RB-1 BASE MOBILE PLATFORM

System Software and Architecture Manual

Version 2.0

RBTNK-DOC-170727A

Robotnik Automation, S.L.L.

Contents

1. Software Architecture

This manual describes the RB-1 BASE software architecture.

The RB-1 BASE software architecture is based on ROS (Robot Operating System www.ros.org).

The second chapter gives a brief description of the ROS open source architecture. The third chapter describes the implementation of the ROS architecture in the RB-1 BASE robot. The different robot software components are described then.

2. ROS Architecture

ROS is an open-source meta-operating system for your robot that provides inter-process message passing services (IPC) in a network.

ROS is also an integrated framework for robots that provides:

- Hardware abstraction layer
- Low level device control
- Robot common functionality (simulation, vision, kinematics, navigation, etc.)
- IPC
- Package and stack management

ROS provides libraries and tools to ease the development of robot software in a multi-computer system.

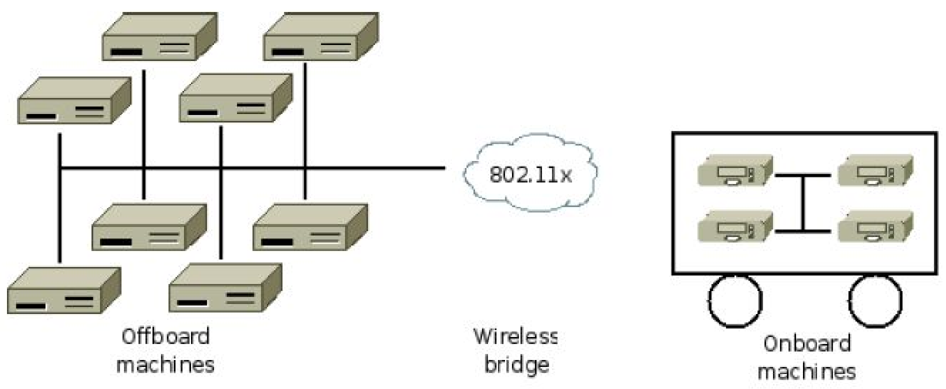


Figure 1 - ROS typical network configuration

ROS offers a framework to solve common research and development needs of robot systems:

- Cooperation of different research groups
- Proven functionality
- Easy and robust access to robotics hardware

One of the main objectives of ROS is the code reusability. This objective is fulfilled by a large and growing community of developers that share their results worldwide, and by the inclusion of other robot frameworks (ROS integrates Player/Stage, Orocos, etc.) and other open-source components (like Gazebo or Openrave).

ROS integrates additional development tools like rviz (simulation of complete robots and environments with maps), rqt_graph (visualization of node interconnection), rosbag (extreme useful data logging utility), etc.

For detailed systems descriptions, tutorials, and a really important number of stacks and packages, please visit www.ros.org.

