

ROS2_day1 hw2 ROS2 python/C++ 통신 과제 보고서

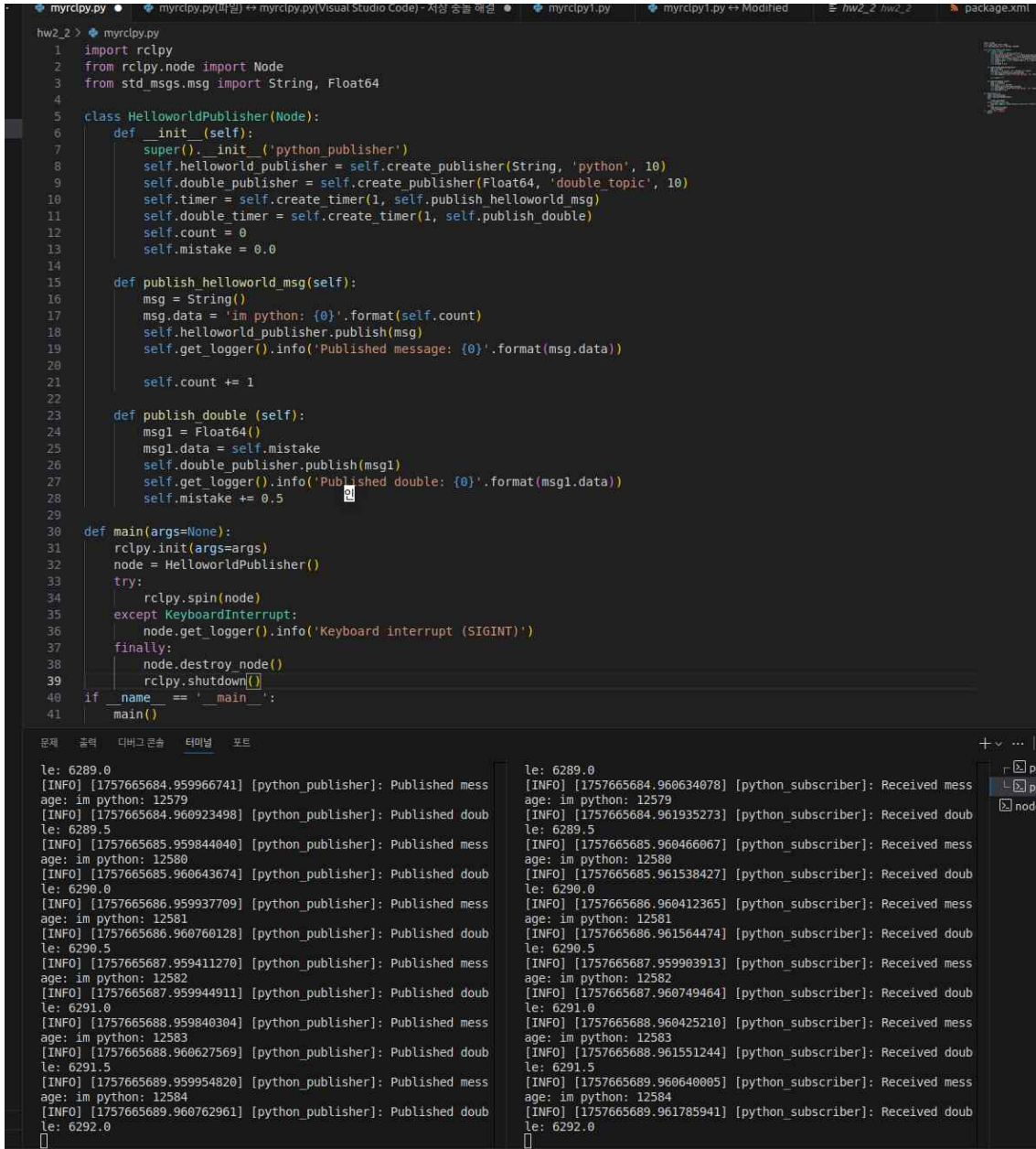
로봇 20기 인턴 현창석

목차

1. 알고리즘 설명

2. 시연 사진

1. 알고리즘 설명



```
hw2_2 > myrcipy.py
1 import rclpy
2 from rclpy.node import Node
3 from std_msgs.msg import String, Float64
4
5 class HelloworldPublisher(Node):
6     def __init__(self):
7         super().__init__('python_publisher')
8         self.helloworld_publisher = self.create_publisher(String, 'python', 10)
9         self.double_publisher = self.create_publisher(Float64, 'double_topic', 10)
10        self.timer = self.create_timer(1, self.publish_helloworld_msg)
11        self.double_timer = self.create_timer(1, self.publish_double)
12        self.count = 0
13        self.mistake = 0.0
14
15    def publish_helloworld_msg(self):
16        msg = String()
17        msg.data = 'im python: {}'.format(self.count)
18        self.helloworld_publisher.publish(msg)
19        self.get_logger().info('Published message: {}'.format(msg.data))
20
21        self.count += 1
22
23    def publish_double(self):
24        msg1 = Float64()
25        msg1.data = self.mistake
26        self.double_publisher.publish(msg1)
27        self.get_logger().info('Published double: {}'.format(msg1.data))
28        self.mistake += 0.5
29
30    def main(args=None):
31        rclpy.init(args=args)
32        node = HelloworldPublisher()
33        try:
34            rclpy.spin(node)
35        except KeyboardInterrupt:
36            node.get_logger().info('Keyboard interrupt (SIGINT)')
37        finally:
38            node.destroy_node()
39            rclpy.shutdown()
40
41 if __name__ == '__main__':
42     main()
```

Terminal Output:

```
le: 6289.0
[INFO] [1757665684.959966741] [python_publisher]: Published mess
age: im python: 12579
[INFO] [1757665684.960923498] [python_publisher]: Published doub
le: 6289.5
[INFO] [1757665685.959844040] [python_publisher]: Published mess
age: im python: 12580
[INFO] [1757665685.960643674] [python_publisher]: Published doub
le: 6290.0
