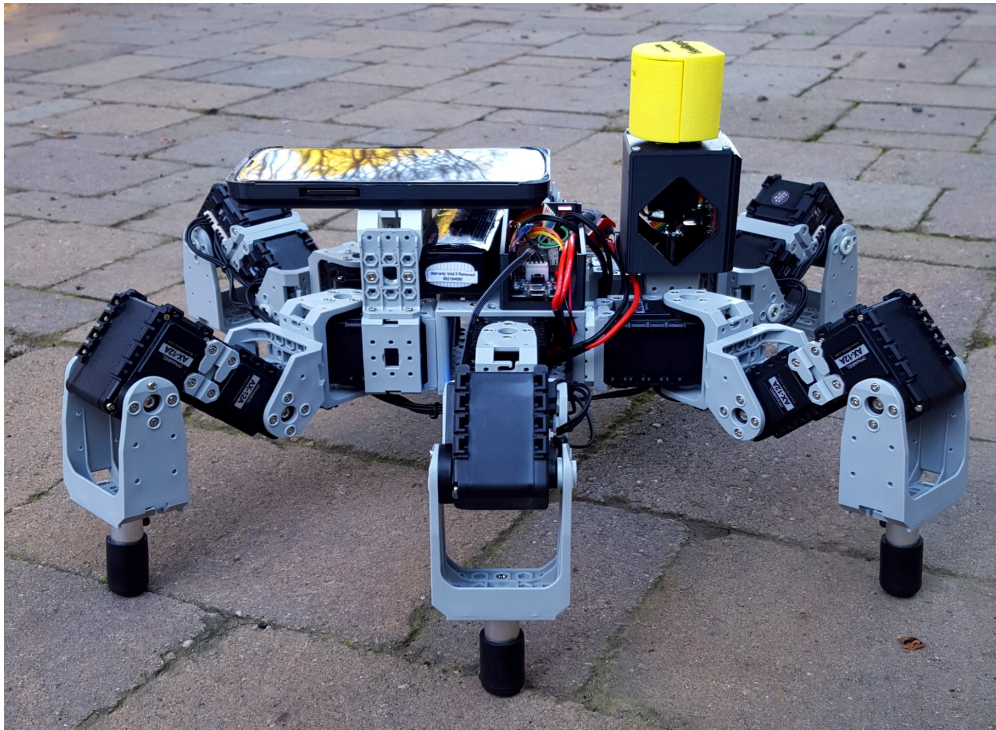


Rhoeby Hexapod Robot – Getting Started V1.0, 20160314

Rhoeby Dynamics – Hexapod Robot – Getting Started



Getting Started with the Rhoebly Hexapod Robot

Thank you for your purchase, and welcome to Rhoebly Dynamics products! This document provides a quick overview of the steps required to run ROS Navigation with the Rhoebly Hexapod platform.

Pre-requisites

To proceed, you will need the following items:

- A ROS enabled PC (for robot control, and running the ROS application RViz)
- Basic Linux and ROS skills for file editing, network configuration, etc
- Your ROS Navigation-capable Hexapod robot!

Please be sure to carefully **follow all steps** described in the document. Failure to do so could result in damage to your robot.

Unpacking Instructions

1. Remove all packaging materials
2. Remove and charge the robot battery using the charger provided (all lights green when fully charged)
3. Remove and charge the phone using the provided USB cable and charger
4. Place the phone in the phone holder on the robot
5. Connect the battery monitor
6. Insert and connect the battery to the robot
7. Sit robot with legs evenly spaced around in a 'hexagon' formation [IMPORTANT]

Installation Instructions

Follow these installation instructions to get your robot set up to work with your ROS-enabled PC. Note that some steps may vary depending on the particulars of your Linux/PC environment.

1. Install the ROS Navigation stack (prefer Indigo release) on to your PC, see:

<http://wiki.ros.org/navigation>

As a quick sanity check, verify correct install of robot_pose_ekf, as follows:

```
roslaunch robot_pose_ekf robot_pose_ekf
```

You should see output similar to:

```
[ INFO] [1456852250.252138379]: output frame: odom_combined
```

```
[ INFO] [1456852250.252771291]: base frame: base_footprint
```

Press Ctrl-C to quit.