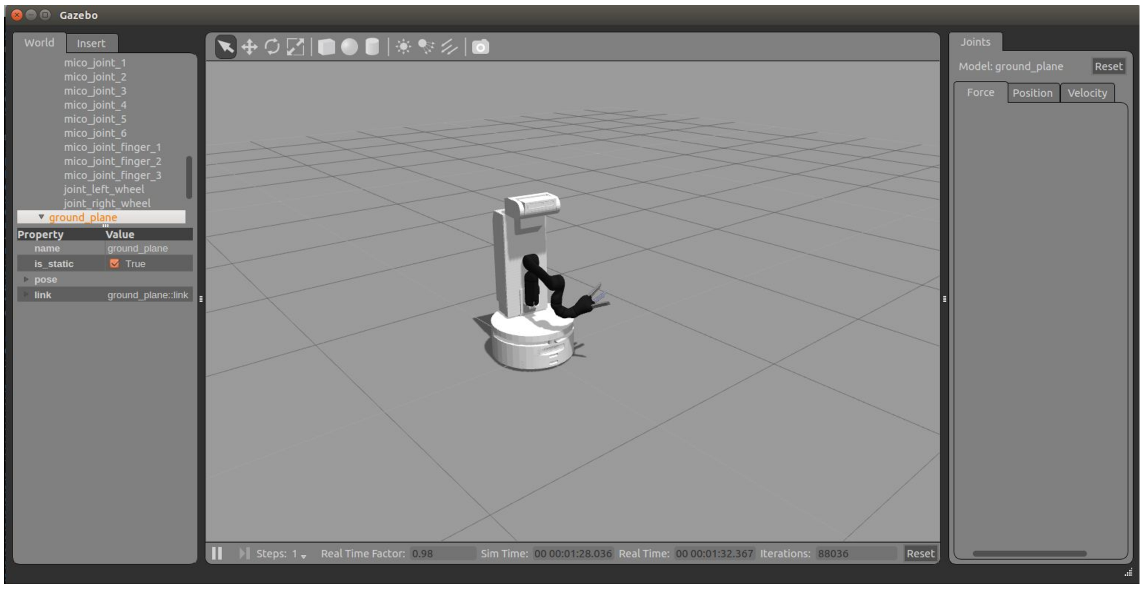


Figure 2 – ROS gazebo robot simulation

3. RB-1 BASE Robot Architecture in ROS

RB-1 BASE ROS architecture is the result of the cooperative operation of several nodes. The robot has ROS Kinetic installed, and robot specific software is provided in 6 stacks (metapackages):

- **rb1_base_robot**: includes all the packages required for the real robot base control.
- **rb1_base_sim**: allows the simulation of the robot base and its sensors in Gazebo.
- **rb1_base_common**: packages shared between the robot and simulation stacks, i.e. robot description, navigation, etc.
- **robotnik_base_hw:** packages based on ros_control architecture compatible with most of the Robotnik's motor hardware.

Additional documentation about the contained packages can be found in the README.md of the repos:

https://github.com/RobotnikAutomation/rb1_base_sim
https://github.com/RobotnikAutomation/rb1_base_robot
https://github.com/RobotnikAutomation/rb1_base_common
https://github.com/RobotnikAutomation/robotnik_base_hw

The simulated robot publishes almost the same data as the real robot and accepts the same commands thus allowing to easy test programs in simulation and directly testing on the real robots.

The default workspace of the robot is located at:

```
> /home/rb1/catkin_ws/
```

In addition the robot includes additional packages depending on the installed components (installed via apt-get or from sources). The source code packages installed locally is located at:

```
> /home/rb1/sources
```

Local packages are linked symbolically from the workspace.

A number of components used by the robot can be found in the gitHub repo:

https://github.com/RobotnikAutomation

Other required packages are:

- **robotnik_msgs**

Simple package that contains standard services and messages commonly used in mobile robots.

https://github.com/RobotnikAutomation/robotnik_msgs

- **robotnik_sensors**

A package that contains the URDF files that describe the sensors that are mounted in the robot. These are used for simulation, but also for the robot description, that is used for visualization or packages as MoveIt!.

https://github.com/RobotnikAutomation/robotnik_sensors

Other **optional components** that may be installed in your robot and that can be downloaded from the gitHub. The most common are:

- **orbbec_camera**

Package that implements a device driver for Orbbec Astra RGBD cameras.

- **hokuyo_node/urg_node**

Driver to read the scan data from Hokuyo laser devices.

- **mavros + imu_complementar_filter**

Drivers for the Pixhawk IMU

3.1 Odometry

The robot's odometry is calculated by using the encoders wheels plus the IMU.

In order to have a good and reliable estimation it is very important that the IMU is calibrated to avoid the errors produced by the temperature differences.

The calibration procedure is the following:

```
>rosservice call /calibrate_imu_gyro "{}"
success: True
message: ''
```

In the terminal running the controller should appear:

```
[ INFO] [1497541536.032796269]: FCU: [cal] calibration started: 2 gyro
[ INFO] [1497541537.332209945]: FCU: [cal] progress <5>
[ INFO] [1497541538.382425775]: FCU: [cal] progress <10>
[ INFO] [1497541539.381626510]: FCU: [cal] progress <15>
[ INFO] [1497541540.432835366]: FCU: [cal] progress <20>
[ INFO] [1497541541.482041834]: FCU: [cal] progress <25>
[ INFO] [1497541542.481378200]: FCU: [cal] progress <30>
[ INFO] [1497541543.532456996]: FCU: [cal] progress <35>
[ INFO] [1497541544.582835750]: FCU: [cal] progress <40>
[ INFO] [1497541545.634872674]: FCU: [cal] progress <45>
[ INFO] [1497541546.630241336]: FCU: [cal] progress <50>
[ INFO] [1497541547.681402893]: FCU: [cal] progress <55>
[ INFO] [1497541548.730623022]: FCU: [cal] progress <60>
[ INFO] [1497541549.781817885]: FCU: [cal] progress <65>
[ INFO] [1497541550.780146296]: FCU: [cal] progress <70>
[ INFO] [1497541551.830358689]: FCU: [cal] progress <75>
[ INFO] [1497541552.880580533]: FCU: [cal] progress <80>
[ INFO] [1497541553.930732304]: FCU: [cal] progress <85>
[ INFO] [1497541554.981977490]: FCU: [cal] progress <90>
[ INFO] [1497541555.981183073]: FCU: [cal] progress <95>
[ INFO] [1497541557.032388460]: FCU: [cal] progress <100>
[ INFO] [1497541557.280761856]: FCU: [cal] calibration done: gyro
```

The duration of the procedure is around 30 seconds.

