Programowanie obrabiarek CNC i robotów przemysłowych

Wprowadzenie do systemu ROS

Paweł Strączyński

p.straczynski@tu.kielce.pl

Katedra Urządzeń Elektrycznych i Automatyki

Politechnika Świętokrzyska

Wprowadzenie

ROS (*Robot Operating System*) to platforma programistyczna do tworzenia oprogramowania oraz sterowania robotów. System ROS składa się ze zbioru bibliotek które pozwalają na sterowanie oraz symulację pracy robota – w tym także robota przemysłowego.

ROS dostarczany jest w ramach wolnej licencji BSD.

System ROS oferuje szereg narzędzi służących do:

- tworzenia paczek oraz budowania węzłów ROS,
- wymiany informacji pomiędzy węzłami ROS,
- podglądu i monitorowania sieci.

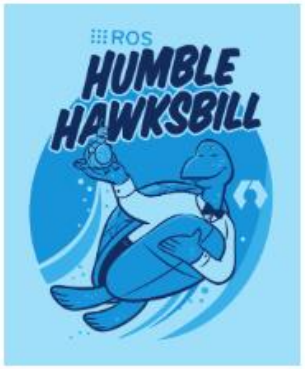
ROS opiera się na modelu mulitagentowym. Wspiera m.in. takie języki programowania jak C++, Python, Java