[INFO] [1757665686.959937709] [python_publisher]: Published mess
age: im python: 12581
[INFO] [1757665686.960760128] [python_publisher]: Published doub
le: 6290.5
[INFO] [1757665687.959411270] [python_publisher]: Published mess
age: im python: 12582
[INFO] [1757665687.959944911] [python_publisher]: Published doub
le: 6291.0
[INFO] [1757665688.959840304] [python_publisher]: Published mess
age: im python: 12583
[INFO] [1757665688.960627569] [python_publisher]: Published doub
le: 6291.5
[INFO] [1757665689.959954820] [python_publisher]: Published mess
age: im python: 12584
[INFO] [1757665689.960762961] [python_publisher]: Published doub
le: 6292.0
[INFO] [1757665684.960634078] [python_subscriber]: Received mess
age: im python: 12579
[INFO] [1757665684.961935273] [python_subscriber]: Received doub
le: 6289.5
[INFO] [1757665685.960466067] [python_subscriber]: Received mess
age: im python: 12580
[INFO] [1757665685.961538427] [python_subscriber]: Received doub
le: 6290.0
[INFO] [1757665686.960412365] [python_subscriber]: Received mess
age: im python: 12581
[INFO] [1757665686.961564474] [python_subscriber]: Received doub
le: 6290.5
[INFO] [1757665687.959903913] [python_subscriber]: Received mess
age: im python: 12582
[INFO] [1757665687.960749464] [python_subscriber]: Received doub
le: 6291.0
[INFO] [1757665688.960425210] [python_subscriber]: Received mess
age: im python: 12583
[INFO] [1757665688.961551244] [python_subscriber]: Received doub
le: 6291.5
[INFO] [1757665689.960640005] [python_subscriber]: Received mess
age: im python: 12584
[INFO] [1757665689.961785941] [python_subscriber]: Received doub
le: 6292.0
```

이 사진은 published 기능을 구현한 파이썬 코드를 캡처한 사진이다.

먼저 통신하기 위한 rcl 모듈을 불러온다.
그 후 NODE를 상속하는 helloworldpublisher 클래스를 생성하였다.
생성자에서 퍼블리셔를 만드는데 각각 문자열과 정수형, 그리고 실수형이다. 퍼블리셔를 생성할 때 메시지 타입, 토픽 이름을

지정하고 큐 사이즈를 10으로 설정하였다..

그 다음 `create_timer`를 사용하여 두 개의 타이머를 설정하였다. 첫 번째 타이머는 1초마다 `publish_helloworld_msg` 함수를 호출하고, 두 번째 타이머는 1초마다 `publish_double` 함수를 호출한다. 이렇게 하면 같은 주기로 서로 다른 타입의 메시지가 통신된다.

