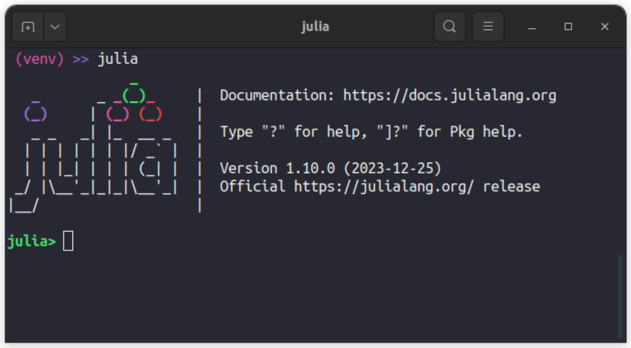
Lab #4: ROS2 using RCLPY in Julia

Abdelbacet Mhamdi
Senior-lecturer, Dept. of EE
ISET Bizerte — Tunisia
a-mhamdi

Aouini Khalil
Dept. of EE
ISET Bizerte — Tunisia
khalilaouini7

- In this lab you gonna find two part the first part is a execution of the programme and the second part is an explination of code lines
- You are required to carry out this lab using the REPL as in Figure 1.



```
(venv) >> julia
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.0 (2023-12-25)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia>
```

Figure 1: Julia REPL

I. First part :

in this part we execute the publisher/subscriber prograame and show you the result . so we gonna use ROS2 and julia to give them name first to and link them to a topic by using this two programme after we install ros2
We begin first of all by sourcing our ROS2 installation as follows:

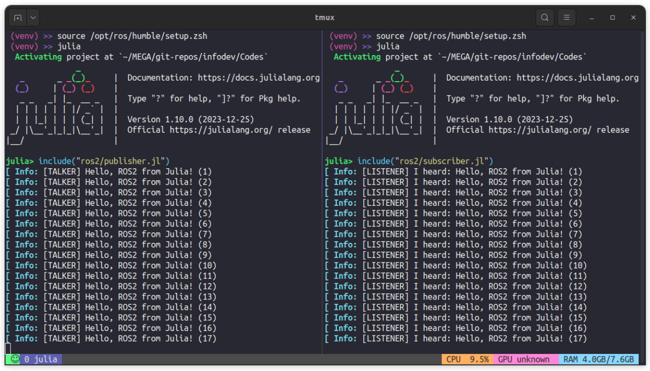
```
source /opt/ros/humble/setup.zsh
```

```julia
using PyCall
# Import the rclpy module from ROS2 Python
rclpy = pyimport("rclpy")
str = pyimport("std_msgs.msg")

# Initialize ROS2 runtime
rclpy.init()

# Create node
node = rclpy.create_node("my_publisher")
rclpy.spin_once(node, timeout_sec=1)

# Create a publisher, specify the message type and
the topic name
pub    =     node.create_publisher(str.String,
"infodev", 10)

# Publish the message `txt`
for i in range(1, 100)
    msg = str.String(data="Hello, ROS2 from Julia!
($(string(i)))")
     pub.publish(msg)
     txt = "[TALKER] " * msg.data
     @info txt
     sleep(1)
end

# Cleanup
rclpy.shutdown()
node.destroy_node()
```

```julia
using PyCall

rclpy = pyimport("rclpy")
str = pyimport("std_msgs.msg")

# Initialization
rclpy.init()

# Create node
node = rclpy.create_node("my_subscriber")

# Callback function to process received messages
function callback(msg)
    txt = "[LISTENER] I heard: " * msg.data
    @info txt
end

# Create a ROS2 subscription
sub    =    node.create_subscription(str.String,
"infodev", callback, 10)

while rclpy.ok()
      rclpy.spin_once(node)
end
# Cleanup
node.destroy_node()
rclpy.shutdown()
```

In a newly opened terminal, we need to start the publisher programme how start broadcasted a message first . second execute subscriber programme that listens to the messages being by our previous publisher