Note: Don’t move the robot during the calibration procedure, otherwise the calibration will fail and the measurements will be incorrect.

4. Network configuration

RB-1 BASE is equipped with a router that provides connectivity through ethernet and wireless. The network configuration is as follows:

IP Range: 192.168.0.X
Netmask: 255.255.255.0
Gateway: 192.168.0.1

DHCP: 192.168.0.[50-100]

Wifi SSID: RB1_*SerialNumber*
Wifi Password: R0b0tn1K (R and K capital letters)

RB-1 BASE router and CPU have the following configuration:

RB-1 BASE Router:
User/Password: root / R0b0tn1K (R and K capital letters)
Router IP Address: 192.168.0.1

RB-1 BASE CPU:
User/Password: rb1 / R0b0tn1K (R and K capital letters)
RB-1 BASE IP Address: 192.168.0.200

The easiest way to access the RB-1 BASE Mobile Platform is to use a Secure Shell (SSH) client.

First, you need to connect to the RB-1 BASE Network using the RB-1 BASE Router SSID and password. Then, open your preferred SSH client and connect using the RB-1 BASE CPU IP Address, user and password. If you use the OpenSSH client:

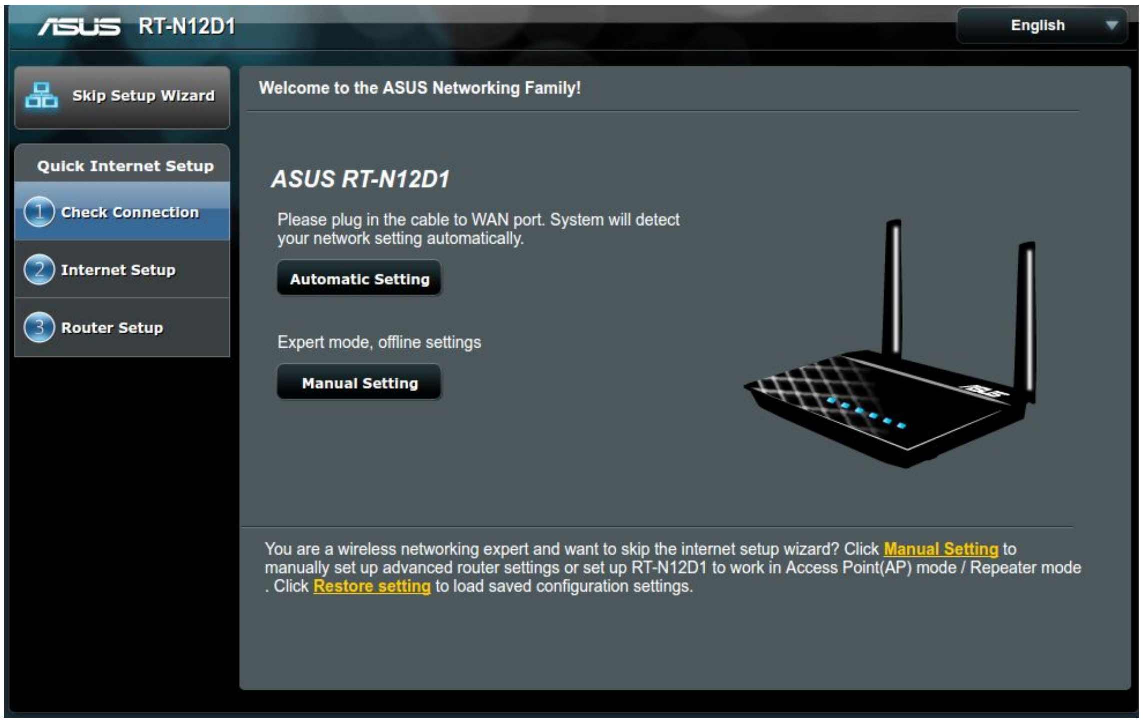
```
user@remote:~$ ssh rb1@192.168.0.200
```

4.1 Configuring the router as a repeater

<u>Asus RT-N12D1</u>

The router can be configured as repeater, enabling the robot to connect to an external WiFi network and extending the range of it.