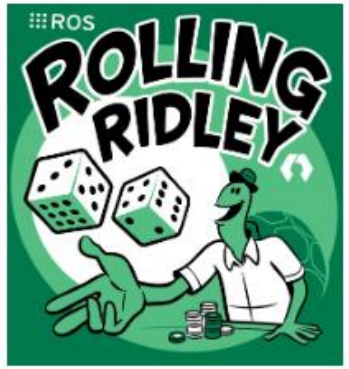
Dystrybucje ROS – ROS 1 / ROS 2

Active ROS 2 distributions

Recommended

Development

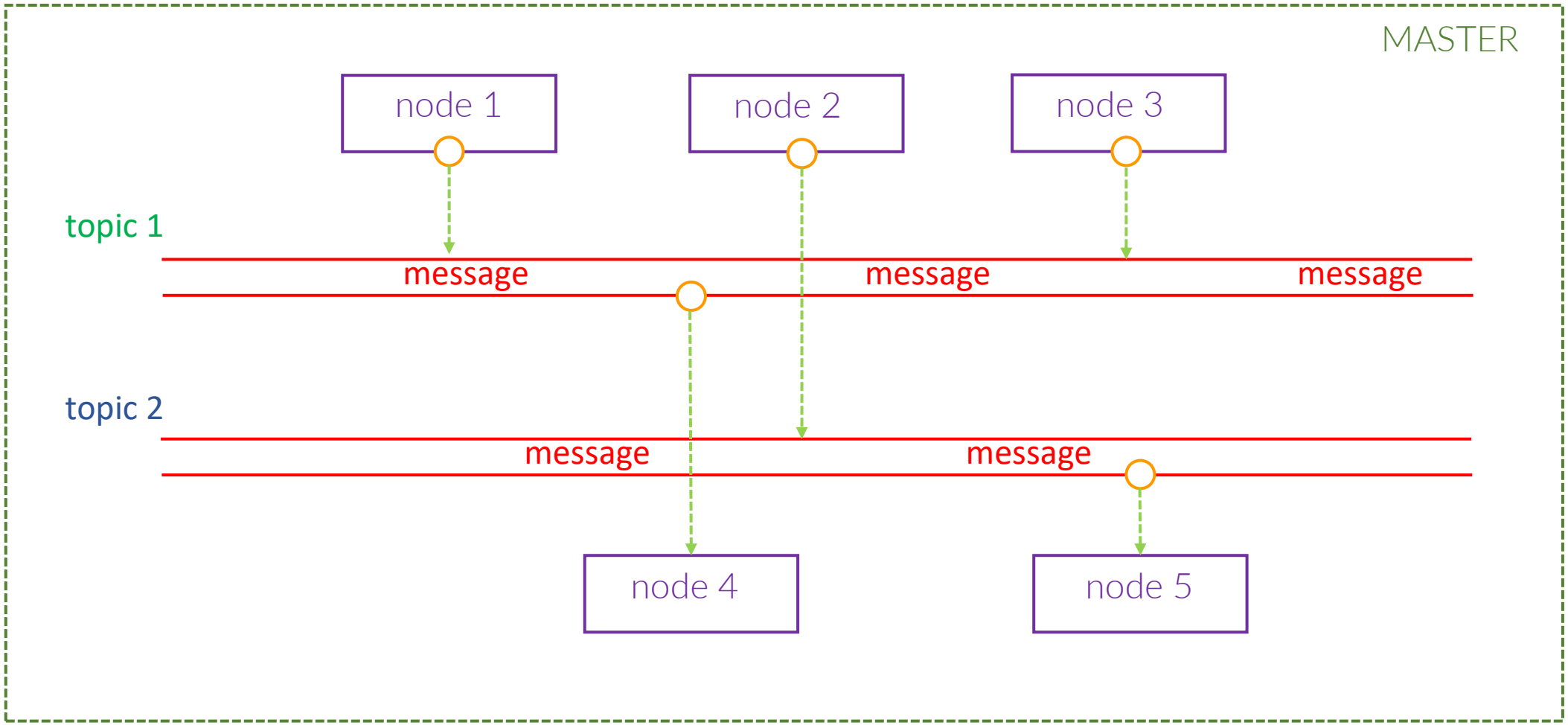




Active ROS 1 distributions

Recommended

Architektura ROS



Instalacja ROS

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.
d/ros-latest.list'
```

```
sudo apt install curl # if you haven't already installed curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

```
sudo apt update
```

```
sudo apt install ros-noetic-desktop-full
```

```
source /opt/ros/noetic/setup.bash
```

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Pakiety ROS (rospack)

Oprogramowanie systemu ROS stanowi zbiór pakietów. Pakiety to zbiory plików realizujący określone zadania.

Lista wszystkich dostępnych pakietów

```
rospack list
```

Wyszukiwanie katalogu z danym pakietem

```
rospack find package-name
```

Przeglądanie zawartości katalogu z pakietem

```
rosls package-name
```

Przejście do katalogu z danym pakietem

```
roscd package-name
```

Węzły ROS (rosnodes)

Węzły (*rosnodes*) to wykonywalne instancje programów systemu ROS.

Tworzenie węzła ROS

```
rosrun package-name executable-name
```

Wyświetlanie listy uruchomionych wezłów

```
rosnode list
```

Wyświetlanie informacji na temat węzła

```
rosnode info package-name
```

Zatrzymanie pracy węzła

```
rosnode kill package-name
```

Tematy i wiadomości ROS

Wiadomości służą do wymiany informacji w systemie ROS w oparciu o zasadę *publish-subscribe* z wykorzystaniem tematów (*rostopics*)

Lista aktywnych tematów ROS

```
rostopic list
```

Wyświetlanie wiadomości wysłanych w obrębie danego tematu

```
rosnode echo topic-name
```

Uzyskiwanie informacji o danych temacie

```
rostopic info topic-name
```

Publikowanie wiadomości

```
rostopic pub -r rate-in-hz topic-name message-type message-content
```

Publisher i subscriber w Pythonie

```python
#!/usr/bin/env python
# license removed for brevity
import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('chatter', String, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        hello_str = "hello world %s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```

```python
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + "I heard %s", data.data)

def listener():

    # In ROS, nodes are uniquely named. If two nodes with the same
    # name are launched, the previous one is kicked off. The
    # anonymous=True flag means that rospy will choose a unique
    # name for our 'listener' node so that multiple listeners can
    # run simultaneously.
    rospy.init_node('listener', anonymous=True)

    rospy.Subscriber("chatter", String, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == '__main__':
    listener()
```

Usługi w ROS (rosservice)

Usługi zapewniają komunikację dwukierunkową i realizują funkcjonalność zdalnego wykonywania funkcji.

Lista aktywnych usług ROS

```
rosservice list
```

Lista usług oferowanych przez dany węzeł

```
rosnode info node-name
```

Odczytywanie typu danych konkretnej usługi ORS

```
rosservice type service-name
```

Wywoływanie usługi

```
rosservice call service-name request-content
```