

b3b33lar: Turtlebot

Libor Wagner

2019

◀ ▶ 🔍 ↺ ↻ ⌂

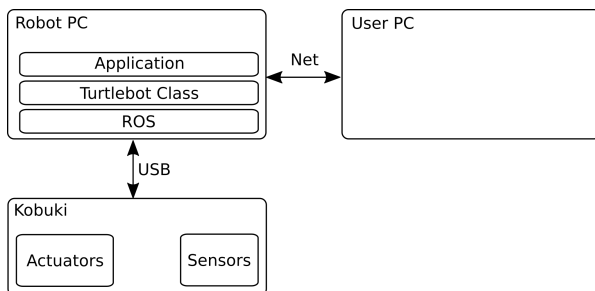
Turtlebot 2

- ▶ Kobuki base
 - ▶ Control
 - ▶ Odometry
 - ▶ Bumper
 - ▶ ...
- ▶ NUC PC
 - ▶ SSH
 - ▶ Wifi
 - ▶ ROS
- ▶ RGBD Sensor
 - ▶ Intel RealSense
 - ▶ Orbx Astra



◀ ▶ 🔍 ↺ ↻ ⌂

System overview



◀ ▶ 🔍 ↺ ↻ ⌂

Robot Operating System (ROS)

- ▶ Middleware that integrates, sensors, robots and logic into modular system.
- ▶ In barebones it provides communication layer between processing units.
- ▶ Supports multiple language and multiple machines.
- ▶ The main building blocks are Nodes, Topics and Services.
- ▶ Node - building block of robotic system. (camera driver, robot controller, image filter ...)
- ▶ Topic - named stream of data with same type. (rgb camera image, odometry, robot cmd ...)
- ▶ Service - named function, with specific request and response. (reset odometry, open gripper, compute ik ...)

◀ ▶ 🔍 ↺ ↻ ⌂

Turtlebot Python Class

- ▶ `cmd_velocity(linear=0, angular=0)` -> None
commands linear and angular velocity to the robot, this command has to be called repeatedly to ensure that the robot is moving.
- ▶ `get_odometry()` -> [x,y,a]
get current position, estimated from the encoders and gyroscope.
- ▶ `reset_odometry()` -> None
sets current position as an origin.

◀ ▶ 🔍 ↺ ↻ ⌂

Turtlebot Python Class continue

- ▶ `get_rgb_image()` -> image
gets RGB image from the RGBD camera.
- ▶ `get_depth_image()` -> image
gets depth image from RGBD camera.
- ▶ `get_point_cloud()` -> point_cloud
gets pointcloud from RGBD camera.
- ▶ `get_rgb_K(self)` -> K
gets calibration matrix K for RGB camera.
- ▶ `get_depth_K(self)` -> K
gets calibration matrix K for Depth camera.

◀ ▶ 🔍 ↺ ↻ ⌂

Turtlebot Python Class continue

- ▶ `register_button_event_cb(fun) -> None`
register button event callback.
- ▶ `register_bumper_event_cb(fun) -> None`
register bumper event callback.
- ▶ `play_sound(sound_id=0) -> None`
plays one of the predefined sounds.

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

Examples: What does it do?

```
1 from turtlebot import Turtlebot, Rate, get_time
2
3 turtle = Turtlebot()
4 rate = Rate(10)
5
6 t = get_time()
7
8 while get_time() - t < 10:
9     turtle.cmd_velocity(linear=0.1)
10    rate.sleep()
```

Connect to robots

- ▶ Network in the e210 ssid: e210bot, key: j6UsAC8a
- ▶ Using ssh: `ssh ros@turtle01` pass: r0sr0s
- ▶ Start robot driver: `turtle_start`
- ▶ Examples are in `./examples/`
- ▶ To start one: `python example_move_1m.py`

A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

Resources

- ▶ Turtlebot
 - ▶ <https://gitlab.fel.cvut.cz/wagnelib/turtlebot>
 - ▶ <http://www.turtlebot.com/turtlebot2/>
 - ▶ <http://wiki.ros.org/Robots/TurtleBot>
 - ▶ <http://wiki.ros.org/kobuki>
- ▶ Python
 - ▶ <https://www.python.org>
 - ▶ <https://docs.python.org/2.7/>
- ▶ ROS
 - ▶ <http://www.ros.org>
 - ▶ <http://wiki.ros.org>
 - ▶ <https://answers.ros.org>