1. Acces to the router panel via a network browser (router.asu.com or the current ip address)
2. Reset the configuration to factory values or go to the Setup Wizard.
3. Check connection:
    a. Select Manual Setting


RT-N12D1

English

Skip Setup Wizard

Quick Internet Setup

1 Check Connection

2 Internet Setup

3 Router Setup

Welcome to the ASUS Networking Family!

ASUS RT-N12D1

Please plug in the cable to WAN port. System will detect your network setting automatically.

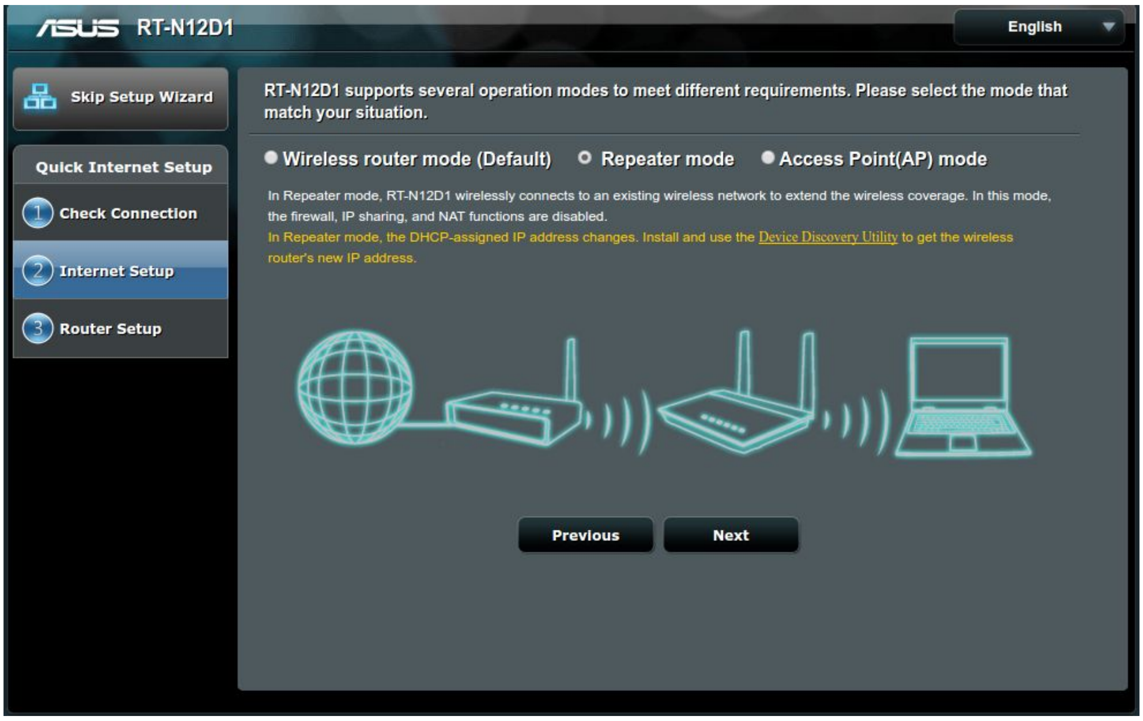
Automatic Setting

Expert mode, offline settings

Manual Setting

You are a wireless networking expert and want to skip the internet setup wizard? Click Manual Setting to manually set up advanced router settings or set up RT-N12D1 to work in Access Point(AP) mode / Repeater mode . Click Restore setting to load saved configuration settings.

4. Login information setup
5. Internet Setup:
   a. Select "Repeater Mode"


RT-N12D1

English

Skip Setup Wizard

Quick Internet Setup

1 Check Connection

2 Internet Setup

3 Router Setup

RT-N12D1 supports several operation modes to meet different requirements. Please select the mode that match your situation.

