ROS User Manual for 0x series robots

Nex Robotics Pvt. Ltd.

Version 2.1

Read sentences carefully which are marked with ⚠ caution symbol.

⚠ Important:

User must go through hardware and software manuals before using robot.
This manual is intended for 0x series of robots.

⚠ Safety precautions:

- ✔ Robot's electronics is static sensitive. Use robot in static free environment.
- ✔ Do not access any part of the robot unless robot is in the anti static environment and user is wearing anti static strap.
- ✔ If robot's battery low buzzer starts beeping, immediately charge the batteries.
- ✔ To prevent fire hazard, do not expose the equipment to rain or moisture.
- ✔ Refrain from dismantling the unit or any of its accessories once robot is assembled.
- ✔ Charge the battery only with the charger provided with the robot.
- ✔ Charge the battery in the open area and on the concrete or ceramic flooring.
- ✔ Never allow battery to deep discharge. If it is deep discharged, charger will refuse to charge the battery because of safety concerns.
- ✔ Mount all the components with correct polarity.
- ✔ Keep wheels away from long hair or fur.
- ✔ Keep your hands away from the wheels. Do not wear loose clothes while operating the robot. Loose cloth may get entangled in robot's wheels and can cause serious injury.
- ✔ Keep the robot away from the wet areas. Contact with water will damage the robot.
- ✔ To avoid risks of fall, keep your robot in a stable position.
- ✔ Do not attach any connectors while robot is powered ON.
- ✔ Never leave the robot powered ON when it is not in use.
- ✔ Before operating the robot, make sure that you have access to at least "Class A/B" type fire extinguisher.

⚠Inappropriate Operation:

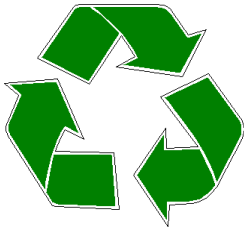
Inappropriate operation can damage your robot. Inappropriate operation includes, but is not limited to:

- ✔ Dropping the robot, running it off an edge, or otherwise operating it in an irresponsible manner.
- ✔ Interfacing new hardware without considering compatibility
- ✔ Overloading the robot above its payload capacity.
- ✔ Exposing the robot to wet environments.
- ✔ Continuing to run the robot after hair, yarn, string, or any other item has become entangled in the robot's axles or wheels.
- ✔ All other forms of inappropriate operation.
- ✔ Using robot in areas prone to static electricity.

Notice

The contents of this manual are subject to change without notice. All efforts have been made to ensure the accuracy of contents in this manual. However, should any errors be detected, NEX Robotics welcomes your corrections. You can send us your queries / suggestions at info@nex-robotics.com

- **Robot's electronics is static sensitive. Use robot in static free environment.**
- **Read the Robot's manual completely before start using this robot**



Recycling:

Almost all of the robot parts are recyclable. Please send the robot parts to the recycling plant after its operational life. By recycling we can contribute to cleaner and healthier environment for the future generations.

⚠User must go through hardware and software manuals before using 0x series of Robots.

Commonly Used Abbreviations in manual:

Short Forms	Full Forms
IR	*Infrared*
GPS	*Global Positioning System*
DGPS	*Differential Global Positioning System*
GPIO	*General Purpose Input Output*
S	*Success*
F	*Failure*
MSB	*Most Significant Byte*
LSB	*Least Significant Byte*
UL	*Ultrasonic*
WL	*White Line*
GUI	*Graphical User Interface*

⚠Important notes related to safety feature of robot:

Behavior during robot safety ON feature

1. Max velocity of robot gets limited to a safe value. This safe max velocity for Firebird VI robot is 0.078 m/Sec.

Behavior during robot safety OFF feature

1. Max velocity of robot can be altered up to robot's maximum reachable velocity.
2. Max reachable linear velocity of Firebird VI robot is +/- 0.4 m/Sec.
3. Max reachable angular velocity of 0x Delta robot is +/- 8 radians/Sec.

Table of Contents

1 Introduction.....7

2 Installation and Getting Started.....7

3 Reporting bugs and making feature requests.....7

4 ros0xrobot node.....7

4.1 Subscribed topics.....7
4.2 Published topics.....7
4.3 Parameters.....8
4.4 Running ros0xrobot node.....8

5 ROS node for Hokuyo laser range finder.....8

6 ROS node for SICK LMS 1xx laser range finder.....8

7 ROS node for IMU.....9

8 ROS node for GPS.....9

9 Launch files.....10

1 Introduction

ros0xrobot provides ROS interface for Nex Robotics robot bases in 0x series (pronounced as Ox) and Fire Bird series. It is supported by 0xRobotCpp library. Information from the robot base, velocity and acceleration control, is implemented via a ros0xrobot node, which publishes topics providing data received from the robot's embedded controller by 0xRobotCpp library, and sets desired velocity, acceleration and other commands in robot when new commands are received from command topics.

2 Installation and Getting Started

Instructions on installing ROS can be found here.

3 Reporting bugs and making feature requests

Bugs and requests can be reported here: rosdev@nex-robotics.com

4 ros0xrobot node

ros0xrobot is available under topic name `ros0xrobot`/topic name. You can remap these to different names if necessary. Please refer Names, Remapping Arguments and the <remap> tag for launch files for more information. The ros0xrobot node requires *0xRobotCpp* library for establishing communication with robot base. The library is available in the */lib* folder of source provided.