`publish_helloworld_msg` 메서드는 `String` 타입 메시지를 생성하여 `count` 값과 문자열을 함께 `python` 토픽으로 발행한다. 메시지를 발행할 때마다 로그를 남기고, `count` 값을 1씩 증가시켜 다음 호출했을 때 계속 정수형 `count`가 발행되는지 알 수 있도록 하였다.

`publish_double` 메서드는 `Float64` 타입 메시지를 만들어 현재 `mistake` 값을 `double` 토픽으로 발행한다. 발행이 끝나면 로그를 남긴 후, `mistake` 값에 0.5를 더해 다음 호출 때 점점 커지도록 하여 계속 실수형 `mistake`가 발행되는지 알 수 있도록 하였다.

`main` 함수에서는 `rclpy.init()`으로 ROS 2를 초기화한 후 `HelloWorldPublisher` 객체를 생성한다. `rclpy.spin(node)`은 이 노드를 계속 실행시키는 코드이며 중간에 사용자가 `Ctrl+C`로 종료하면 `KeyboardInterrupt`로 인해 로그를 남기고 `rclpy.shutdown()`으로 노드와 ROS 2를 종료한다.

```
hw2_2 > myrcipy1.py
1 import rclpy
2 from rclpy.node import Node
3 from std_msgs.msg import String,Float64
4 class HelloworldSubscriber(Node):
5
6     def __init__(self):
7         super().__init__('python_subscriber')
8         self.helloworld_subscriber = self.create_subscription(
9             String,
10            'python',
11            self.subscribe_topic_message,
12            10
13        )
14
15        self.helloworld_subscriber1 = self.create_subscription(
16            Float64,
17            'double_topic',
18            self.subscribe_topic_double,
19            10
20        )
21
22    def subscribe_topic_message(self,msg):
23        self.get_logger().info('Received message: {0}'.format(msg.data))
24
25    def subscribe_topic_double(self,msg1):
26        self.get_logger().info('Received double: {0}'.format(msg1.data))
27
28    def main (args = None ):
29        rclpy.init(args=args)
30        node = HelloworldSubscriber()
31        try :
32            rclpy.spin (node)
33
34        except KeyboardInterrupt :
35            node.get_logger().info('Keyboard interrupt (SIGINT)')
36        finally :
37            node.destroy_node()
38            rclpy.shutdown()
39
40 if __name__ == '__main__':
41     main()
```

터미널

```
le: 6314.5
[INFO] [1757665735.959867665] [python_publisher]: Published mess
age: im python: 12630
[INFO] [1757665735.960666421] [python_publisher]: Published doub
le: 6315.0
[INFO] [1757665736.959960883] [python_publisher]: Published mess
age: im python: 12631
[INFO] [1757665736.960771922] [python_publisher]: Published doub
le: 6315.5
[INFO] [1757665737.959194844] [python_publisher]: Published mess
age: im python: 12632
[INFO] [1757665737.959568645] [python_publisher]: Published doub
le: 6316.0
[INFO] [1757665738.959831939] [python_publisher]: Published mess
age: im python: 12633
[INFO] [1757665738.960615222] [python_publisher]: Published doub
le: 6316.5
[INFO] [1757665739.959741880] [python_publisher]: Published mess
age: im python: 12634
[INFO] [1757665739.960650519] [python_publisher]: Published doub
le: 6317.0
[INFO] [1757665740.959783144] [python_publisher]: Published mess
age: im python: 12635
[INFO] [1757665740.960664524] [python_publisher]: Published doub
le: 6317.5

le: 6314.5
[INFO] [1757665735.960502551] [python_subscriber]: Received mess
age: im python: 12630
[INFO] [1757665735.961512803] [python_subscriber]: Received doub
le: 6315.0
[INFO] [1757665736.960501472] [python_subscriber]: Received mess
age: im python: 12631
[INFO] [1757665736.961514520] [python_subscriber]: Received doub
le: 6315.5
[INFO] [1757665737.959565679] [python_subscriber]: Received mess
age: im python: 12632
[INFO] [1757665737.960052912] [python_subscriber]: Received doub
le: 6316.0
[INFO] [1757665738.960386350] [python_subscriber]: Received mess
age: im python: 12633
[INFO] [1757665738.961386513] [python_subscriber]: Received doub
le: 6316.5
[INFO] [1757665739.960285155] [python_subscriber]: Received mess
age: im python: 12634
[INFO] [1757665739.961381015] [python_subscriber]: Received doub
le: 6317.0
[INFO] [1757665740.960280076] [python_subscriber]: Received mess
age: im python: 12635
[INFO] [1757665740.961274003] [python_subscriber]: Received doub
le: 6317.5
```

이 사진은 파이썬으로 구현한 구독자 노드를 캡처한 사진이다. , ROS 2의 기본 노드 클래스를 상속받는 클래스 HelloworldSubscribe를 선언하였다.

