Robot Operating System – subskrybowanie i publikowanie danych

Dominik Belter[1]

Instytut Automatyki, Robotyki i Inżynierii Informatycznej
Politechnika Poznańska, Poznań, Poland

Poznań 8.11.2017

Budowanie pakietów – Catkin

- Polecenia wydajemy w workspace:

```
$ cd ~/catkin_ws
```

- Aktualizacja środowiska:

```
$ source devel/setup.bash
```

- Budowanie wszystkich pakietów:

```
$ catkin_make
```

Uruchomienie ROSa

- ROS Master:

```
$ roscore
```

ROS - kamera video


```
$ sudo apt-get install ros-kinetic-cv-bridge
$ sudo apt-get install ros-kinetic-cv-camera
```

ROS - kamera video


```
$ rosparam set cv_camera/device_id 0
$ rosrun cv_camera cv_camera_node
```

ROS - subskrybowanie

Utworzenie pakietu

```
$ cd src
$ catkin_create_pkg opencv_example std_msgs roscpp
```

ROS - subskrybowanie

Edycja CMakelists.txt
odkomentowanie:

```
add_executable(${PROJECT_NAME}_node src/opencv_example_node.cpp)
```

dodanie:

```
target_link_libraries(${PROJECT_NAME}_node ${catkin_LIBRARIES})

add_dependencies(${PROJECT_NAME}_node ${PROJECT_NAME}_generate_messages_cpp)
```

odkomentowanie i dodanie zależności od cv_bridge:

```
CATKIN_DEPENDS roscpp roscpp std_msgs message_runtime cv_bridge sensor_msgs
```

ROS - subskrybowanie

Edycja CMakelists.txt
ustawienie:

```
find_package(catkin REQUIRED COMPONENTS
roscpp
message_generation
std_msgs
cv_bridge
sensor_msgs
image_transport
)
```

ROS - subskrybowanie

Edycja package.xml
zależności od OpenCV:

```
<build_depend>cv_bridge</build_depend>
<exec_depend>cv_bridge</exec_depend>
<build_depend>sensor_msgs</build_depend>
<build_depend>image_transport</build_depend>
<exec_depend>sensor_msgs</exec_depend>
```

ROS - subskrybowanie

Edycja package.xml
odkomentowanie:

```
<build_export_depend>message_generation</build_export_depend>
<exec_depend>message_runtime</exec_depend>
<exec_depend>image_transport</exec_depend>
```

ROS - subskrybowanie

Utworzenie źródeł:

```
$ cd src/opencv_example/src/ $ touch opencv_example_node.cpp
```

ROS - subskrybowanie

opencv_example_node.cpp:

```
#include "ros/ros.h"

#include <iostream>

#include "cv_bridge/cv_bridge.h"

#include "sensor_msgs/Image.h"

#include <opencv2/opencv.hpp>

#include <image_transport/image_transport.h>


using namespace std;

using namespace cv;
```

ROS - subskrybowanie

opencv_example_node.cpp:

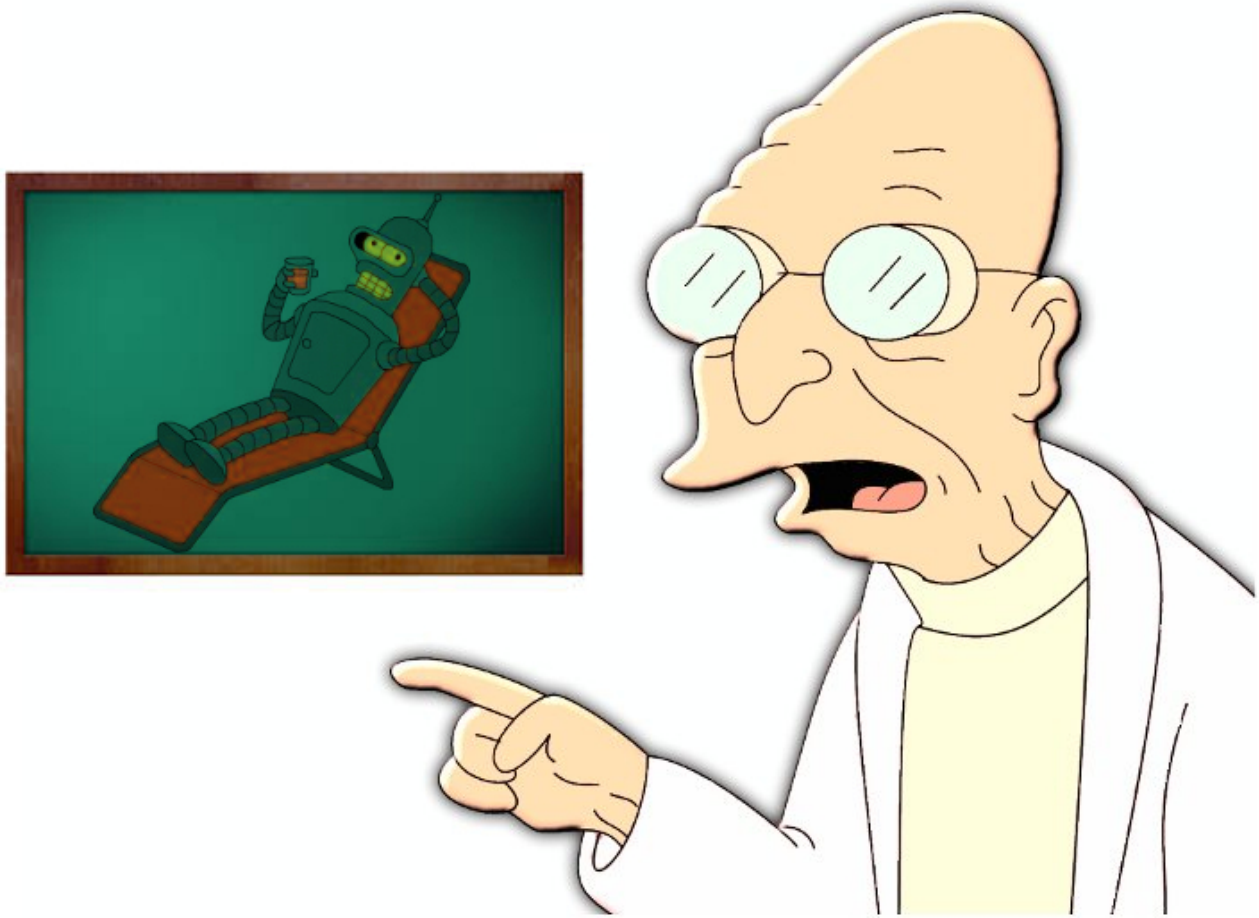
```
void imageCallback(const sensor_msgs::ImageConstPtr& msg){
  imshow("image", cv_bridge::toCvShare(msg, bgr8)->image);
  waitKey(30);
}


int main(int argc, char **argv) {

  //initialize node

  ros::init(argc, argv, "cv_example");

  // node handler

  ros::NodeHandle n;

  // subsribe topic

  ros::Subscriber sub = n.subscribe("/cv_camera/image_raw", 1000, imageCallback);

  //ros loop

  ros::spin();

  return 0;

}
```

ROS - subskrybowanie

Projekt do pobrania z Github:

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/dominikbelter/opencv_example
$ cd ..
$ catkin_make
```

Dziękuję za uwagę



lrm.put.poznan.pl
www.monoscience.com