4.1 Subscribed topics

`cmd_vel` (geometry_msgs/Twist)
Receives new velocity commands. The received velocities will be updated and maintained by robot. There is no need to send the velocity repeatedly to maintain the velocity. If the robot is kept in timeout mode, then any communication/command needs to be sent before timeout to keep the robot moving.

4.2 Published topics

`pose` (nav_msgs/Odometry)

publishes odometry information (rate depends on the robot, normally 10Hz)

`sonar` (sensor_msgs/PointCloud)
publishes sonar readings. Requires enableSonar parameter to be set to true.

`imu` (sensor_msgs/Imu)

publish imu readings. Requires enableSonar parameter to be set to true.

4.3 Parameters

~port (string, default: /dev/ttyUSB0)
Serial port device the robot is is connected to. Set to /dev/ttyUSBx depending on port on which robot is connected.

CountsPerRev (`int`, default: 3840)
Set counts per revolution of the robot. Refer hardware manual of robot for appropriate value.

WheelDiameter (`float`, default: 100.0)
Set wheel diameter of robot in mm.

AxelLength (`float`, default: 290.0)
Set axle length in mm.

enableSonar (bool, default: false)
Enable or disable Sonar. This will cause the node to start publishing sonar messages when enabled.

enableImu (bool, default: false)
Enable or disable IMU. This will cause the node to start publishing imu messages when enabled.

4.4 Running ros0xrobot node

For basic robot node

```
$roslaunch ros0xrobot  ros0xrobot_minimal.launch
```

For node with on-board sonar

```
$roslaunch ros0xrobot  ros0xrobot_sonar.launch
```

For node with on-board IMU

```
$roslaunch ros0xrobot  ros0xrobot_imu.launch
```

5 ROS node for Hokuyo laser range finder

The ROS node to provide access to SCIP 2.0-compliant Hokuyo laser range finders (including 04LX) is available on http://wiki.ros.org/hokuyo_node. To run this node a launch file named ros0xrobot_hokuyo.launch is provided in launch folder. To launch this node...

```
$roslaunch ros0xrobot  ros0xrobot_hokuyo.launch
```

You can modify the transform location in launch file to suit the location of sensor as per your choice.

6 ROS node for SICK LMS 1xx laser range finder

The lms1xx package that contains a basic ROS driver for the SICK LMS1xx line of LIDARs is available on http://wiki.ros.org/LMS1xx. To run this node a launch file named ros0xrobot_lms1xx.launch is provided in launch folder. To launch this node...

```
$roslaunch ros0xrobot  ros0xrobot_lms1xx.launch
```

The IP address of LMS 1xx and topic name for publishing messages can be specified in the launch file.

You can modify the transform location in launch file to suit the location of sensor as per your choice.

7 ROS node for IMU

The 9DOF imu unit can be accessed in ROS by using razor_imu_9dof node. Install the node by following the procedure given on ROS wiki here http://wiki.ros.org/razor_imu_9dof#Install_ROS_razor_imu_9dof_Package.
Connect the unit to host running ROS. It will be available over a communication port /dev/ttyUSBn, where n is the port number. Set the correct port number in config file named my_razor.yaml.

1. Go to imu node directory by

```
$roscd razor_imu_9dof/config
```

2. Open the config file named my_razor.yaml

```
$sudo nano my_razor.yaml
```

3. Modify the port parameter. Save the file.

A launch file is provided to run the node.

To run the node,

1. Open a terminal window and

```
$roslaunch razor_imu_9dof razor-pub.launch
```

2. The published messages can be seen by echoing the imu topic as follows

```
$rostopic echo /imu
```

8 ROS node for GPS

GPS access is provided through *nmea_navsat_driver*. For more information please refer http://wiki.ros.org/nmea_navsat_driver . We use a GPS module with 38400 baud rate and update rate of 5 Hz. A launch file is provided to run the node.

1. Open the terminal window and

```
$roslaunch ros0xrobot ros0xrobot_gps.launch
```

2. The published messages can be seen by echoing the gps topic as follows

$rostopic echo /fix

You can modify the transform location in launch file to suit the location of sensor as per your choice.

for using GPS for localization, please refer ...
http://wiki.ros.org/robot_localization/Tutorials/GPS%20Integration

You can modify the transform location in launch file to suit the location of sensor as per your choice.

9 Launch files

The launch files are available for various devices present on robot. They are present in launch folder. The launch files will work only if the corresponding device is present on the robot.

1. ros0xrobot_minimal.launch : This launch file launches a basic robot node.
2. ros0xrobot_hokuyo.launch : This will launch robot node and hokuyo laser range finder node.
3. ros0xrobot_imu.launch : This will launch the robot node with imu enabled.
4. ros0xrobot_sonar.launch : This will launch robot node with sonar enabled.
5. ros0xrobot_gps.launch : This will launch the robot node with gps node.
6. ros0xrobot_teleop.launch : This will launch the teleop node to control the robot movement using keyboard.
7. ros0xrobot_all.launches : This will launch all the sensors on robot (sonar, imu, gps, hokuyo). This requires all sensors to be present and switched ON.