2. Download the Hexapod ROS driver code to your PC:

https://github.com/Rhoeby/hexapod_ros

OR

https://github.com/Rhoeby/hexapod_ros/archive/master.zip

3. Install Hexapod ROS driver into your catkin workspace

eg. place the 'hexapod_ros' source tree in ~/catkin_ws/src

(See http://wiki.ros.org/catkin/Tutorials/create_a_workspace)

4. Build the Hexapod ROS driver:

```
eg. cd ~/catkin_ws; catkin_make
```

5a. Setup the robot Bluetooth connection:

Power up the robot.

Pair the device in Linux using the Bluetooth Settings.

Device name: ROBOTIS BT-210

Pairing code: 0000

After pairing, determine the device address (check in the Bluetooth Settings):

eg. B8:63:BC:00:1F:8C

Bind the bluetooth device:

```
sudo rfcomm bind /dev/rfcomm0 B8:63:BC:00:1F:8C
```

As a sanity check, run:

```
sudo cat /dev/rfcomm0
```

You should see output similar to:

```
user@ubuntu:~$ sudo cat /dev/rfcomm0
1000, 370
1000, 370
1000, 365
```

Set permissions on the device:

```
sudo chmod 666 /dev/rfcomm0
```

The bluetooth connection to the robot is now ready.

5b. Test Hexapod connectivity:

Power up, or power-cycle the robot (turn off and on again), then run (on PC, from the catkin workspace directory):

```
source devel/setup.bash
roslaunch hexapod_ros hexapod_ros -np
```

Robot should initiate start behavior (scanner spins, robot stands up). Use arrow keys in terminal window to tele-op the robot. When done, press 'q' to quit.

6. Start app on phone (SensoDuino / Rhoebly app)

Run the phone app 'SensoDuino'

7. Test the phone connectivity (in **new** linux terminal):

```
sudo sdptool add --channel=22 SP
sudo rfcomm listen /dev/rfcomm1 22
sudo cat /dev/rfcomm1
```

You should see data similar to:

```
>4,25362,-7.324219E-4,-0.0010528564,-0.0010528564
>1,25363,-1.1710815,0.075424194,7.155319
>4,25364,3.3569336E-4,1.5258789E-5,1.5258789E-5
>1,25365,-1.1722717,0.08732605,7.121994
```

The presence of the above output indicates that the PC is receiving data from the phone. If all the above steps are completed successfully, jump to the next section 'Running'.

NOTE: the above steps 5 and 7 do not apply to Linux VM (Windows-hosted Virtual Machine) setups.

Alternative instructions for Linux VM:

5a. Setup bluetooth device 'ROBOTIS BT-210' in Windows ("Add Device" in Bluetooth)

Pair the robot in Windows. The robot uses the pairing code: '0000'. Note the COM port used.

Setup the COM port in Linux VM. You will need to modify the startup script to match the serial port used (see section 'Running' for more details).

7.1 Pair the phone in Windows. After pairing, determine the COM port used and add that to your serial port setup in your VM Settings. For the phone (eg. on COM10), must set "Connected" in the VM Settings. Also, may need to toggle "Connected" in the VM Settings:

7.2. Uncheck "Connected"

7.3. Start SensoDuino (on phone)

7.4. Check "Connected"

7.5. use 'sudo cat /dev/ttyS1' to verify data is flowing

*** Please be sure to **complete all of the above steps** successfully before continuing ***

Running

1. power cycle robot (turn off and on again)

2. In Linux terminal (on PC):

```
cd hexapod_ros/scripts
./rhoeby_init.sh
./rhoeby_mapping.sh
```

If everything is setup correctly, it should run RViz and start displaying a partially built map. You can now tele-op the robot to build a map.

Notes

The startup scripts assume:

- robot will connect via '/dev/rfcomm0'
- phone will connect via '/dev/ttyS1'

However, these items are configurable in the script. You may need to set it up according to your environment (which might use a different configuration). See 'hexapod_ros/scripts/rhoeby_init.sh' and 'hexapod_ros/scripts/rhoeby_mapping.sh' for details. Please make copies of the scripts and then modify the originals to suit.

For further assistance, visit www.rhoeby.com/support, or send email to 'support@rhoeby.com'.

