

ROS DAY1

2번 과제

2023741024
로봇학부 박건후

목차

- (1) rclcpp의 hpp
- (2) rclcpp의 cpp
- (3) rclpy의 py파일

(1) rclcpp의 hpp 파일

```
#pragma once
#include <rclcpp/rclcpp.hpp>
#include <std_msgs/msg/string.hpp>
#include <std_msgs/msg/int32.hpp>
#include <std_msgs/msg/float32.hpp>
class MyCppNode : public rclcpp::Node{
public:
    MyCppNode();
private:
    rclcpp::TimerBase::SharedPtr timer_;
    void timer_callback();
    rclcpp::Publisher<std_msgs::msg::String>::SharedPtr mycpp_publisher_;
    rclcpp::Publisher<std_msgs::msg::Int32>::SharedPtr mycpp_publisher_int;
    rclcpp::Publisher<std_msgs::msg::Float32>::SharedPtr mycpp_publisher_float;
    int count_ = 1;
};

class subscribe_node : public rclcpp::Node{
    rclcpp::Subscription<std_msgs::msg::String>::SharedPtr mycpp_subscriber_;
    rclcpp::Subscription<std_msgs::msg::Int32>::SharedPtr mycpp_subscriber_int;
    rclcpp::Subscription<std_msgs::msg::Float32>::SharedPtr mycpp_subscriber_float;
    void topic_callback_str(const std_msgs::msg::String::SharedPtr msg);
    void topic_callback_int(const std_msgs::msg::Int32::SharedPtr msg);
    void topic_callback_float(const std_msgs::msg::Float32::SharedPtr msg);
public:
    subscribe_node();
};
```

위 사진은 스트링, 정수, 실수 3 종류의 데이터를 보내기 위해 3개의 추가적인 헤더파일을 선언한 것입니다. publish할 클래스로 MyCppNode를 클래스로 설정하였고, subscribe 역할을 할 subscribe_node를 클래스로 생성하였습니다. 서로 다른 종류의 메시지를 발행하기 위해 각각의 타입에 맞춰 발행을 하였습니다. 그리고 timer_callback()으로 타이머가 1초 후에 해당 함수를 호출하여 publish하도록 하였습니다.

그리고 subscribe_node 클래스로 각각에 대해 콜백을 받을 수 있도록 세개의 메서드를 생성하였습니다.

(2) rclcpp의 cpp

```
pkg_hw2.cpp > ...
#include "pkg_hw2.hpp"
#include <chrono>
using namespace std::chrono_literals;

MyCppNode:: MyCppNode():Node("publish_node"){
    RCLCPP_INFO(this->get_logger(), "Hello, ROS 2 C++ Node!");
    mycpp_publisher_ = this->create_publisher<std_msgs::msg::String>("topicname", 10);
    mycpp_publisher_int= this->create_publisher<std_msgs::msg::Int32>("int_topic", 10)
    mycpp_publisher_float= this->create_publisher<std_msgs::msg::Float32>("float_topic", 10);
    timer_ = this->create_wall_timer(1s, std::bind(&MyCppNode::timer_callback, this));
}

void MyCppNode:: timer_callback_str(){
    auto msg = std_msgs::msg::String();
    auto num = std_msgs::msg::Int32();
    auto fl= std_msgs::msg::Float32();
    msg.data= "Hello World: "+ std::to_string(count++);
    num.data = 3456;
    fl.data = 3.4567;
    RCLCPP_INFO(this->get_logger(), "Published message: '%s', int: '%d', float: '%f'", msg.data.c_str(), num.data, fl.data);

    mycpp_publisher_ -> publish(msg);
    mycpp_publisher_int->publish(num);
    mycpp_publisher_float->publish(fl);
}
```

cpp에서 콜백 메서드를 통해 해당 값을 각각 할당하고 각각 publish하는 부분입니다.

(3) rclpy의 python