Wireless router mode (Default) [✓ Repeater mode] Access Point(AP) mode

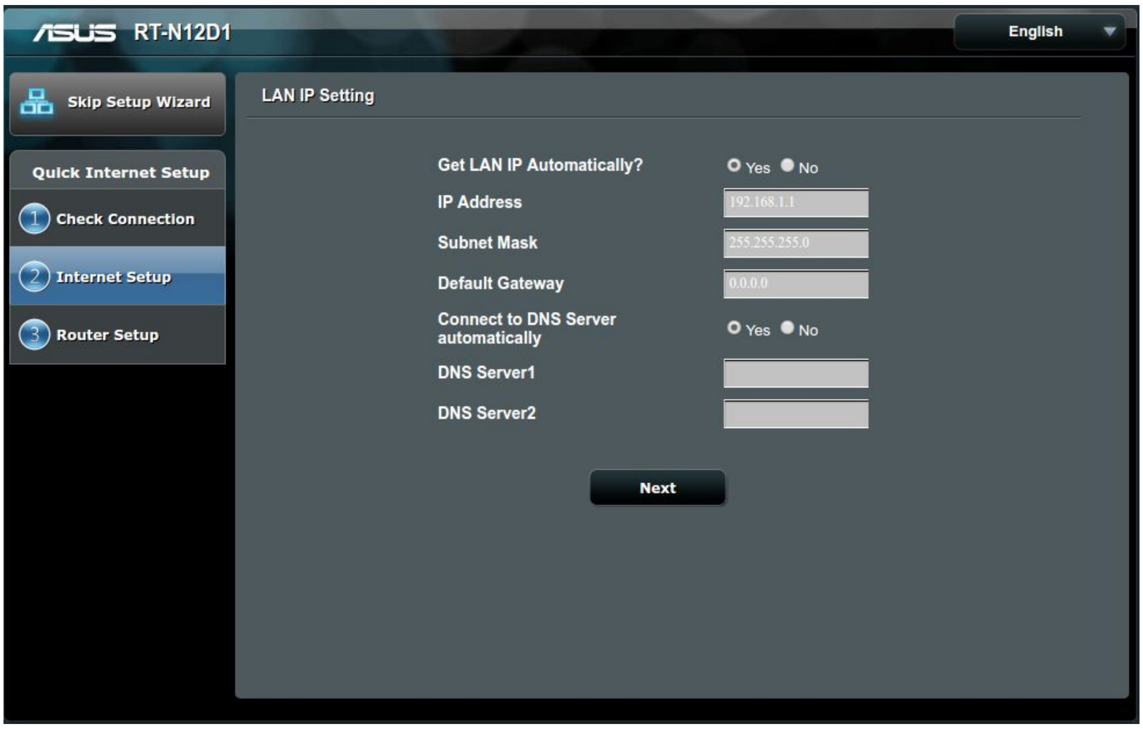
In Repeater mode, RT-N12D1 wirelessly connects to an existing wireless network to extend the wireless coverage. In this mode, the firewall, IP sharing, and NAT functions are disabled.

In Repeater mode, the DHCP-assigned IP address changes. Install and use the Device Discovery Utility to get the wireless router's new IP address.

Previous

Next

6. Select the network you want to connect.
7. Wireless settings.
   a. Use the default settings
8. LAN IP Settings
   a. Set 'Yes' to get the LAN IP automatically.



LAN IP Setting

Get LAN IP Automatically?	[✓ Yes] No
IP Address	192.168.1.1
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
Connect to DNS Server automatically	[✓ Yes] No
DNS Server1	
DNS Server2	

9. Apply the changes and wait until the router applies the modifications.

NOTE: don’t forget to update the network configuration in the pc controller in order to work with the new ip.

5. Robot startup

5.1 Start-up sequence

Have a look to the back panel of the robot:

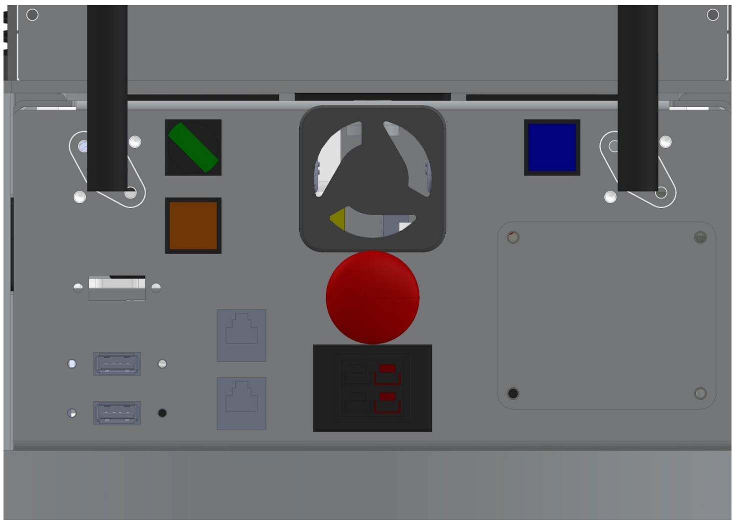


Figure 3. RB-1 BASE back panel.