노드가 생성될 때 정수형, 문자열 메시지를 받는 구독자와 실수형을 받는 구독자가 생성된다. 첫 번째 구독자는 'python' 토픽에 연결된다. 이 구독자는 새로운 메시지를 받을 때마다 subscribe_topic_message라는 함수를 실행시킨다,

두 번째 구독자는 double_topic이라는 토픽에 연결된다. 이 구독자는 새로운 데이터가 수신될 때마다 subscribe_topic_double 함수를 실행시킨다,

main 함수에서는 ROS시스템을 초기화한 뒤 HelloWorldSubscriber 노드를 생성하고, rclpy.spin(node)을 호출하여 노드가 계속해서 실행되도록 하였다. 따라서 퍼블리셔가 메시지를 보낼 때마다 해당 메시지에 대한 로그가 자동으로 출력되도록 하였다.

```
src > my_cpp_node.cpp > topic_double(const std_msgs::msg::Float32::SharedPtr)
1 #include "rclcpp/rclcpp.hpp"
2 #include "std_msgs/msg/string.hpp"
3 #include <chrono>
4 #include <memory>
5 #include <functional>
6 #include "std_msgs/msg/float32.hpp"
7 class MyNode : public rclcpp::Node
8 {
9 public:
10 MyNode();
11
12 private:
13 rclcpp::Subscription<std_msgs::msg::String>::SharedPtr subscriber;
14 rclcpp::Subscription<std_msgs::msg::Float32>::SharedPtr subscriber_float;
15
16 void topic_callback(const std_msgs::msg::String::SharedPtr msg);
17 void topic_double(const std_msgs::msg::Float32::SharedPtr msg1);
18
19 };
20
21 MyNode::MyNode() : Node("mynode")
22 {
23
24 subscriber = this->create_subscription<std_msgs::msg::String>("topicname", 10, std::bind(&MyNode::topic_callback, this, std::placeholders::_1));
25
26 subscriber_float = this->create_subscription<std_msgs::msg::Float32>("topicdouble", 10, std::bind(&MyNode::topic_double, this, std::placeholders::_1));
27
28
29 }
30
31 void MyNode::topic_callback(const std_msgs::msg::String::SharedPtr msg) {
32
33 RCLCPP_INFO(this->get_logger(), "Subscribed to '%s'", msg->data.c_str());
34
35 }
36
37 void MyNode::topic_double(const std_msgs::msg::Float32::SharedPtr msg1){
38 RCLCPP_INFO(this->get_logger(), "Subscribed Double to '%.2f'", msg1->data);
39 }
40
41 int main(int argc, char ** argv)
42 {
43 rclcpp::init(argc, argv);
44 auto node = std::make_shared<MyNode>();
45 rclcpp::spin(node);
46 rclcpp::shutdown();
47 return 0;
48 }
```

MyNode 클래스는 기본 노드 클래스를 상속받는다. 이 클래스 안에도 파이썬으로 구현한 것과 같이 정수형, 실수형 그리고 문자열을 받는 구독자가 있다.

생성자에서는 create_subscription 함수를 이용해 두 구독자를 생성한다. "topicname"이라는 이름의 토픽은 String 타입 메시지를 구독하고 이 구독자는 새로운 메시지가 들어올 때마다 topic_callback 함수를 실행시킨다. "topicdouble"이라는 이름의 토픽은 Float32 타입 메시지를 구독한다. 이 구독자는 메시지를 받으면 topic_double 함수를 실행시킨다.