- **the result of the execution :**



```
tmux
(venv) >> source /opt/ros/humble/setup.zsh
(venv) >> julia
 Activating project at `~/MEGA/git-repos/infodev/Codes`

               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.0 (2023-12-25)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> include("ros2/publisher.jl")
[ Info: [TALKER] Hello, ROS2 from Julia! (1)
[ Info: [TALKER] Hello, ROS2 from Julia! (2)
[ Info: [TALKER] Hello, ROS2 from Julia! (3)
[ Info: [TALKER] Hello, ROS2 from Julia! (4)
[ Info: [TALKER] Hello, ROS2 from Julia! (5)
[ Info: [TALKER] Hello, ROS2 from Julia! (6)
[ Info: [TALKER] Hello, ROS2 from Julia! (7)
[ Info: [TALKER] Hello, ROS2 from Julia! (8)
[ Info: [TALKER] Hello, ROS2 from Julia! (9)
[ Info: [TALKER] Hello, ROS2 from Julia! (10)
[ Info: [TALKER] Hello, ROS2 from Julia! (11)
[ Info: [TALKER] Hello, ROS2 from Julia! (12)
[ Info: [TALKER] Hello, ROS2 from Julia! (13)
[ Info: [TALKER] Hello, ROS2 from Julia! (14)
[ Info: [TALKER] Hello, ROS2 from Julia! (15)
[ Info: [TALKER] Hello, ROS2 from Julia! (16)
[ Info: [TALKER] Hello, ROS2 from Julia! (17)
```

```
(venv) >> source /opt/ros/humble/setup.zsh
(venv) >> julia
 Activating project at `~/MEGA/git-repos/infodev/Codes`

               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.0 (2023-12-25)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> include("ros2/subscriber.jl")
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (1)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (2)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (3)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (4)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (5)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (6)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (7)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (8)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (9)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (10)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (11)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (12)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (13)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (14)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (15)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (16)
[ Info: [LISTENER] I heard: Hello, ROS2 from Julia! (17)

0 julia                                  CPU  9.5%  GPU unknown  RAM 4.0GB/7.6GB
```

Figure 2: Minimal publisher/subscriber in ROS2

II. Second part :

in this part we gonna explain the first programme function and the result showing up

- first step :

first of all we gonna lunch and initialition the subscriber / publisher programme :

- the publisher :

```
using PyCall
# Import the rclpy module from ROS2 Python
rclpy = pyimport("rclpy")
str = pyimport("std_msgs.msg")

# Initialize ROS2 runtime
rclpy.init()
```

- the subscriber :

```
using PyCall

rclpy = pyimport("rclpy")
str = pyimport("std_msgs.msg")

# Initialization
rclpy.init()
```

- second step

in this step we will create a node contain the two parts names

- publisher :

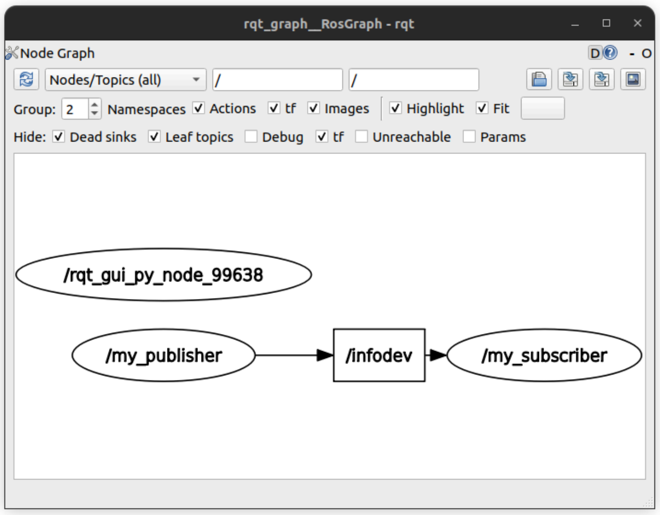
```
# Create node
node = rclpy.create_node("my_publisher")
rclpy.spin_once(node, timeout_sec=1)
```

- subscriber :

```
# Create node
node = rclpy.create_node("my_subscriber")
```

- thrid step :

in this step we gonna link up two node to the topic infodev like in fig 3



rqt_graph__RosGraph - rqt

Node Graph

Nodes/Topics (all) / /

Group: 2 Namespaces ✓ Actions ✓ tf ✓ Images ✓ Highlight ✓ Fit

Hide: ✓ Dead sinks ✓ Leaf topics ☐ Debug ✓ tf ☐ Unreachable ☐ Params

Figure 3: rqt_graph

- the publisher :

```
# Create a publisher, specify the message type and
the topic name
pub = node.create_publisher(str.String,"infodev",
10)
```

- the subscriber :

```
# Create a ROS2 subscription
sub = node.create_subscription(str.String,
"infodev", callback, 10)
```

- fourth step : in this step we gonna choose the message and how many time it will broadcasted

```
# Publish the message `txt`
for i in range(1, 100)
    msg = str.String(data="Hello, ROS2 from Julia!
($(string(i)))")
     pub.publish(msg)
     txt = "[TALKER] " * msg.data
     @info txt
     sleep(1)
end
```

```
# Callback function to process received messages
function callback(msg)
    txt = "[LISTENER] I heard: " * msg.data
    @info txt
end
```

-last step : we onna close up the programme and end the brodcast

```
# Cleanup
rclpy.shutdown()
node.destroy_node()
```

```
# Cleanup
node.destroy_node()
rclpy.shutdown()
```