The following elements are the main components of the back panel necessary to start-up the robot:

🟩	**ON/OFF switch**
🟦	**CPU power button**
🟫	**Motors reset button(only with torso)**
🔴	**Emergency button**

The general ON/OFF SWITCH (green) must be activated for giving energy to all the elements of the system. Press the CPU POWER BUTTON (blue) button to turn ON the computer, the blue button will light up. At this moment the PC (Linux) starts-up and loads all the necessary files for booting.

After booting, it it is possible to connect to the system in a remote way or connect to the robot manually.

To move the robot, the EMERGENCY BUTTON (red) must be pulled out and the RESET BUTTON (orange) must be pressed once.

NOTE: Remember that the robot is able to reach high speeds. Please be careful and use the higher speeds only in wide areas.

5.2 Robot controllers

All the robot controllers are launched automatically (see section 7 for more information).

5.3 PS4 Dualshock Controller

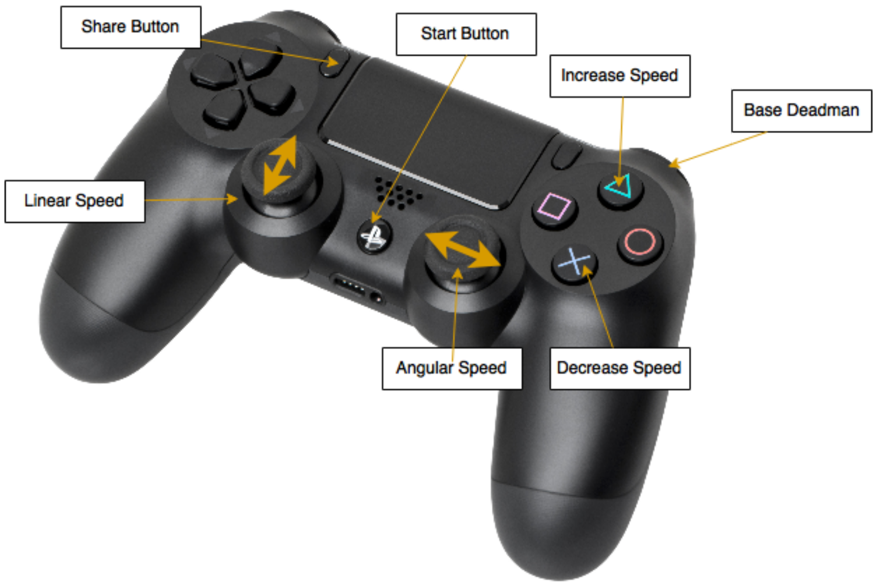
This Game-pad has a smooth and precise control. The Bluetooth receiver is connected to an USB port at the computer.

To control the movements of the Robot and its devices with this controller, please follow the next instructions:

1. Switch on the Robot.
2. Switch on the computer pressing the blue button. (The computer indicatoris light up)
3. Wait a minute until the computer is started.
4. Press the Motors reset button (orange).
5. (PS4) Power on the Gamepad pressing the Start button. The front led will flash until the connection is established with the Bluetooth receiver, and then it should remain on (blue color). If the controller can not reach this state, check the computer and the Bluetooth receiver and restart the robot. Pressing the "Deadman" button you should be able to move the robot.
6. If the led 1 blinks while the others are off you need to charge the controller. You can connect it to any USB port with the provided cable.

NOTE: If the Bluetooth connection is lost, the robot will detect this situation and will STOP for safety.

The operation of the RB-1 BASE is show in the next pictures.



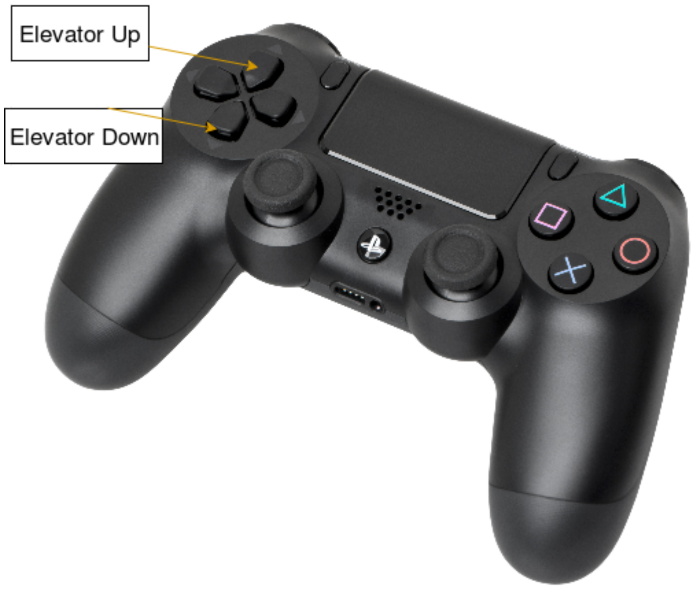


Figure 4 - PS4 Base Pad front view

Left and right joysticks control the linear (forward/backward movement) and the angular (rotational) speeds.