topic_double 함수는 실수 메시지를 받아, 소수점 둘째 자리까지

포매팅하여 로그에 출력한다.

```
src > my_cpp_node1.cpp > ...
1 #include "rclcpp/rclcpp.hpp"
2 #include "std_msgs/msg/string.hpp"
3 #include <chrono>
4 #include <memory>
5 #include <functional>
6 #include "std_msgs/msg/float32.hpp"
7
8
9 // class MyCppNode : public rclcpp::Node
10 // {
11 //     public:
12 //     MyCppNode() : Node("my_cpp_node")
13 //     {
14 //         RCLCPP_INFO(this->get_logger(), "Hello, ROS 2 C++ Node!");
15 //     }
16 // };
17
18 class MyNode : public rclcpp::Node
19 {
20     public:
21     MyNode();
22
23     private:
24     rclcpp::Publisher<std_msgs::msg::String>::SharedPtr publisher_;
25     rclcpp::Publisher<std_msgs::msg::Float32>::SharedPtr publisher_float_;
26     rclcpp::TimerBase::SharedPtr timer_;
27
28     int count = 0;
29     float count_ = 0; //이렇게 안하면 아님
30     void timer_callback();
31
32 };
33
34 MyNode::MyNode() : Node("mynode1")
35 {
36     publisher_ = this->create_publisher<std_msgs::msg::String>("topicname", 10);
37     publisher_float_ = this->create_publisher<std_msgs::msg::Float32>("topicdouble", 10);
38     timer_ = this->create_wall_timer(std::chrono::duration<double>(1.0), std::bind(&MyNode::timer_callback, this));
39
40 }
41
42 }
```

MyNode라는 클래스 안에는 타이머가 정의되어 있다. 타이머는 일정 주기에 맞추어 콜백 함수를 실행하는 역할을 한다.

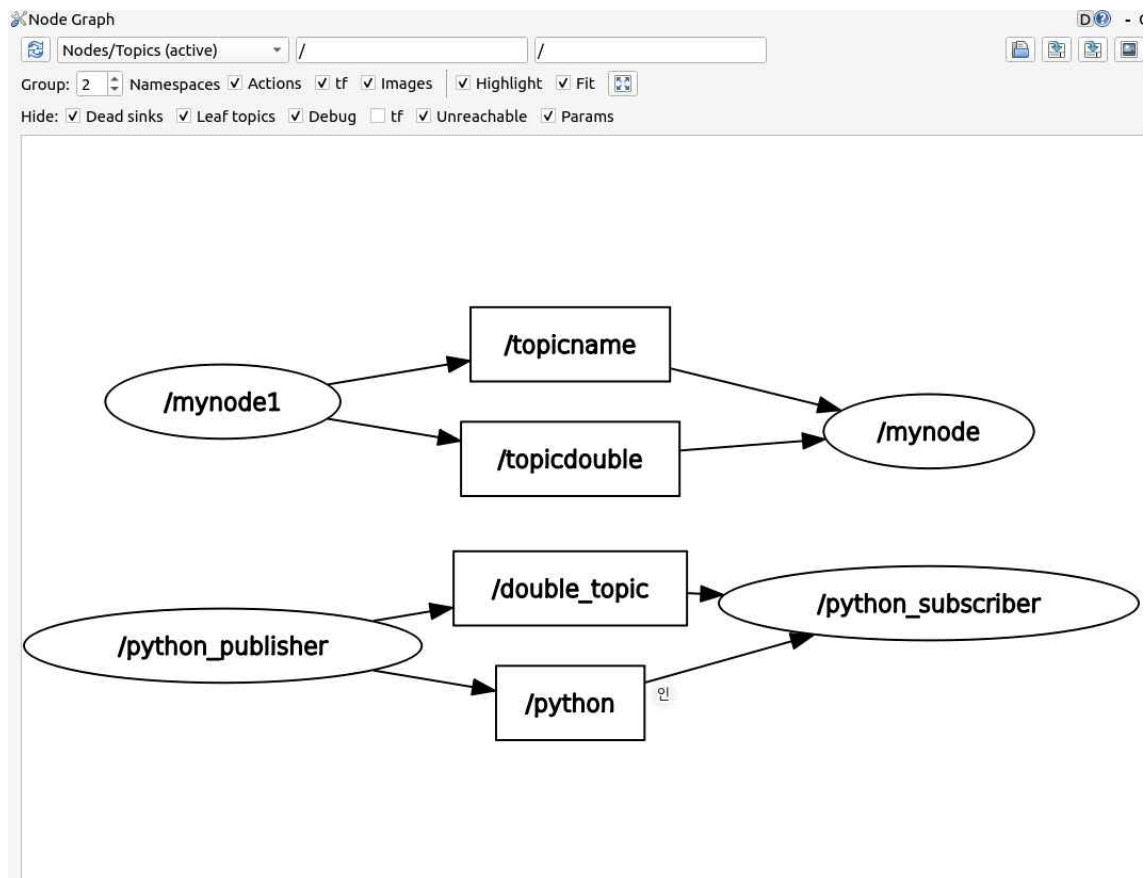
내부 카운트 값(count_)을 "i'm C++"이라는 메시지 뒤에 붙여준 후 로그로 출력한다.

