Tutorial 2

TurtleBot2

Ralph SEULIN

Le2i UMR CNRS 6306 – http://le2i.cnrs.fr
ralph.seulin@u-bourgogne.fr

v1.3 - January 2016

1- Introduction

This tutorial is mainly inspired from the TurtleBot2 manual provided by Robotnik Company.

Special thanks to Romain NAVARRO.

1.1- References

- Yujin Robot - http://kobuki.yujinrobot.com
- Robotnik - http://www.robotnik.eu/mobile-robots/turtlebot-ros/
- ROS Wiki
  - Kobuki Base - http://wiki.ros.org/kobuki
  - TurtleBot - http://wiki.ros.org/Robots/TurtleBot

2- Robot description

Kobuki Turtlebot 2 is a low-cost mobile research base designed for education and research on state of art robotics. With continuous operation in mind, Kobuki provides power supplies for an external computer as well as additional sensors and actuators. Its highly accurate odometry, amended by our factory calibrated gyroscope, enables precise navigation.

This robot is a compatible replacement for the iRobot Create. Although most of the accessories are fully compatible, be careful when plugging any device from the iRobot version to Kobuki version, because the connectors pin-out could be different.

2.1- Functional Specification

- Maximum translational velocity: 65 cm/s
- Maximum rotational velocity: 3.14 rad/s
- Payload: 5 kg (hard floor), 4 kg (carpet)
- Cliff: will not drive off a cliff with a depth greater than 5 cm
- Threshold Climbing: climbs thresholds of 12 mm or lower
- Rug Climbing: climbs rugs of 12 mm or lower
- Expected Operating Time: 3/7 hours (small/large battery)
- Expected Charging Time: 1.5/2.6 hours (small/large battery)
- Docking: can perform docking within a 2mx5m area in front of the docking station

2.2- Hardware Specification

Details available here: http://kobuki.yujinrobot.com/wiki/online-user-guide/

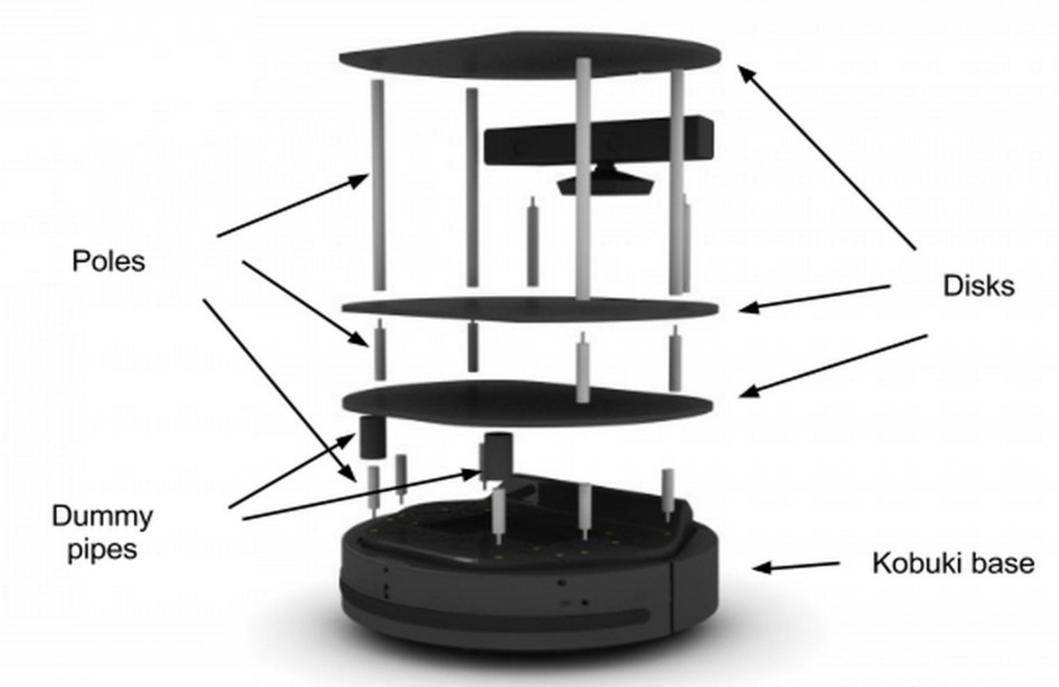
- PC Connection: usb or via RX/TX pins on the parallel port
- Motor Overload Detection : disables power to motors on detecting high current
- Odometry: 25718.16 ticks/revolution, 11.7 ticks/mm
- Gyro: factory calibrated, 1 axis (100 deg/s)
- Bumpers: left, center, right
- Cliff sensors: left, center, right
- Wheel drop sensor: left, right
- Power connectors: 5V/1A, 12V/1.5A, 12V/5A
- Docking recharging connector: 19V/2.1A- Expansion pins: 3.3V/1A, 5V/1A, 4 x analog in, 4 x digital in, 4 x digital out
- Audio : several programmable beep sequences
- Programmable LED: 2 x two-coloured LED
- State LED: 1 x two coloured LED [blinking - charging, Green - high level, Orange - low level]
- Buttons: 3 x touch buttons
- Battery: 14.8 V lithium-Ion 2200 mAh (small) 4400 mAh (large)
- Firmware upgradeable: via usb
- Sensor Data Rate: 50Hz
- Recharging Adapter: Input: 100-240V AC, 50/60Hz, 1.5A max; Output: 19V DC, 3.16A
- Netbook recharging connector (only enabled when robot is recharging): 19V/2.1A DC
- Docking IR Receiver: left, centre, right

