

ROS, CZYLI O ROBOTACH SŁÓW KILKA

Marek Gawryszewski

26-06-2019

THE PROBLEM

ROBOT SOFTWARE CONSTRUCTION

- buildsystem
- algorithm(s) implementation
- logging (online/offline)
- teleoperation
- simulation (optional?)
- interfacing with hardware
- ...?

ROBOT SOFTWARE CONSTRUCTION

Business problem here is only one:

algorithm(s) implementation

The rest is infrastructure.

ROS: WHAT AND WHY

WHAT ROS PROVIDES

- Communication infrastructure
- Message passing
- RPC
- Robot specific features (i.e. pose estimation, mapping, localization)

EXPECTATIONS VS ROS

- buildsystem
 - algorithm(s)
implementation
 - logging (online/offline)
 - teleoperation
 - simulation (optional?)
 - interfacing with
hardware
 - ...?
- catkin
 - existing algorithms
 - rosbag
 - gazebo, v-rep, webots
 - existing drivers

DEFINITIONS

Nodes: A node is an executable that uses ROS to communicate with other nodes.

Messages: ROS data type used when subscribing or publishing to a topic.

Topics: Nodes can publish messages to a topic as well as subscribe to a topic to receive messages.

Master: Name service for ROS (i.e. helps nodes find each other)

roscore: Master + rosout + parameter server

[source](#)

DEMO 1

Turtlesim - LOGO in ROS.

ROSCORE

```
roscore
```

```
roslaunch turtlesim turtlesim_node
```

NODE

```
rosnode list
```

```
rosnode info /turtlesim
```

SERVICE

```
rosservice list
```

TOPIC

```
rostopic list  
rostopic list -v  
rostopic info /turtle1/cmd_vel  
rostopic type /turtle1/cmd_vel  
rostopic echo /turtle1/pose
```

TYPE

```
rosmmsg show geometry_msgs/Twist
```

MORE

```
rostopic type /turtle1/cmd_vel | rosmmsg show
```


LET'S MOVE OUR TURTLE

```
rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0,  
0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

WHERE IS THE TURTLE?

```
rostopic echo /turtle1/pose
```

RESET TURTLE POSITION

```
rosservice call /reset
```

Turtle require message stream with 1Hz.

```
rostopic pub -r 0.5 /turtle1/cmd_vel geometry_msgs/Twist --  
  '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'  
rostopic pub -r 10 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0,  
  0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

RQT

rqt

KEYBOARD CONTROL

Another node to control turtle

```
roslaunch turtlesim turtle_teleop_key
```

SEE WHAT HAPPENED!

```
roscpp list
roscpp info /turtlesim_node
```

LET'S ADD SOME LOGIC!

```
#workspace  
mkdir demo  
cd demo  
mkdir src  
catkin_make  
## package demo  
cd src  
catkin_create_pkg demo roscpp rospy turtlesim
```


ADJUSTMENTS

- CMakeLists.txt
- package.xml

CODE

```
src/demo_node.cpp
```

RUN

```
roslaunch demo demo_node
```

```
source demo/devel/setup.zsh  
catkin_make
```

DEMO 2

ARDUINO

Arduino is controlled with ROS node.

INSTALLATION

- get Arduino IDE from arduino.cc
- install roserial

```
sudo apt install ros-melodic-roserial_arduino
```

```
roslaunch roserial_arduino make_libraries.py .
```

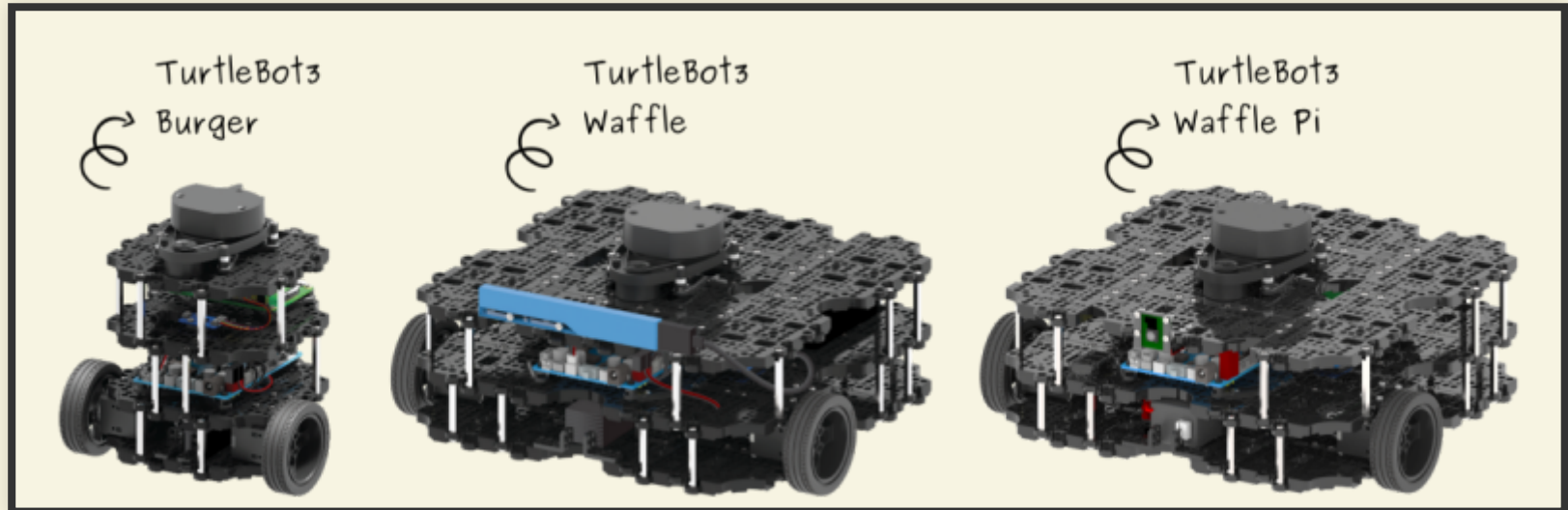
```
sudo chmod a+rw /dev/ttyUSB0
roslaunch roserial_python serial_node.py /dev/ttyUSB0
rostopic list
rostopic pub message std_msgs/String "Kolejny tekst" --once

rostopic echo /message_size
```

roserial

DEMO 3

TURTLEBOT



Turtlebot is simulated in Gazebo controlled with ROS
node

Turtlebot3

TURTLEBOT

Teleoperation in default world

```
export TURTLEBOT3_MODEL=waffle_pi  
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

```
export TURTLEBOT3_MODEL=waffle_pi  
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

TURTLEBOT

- More tutorials
- Simulation
- Teleoperation