Latitude: -90 ~ 90 deg; Longitude: -180 ~ 180 deg;
PositionRMS: 0.00 ~ 655.35 m; ZUPTRMS: 0.000 ~ 65.535 m/s;
Heading: -180.00 ~ 180.00 deg; HeadingRMS: 0.0 ~ 25.5 deg;
Roll: -180.00 ~ 180.00 deg; Pitch: -90.00 ~ 90.00 deg;
```

If the options are not specified, the program runs with default message parameters. If you want to edit the default message parameters, just go to

```
/catkin_ws/src/polyx_nodea/src/
```

and edit "polyx_nodea_geopose.cpp" file.

Event Generator (GUI)

PolyExplore Inc. also provide a GUI to handle static Geo-Pose event. If the users want to use the Event Generator, feel free to contact PolyExplore to get the source code.

To build this EventGenerator, make sure to install ROS “qt_build” package. Use the following commands to install:

```
sudo apt-get install ros-kinetic-qt-build
```

And then go under catkin_ws, type the following commands to build:

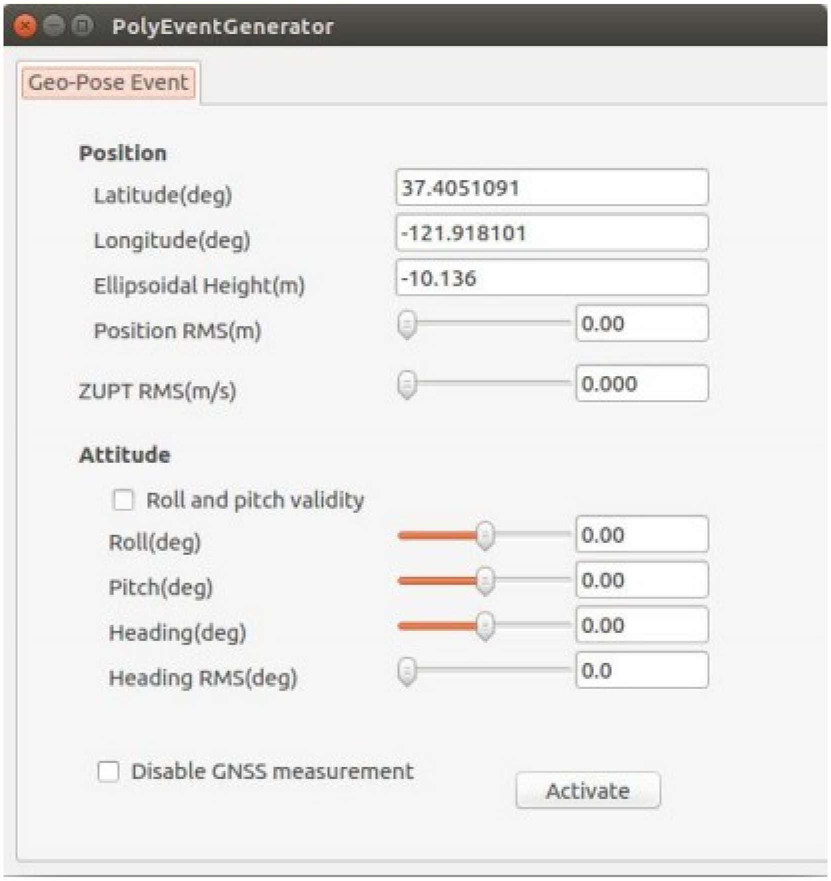
```
catkin_make
```

To run this Event Generator, open a new terminal and follow the steps below:

```
cd catkin_ws
```

```
 source devel/setup.bash
 rosrun eventgenerator eventgenerator
```

See the Figure below. After setting everything well, click ‘Activate’ button and then it will keep sending Geo-Pose messages to system. You can notice that the ‘Activate’ button changed to be ‘Deactivate’. If you want to stop sending messages, just click ‘Deactivate’ button.



Position
Latitude(deg) 37.4051091
Longitude(deg) -121.918101
Ellipsoidal Height(m) -10.136
Position RMS(m) 0.00
ZUPT RMS(m/s) 0.000

Attitude
□ Roll and pitch validity
Roll(deg) 0.00
Pitch(deg) 0.00
Heading(deg) 0.00
Heading RMS(deg) 0.0

□ Disable GNSS measurement

IMU Data

The ROS driver can output both the scaled raw IMU data and the corrected IMU data if the user configured the system to output these messages. Note that the corrected IMU data are available only after the initialization of the inertial navigator. This message contains IMU data corrected for the sensor biases estimated by the fusion algorithm.

Geoid Height

The Geoid message contains the height of the Geoid above the ellipsoid. Thus the height above Geoid, treated normally as the height of mean sea level (MSL), can be computed as follows:

Height above MSL = Height above ellipsoid - Geoid height.