2.3- Software Specification

- Kobuki drivers for ROS and non-ROS in C++
- Kobuki and Turtlebot simulator (Gazebo)
- Kobuki and Turtlebot apps

2.4- Components list

Basic components



TurtleBot2 is composed of a Kobuki mobile base + hardware mounted over the base:

- 16 Poles
- 3 Disks
- 2 Dummy pipes

Battery

- The 4S1P battery 2200 mAh. It provides 3 hours of operational time.

Kinect cable

- Cable to power the Kinect sensor directly from the robot’s battery.

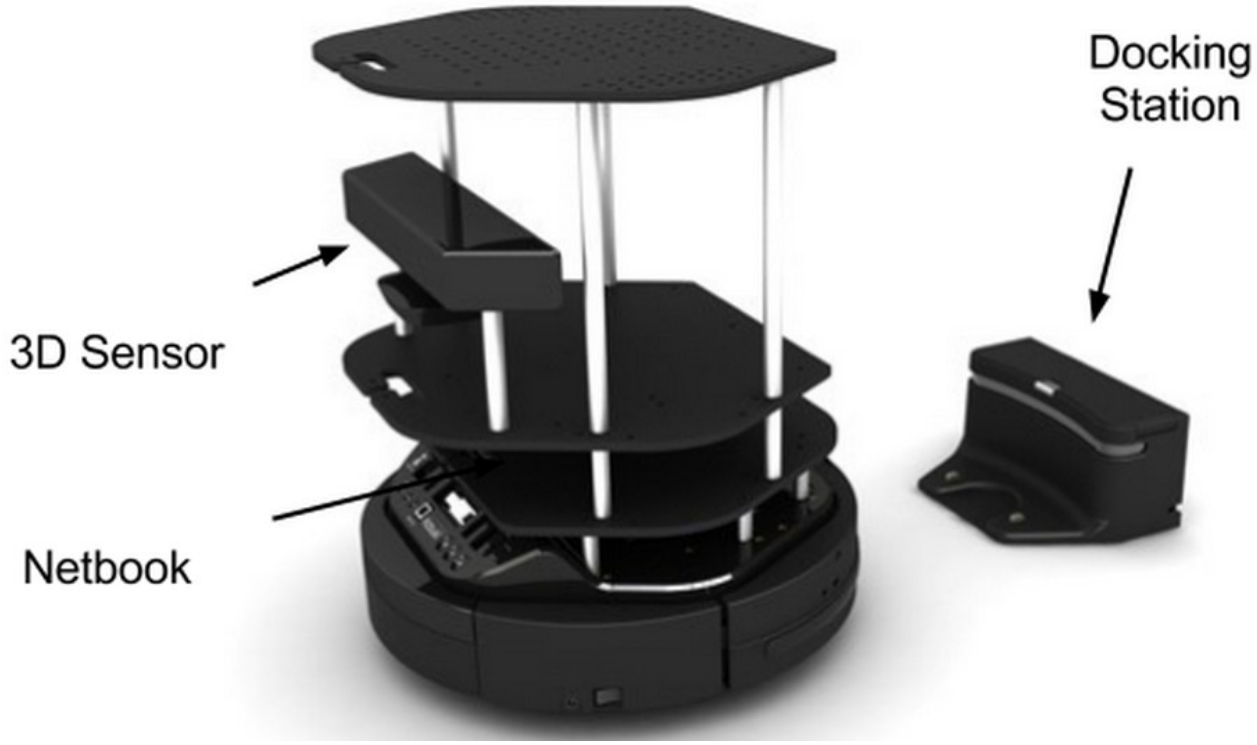
Usb communication cable

- Cable to control the Kobuki base from a PC.