Triangle and Cross buttons control the speed of the base, which by default is set to 10% of the maximum speed.

5.3.1 Pairing the PS4 Dualshock controller with the Bluetooth device.

If you have received one Dualshock together with your robot, it should be already paired and you don't need to do this process.

There can be only one PS4 joystick associated with the robot's Bluetooth device. If you want to associate another PS4 controller or the usual one has lost its association you have to pair it again with the following procedure:

1. Shutdown the robot CPU (for example, by pressing once the CPU POWER BUTTON (blue)).
2. Plug the usb cable to the PS4 joystick and to the robot USB port.
3. Start the robot CPU by pressing the CPU POWER BUTTON (blue). During the boot up process the PS4 pad and the robot will be automatically paired.
4. Unplug the USB cable from the joystick and robot. Wait a minute until the boot up process is finished.
5. Power on the Gamepad by pressing the Start button. Now you are ready to move the robot.

You also can manage the Bluetooth devices from the Bluetooth settings of Ubuntu.

1. Remove all the devices pressing "-" button.
2. Press "+" button to add a new device and start the search.
3. In the joystick hold the shared button and the start button together until the light became blue.

5.3.2 How to charge the Gamepad battery

If you see that all the Gamepad leds are powered off, probably you will need to charge the gamepad battery. You can connect it to any USB port with the provided cable (with the CPU powered on).

6. Remote PC

6.1 Software installation and PC configuration

Some metapackages of the software have to be present in the remote computer in order to operate the robot remotely. In order to test the different components, the most useful tools are rviz (ROS Visualization tool) and rqt. In order to use these **in a remote machine**, we will need to:

1. Install ROS. To do that, follow the instructions at: http://wiki.ros.org/ROS/Installation. If your are not familiar with ROS, a good starting point is the Tutorials section of the official ROS Documentation: http://wiki.ros.org/ROS/Tutorials

2. Create a `catkin_workspace`, following the instructions at: http://www.ros.org/wiki/catkin/Tutorials/create_a_workspace

3. Install `rb1_base_common, robotnik_base_hw` and `rb1_base_sim` stacks in the remote computer (see github repositories). The third is not necessary, but is needed to simulate the robot.

4. Compile the stacks.

The recommended procedure to install the RB-1 BASE packages in a remote PC is as follows:

1. Download the sources to a local folder called `sources` locate at the `$HOME` directory:

```
     user@remote:~$ mkdir ~/sources
     user@remote:~$ cd ~/sources
     user@remote:~/sources/$ git clone \
https://github.com/RobotnikAutomation/rb1_base_common
     user@remote:~/sources/$ git clone \
https://github.com/RobotnikAutomation/rb1_base_sim
     user@remote:~/sources/$ git clone \
https://github.com/RobotnikAutomation/robotnik_base_hw
```

`Be sure that the branch is set at “kinetic-multi-devel”.`

2. Create a catkin workspace at the `$HOME` directory:

```
     user@remote:~$ mkdir ~/catkin_ws/src -p
     user@remote:~$ cd ~/catkin_ws/src
     user@remote:~/catkin_ws/src$ catkin_init_workspace
```

3. Even though the workspace is empty (there are no packages in the 'src' folder, just a single CMakeLists.txt link) you can still "build" the workspace:

```
user@remote:~/catkin_ws/src$ cd ~/catkin_ws/
user@remote:~/catkin_ws/$ catkin_make
```

4. Create symbolic links in the workspace folder to the `rb1_base_common` and the `rb1_base_sim` folders at sources:

```
user@remote:~/catkin_ws/$ cd ~/catkin_ws/src
user@remote:~/catkin_ws/src/$ ln -sf
~/sources/rb1_base_common
user@remote:~/catkin_ws/src/$ ln -sf
~/sources/rb1_base_sim
user@remote:~/catkin_ws/src/$ ln -sf
~/sources/robotnik_base_hw
```