다음으로 실수 메시지를 하나 생성한다. 별도의 변수(count_)를 0.5씩 증가시키고 누적한 이 값 역시 로그로 출력하고, publisher_float를 통해 발행한다. 결과적으로 이 노드는 매 초마다 문자열과 및 정수, 실수 메시지를 동시에 발행한다.

main 함수에서는 ROS 2 시스템을 초기화한 뒤 MyNode 객체를 생성하고, rclcpp::spin(node)으로 노드를 실행한다. 이 상태에서

타이머가 계속 동작한다.

시연사진



python

```
le: 3655.5
[INFO] [1757660417.959855819] [python_publisher]: Published message: im python: 7312
[INFO] [1757660417.960635109] [python_publisher]: Published double: 3656.0
[INFO] [1757660418.959600468] [python_publisher]: Published message: im python: 7313
[INFO] [1757660418.960295292] [python_publisher]: Published double: 3656.5
[INFO] [1757660419.959612800] [python_publisher]: Published message: im python: 7314
[INFO] [1757660419.960274548] [python_publisher]: Published double: 3657.0
[INFO] [1757660420.959921120] [python_publisher]: Published message: im python: 7315
[INFO] [1757660420.960771894] [python_publisher]: Published double: 3657.5
[INFO] [1757660421.959873059] [python_publisher]: Published message: im python: 7316
[INFO] [1757660421.960718412] [python_publisher]: Published double: 3658.0
[INFO] [1757660422.959789245] [python_publisher]: Published message: im python: 7317
[INFO] [1757660422.960468770] [python_publisher]: Published double: 3658.5
[]

le: 3655.5
[INFO] [1757660417.960410745] [python_subscriber]: Received message: im python: 7312
[INFO] [1757660417.961559987] [python_subscriber]: Received double: 3656.0
[INFO] [1757660418.960200218] [python_subscriber]: Received message: im python: 7313
[INFO] [1757660418.961135519] [python_subscriber]: Received double: 3656.5
[INFO] [1757660419.960061722] [python_subscriber]: Received message: im python: 7314
[INFO] [1757660419.960938631] [python_subscriber]: Received double: 3657.0
[INFO] [1757660420.960436028] [python_subscriber]: Received message: im python: 7315
[INFO] [1757660420.961434260] [python_subscriber]: Received double: 3657.5
[INFO] [1757660421.960399542] [python_subscriber]: Received message: im python: 7316
[INFO] [1757660421.961512558] [python_subscriber]: Received double: 3658.0
[INFO] [1757660422.960313045] [python_subscriber]: Received message: im python: 7317
[INFO] [1757660422.961200080] [python_subscriber]: Received double: 3658.5
[]
```

C++

```
1'
[INFO] [1757660195.723964677] [mynode1]: Published Double : '766.00'
[INFO] [1757660196.723936562] [mynode1]: Published message : 'i'm C++1532'
[INFO] [1757660196.724092952] [mynode1]: Published Double : '766.50'
[INFO] [1757660197.723931611] [mynode1]: Published message : 'i'm C++1533'
[INFO] [1757660197.724091211] [mynode1]: Published Double : '767.00'
[]

e]: Subscribed Double to '765.50'
[INFO] [1757660195.724116064] [mynode]: Subscribed to 'i'm C++1531'
[INFO] [1757660195.724226453] [mynode]: Subscribed Double to '766.00'
[INFO] [1757660196.724404227] [mynode]: Subscribed to 'i'm C++1532'
[INFO] [1757660196.724515681] [mynode]: Subscribed Double to '766.50'
[INFO] [1757660197.724263258] [mynode]: Subscribed to 'i'm C++1533'
[INFO] [1757660197.724385818] [mynode]: Subscribed Double to '767.00'
```