Recharging adapter

- Standard adapter to charge the battery manually.

Additional components



3D sensor

- The Turtlebot 2 is intended to use a 3D sensor in order to navigate autonomously and map. By default, the Kinect sensor is provided with the robot, but any other sensor can be attached.

Docking station

- Battery charger intended to carry out an autonomous battery charge.

Large battery

- The 4S2P battery 2200 mAh. It provides 7 hours of operational time.

Netbook

- ASUS F201E – KX066DU
  - Intel Celeron 847 (1.1 GHz - 2 Mo)
  - RAM 4 Go - PC3-10600 - DDR3 1333 MHz
  - HDD 500 Go - 5400 RPM – SATA
  - 1 x RJ45, 2 x USB 2.0, 1 x USB 3.0

Turtlebot’s USB stick

- USB stick that contains all the manuals, and a recovery image of the provided netbook.

3- First steps with Kobuki Base

In this section will be described the procedure to start and test the main parts of the robot.

3.1- Launching Kobuki & Keyboard teleoperation

First, start minimal.launch to bring up Kobuki's basic software (bootstrap layer). (Optional - We use the argument --screen to get verbose output in the terminal).

```
$> roslaunch kobuki_node minimal.launch --screen
```

This launch file starts a nodelet manager and loads the Kobuki nodelet (the ROS wrapper around Kobuki's driver).

This launch file starts the control of the robot using the keyboard keys:

```
$> roslaunch kobuki_keyop keyop.launch
```

3.2- Kobuki base tf

First, start this launch file to bring up Kobuki's base with published tf.

```
$> roslaunch kobuki_node robot_with_tf.launch
```

Start rviz with robot model:

```
$> roslaunch kobuki_rviz_launchers view_robot.launch
```

This launch file starts a node that publish the TF transformation using the robot’s *.urdf file.

Start the control of the robot using the keyboard keys:

```
$> roslaunch kobuki_keyop keyop.launch
```

Make the robot move and observe in rviz the published tf

3.3- Testing the hardware

The kobuki_testsuite package provides a bunch of scripts to thoroughly test specific kobuki hardware components.

(if kobuki testsuite is not available, do the following:

```
$ sudo apt-get update && sudo apt-get install ros-hydro-kobuki-*)
```

Testing events

```
$> rosrun kobuki_testsuite test_events.py
```

Testing battery voltage

```
$> rosrun kobuki_testsuite test_battery_voltage.py
```

Testing the gyro

```
$> rosrun kobuki_testsuite test_gyro.py
```

Testing the LEDs

```
$> rosrun kobuki_testsuite test_led_array.py
```

Testing sounds

```
$> rosrun kobuki_testsuite test_sounds.py
```

4- Configuration for TurtleBot2 management from a Workstation

Managing a TurtleBot2 from a Workstation can be very convenient for development (you do not have to be sitted on the floor all day to work on the netbook :-) ...).

A specific setting of ROS has to be configured to enable this comfortable and powerful setup.

4.1- To Do

- Study tutorial on ROS to run on multiple machines:

  http://wiki.ros.org/ROS/Tutorials/MultipleMachines
- Modify the ~/.bashrc file of the Workstation to get the ROSMASTER working on the TurtleBot2 Netbook and to be able to launch some ROS commands from the Workstation
- To test your setup do the following tutorial

  !!! SKIP Chapter 3. TurtleBot Setup & Chapter 4. Workstation Setup !!!
  - http://wiki.ros.org/turtlebot/Tutorials/hydro/Network%20Configuration

5- TurtleBot2 BringUp

You are now ready to start managing your robot from your workstation.

- To start, do the following tutorial:

  http://wiki.ros.org/turtlebot_bringup/Tutorials/hydro/TurtleBot%20Bringup