5. And rebuild the workspace:

```
user@remote:~/catkin_ws/src/$ cd ~/catkin_ws
user@remote:~/catkin_ws/$ catkin_make
```

Once ROS is installed in your machine, you will need to setup your network in order to connect to the robot. A detailed guide can be found at http://wiki.ros.org/ROS/NetworkSetup. To connect your computer to the RB-1 BASE, you must follow the next steps:

Add the following line to the `/etc/hosts` file in your computer:

```
user@remote:~$ sudo vim.tiny /etc/hosts

# add the following line
192.168.0.200   rb1
```

Start the RB-1 BASE robot, connect your computer to its wifi network, open a terminal and type:

```
user@remote:~$ export ROS_MASTER_URI=http://rb1:11311

user@remote:~$ rostopic list
```

And you'll see a list of ROS topics where ROS nodes are interchanging information.

- Add your computer hostname to the /etc/hosts file in the RB-1 BASE computer:

```
user@remote:~$ ssh rb1@rb1
```

```
rb1@rb1:~$ sudo vim.tiny /etc/hosts
# add your hostname and ip address there
```

If everything worked you should be able to see all the topics published by the robot and the services offered by the robot controller:

```
rb1@rb1:~$ rostopic list
…
rb1@rb1:~$ rosservice list
…
```

You can view the values published by the nodes with rostopic echo, e.g.:

```
user@remote:~$ rostopic echo /rb1_control/odom
```

At this stage you can have a first look at the robot with

```
user@remote:~$ rqt
```

and

```
user@remote:~$ rosrun rviz rviz
```

6.2 Component testing

Instructions about how to launch and test each of the components are documented in the README.md file of each component and can be accessed via gitHub.

6.3 Robot simulation

Instructions for the robot simulation can be found in the README.md file of the rb1_base_sim repository.

7. Scripts and Start Configuration (in the robot)

The bootup process usually does not need to be changed by the user, but it is explained to simplify customization.

The robot control programs are launched during the boot-up process. This is achieved by starting three terminals with autologin, each one of those runs a different process. Each terminal is configured by a file locate at:

```
> getty@ttyX
```

where X is a number between 1 and 3. `TTY1` is configured to login as `root` user, while `TTY2, TTY3` and `TTY4` are configured to login as `rb1` user.

Then, the program run by each terminal is defined by the `.bashrc` file located at the `$HOME` directory of each user:

```
> /root/.bashrc
```

```
> /home/rb1/.bashrc
```

7.1 /etc/init/ttyX.conf

Four terminals with autolog are initiated during boot

`> getty@tty1` autologs as `root` in `TTY1` terminal. Its content is:

```
[Service]
ExecStart=
ExecStart=-/sbin/agetty --autologin root --noclear %I 38400
linux
```

`> getty@tty[2-5]` autolog as `rb1` in `TTY[2-5]` terminals. Its content is:

```
[Service]
ExecStart=
ExecStart=-/sbin/agetty --autologin rb1 --noclear %I 38400
linux
```

7.2 .bashrc

This is a start script, which every user has in its home directory. It's content is executed each time the user logins on the system.

When the `root` user logs in the `TTY1` terminal, the PS3 Pad pairing tool is run. To do that, the following code has been added to the `/root/.bashrc` file:

```
    > /root/.bashrc

    #### BOOT ####
         echo "ROBOTNIK RB-1 BASE"
         Terminal=`tty`
         case $Terminal in
         "/dev/tty1") sleep 25;
         ds4drv;;
         esac
```

When the `rb1` user logs in the `TTY2` terminal, a roscore instance is run. When the `rb1` user logs in the `TTY3` terminal, the `rb1_base_complete.launch` file from the `rb1_bringup` package is run, which brings up all the components of the robot. To do that, the following code has been added to the `/home/rb1/.bashrc` file:

```
    > /home/rb1/.bashrc

    #### BOOT ####
        echo "ROBOTNIK RB-1 BASE"
        Terminal=`tty`
         case $Terminal in
         "/dev/tty2") roscore;;
          "/dev/tty3") sleep 15;
          roslaunch rb1_base_bringup
rb1_base_complete.launch;;
"/dev/tty4") sleep 15;
cd /home/rb1/catkin_ws/src/rb1_robot/rb1_base_web;
        python -m SimpleHTTPServer;;
         esac
```

Note that the sleep value has been adjusted. If the server starts before the dynamic devices have been recognized, the server will exit with fault and the user will need to connect to the robot to start the server manually.