hw3 보고서

20기 인턴 송수민

1. 메인 윈도우 파일

```cpp
#include "hw3_pkg/main_window.hpp"
#include "ui_MainWindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent), ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    // ROS2 초기화
    rclcpp::init(0, nullptr);
    node = std::make_shared<rclcpp::Node>("qt_talker_listener");

    // 퍼블리셔
    pub = node->create_publisher<std_msgs::msg::String>("chatter", 10);

    // 서브스크라이버
    sub = node->create_subscription<std_msgs::msg::String>(
        "chatter", 10,
        [this](std_msgs::msg::String::SharedPtr msg) {
            // GUI 스레드 안전하게 라벨 업데이트
            QString text = QString::fromStdString(msg->data);
            QMetaObject::invokeMethod(ui->label, "setText",
                                      Qt::QueuedConnection,
                                      Q_ARG(QString, text));
        });

    connect(ui->pushButton, &QPushButton::clicked, this, &MainWindow::onPublishButton);

    // ROS2 spin 별도 스레드
    ros_thread = std::thread([this]() { rosSpin(); });
}
MainWindow::~MainWindow()
{
    rclcpp::shutdown();
    if (ros_thread.joinable()) ros_thread.join();
    delete ui;
}
```

먼저 ui->setupUi(this);로 Qt UI를 초기화한다.

그다음 rclcpp::init(0, nullptr);를 통해 ROS2를 초기화한다.
"qt_talker_listener"라는 이름의 ROS2 노드를 생성한다.

퍼블리셔를 만든다. 토픽 이름은 "chatter", 큐 사이즈는 10이다.
서브스크라이버를 만든다. 토픽 이름은 "chatter", 큐 사이즈는 10이며, 메시지를 수신하면 람다 함수가 호출된다.

람다 안에서는 QString::fromStdString으로 메시지를 변환하고,
QMetaObject::invokeMethod를 사용하여 GUI 스레드에서도 안전하게 ui->label을 업데이트한다.

connect(...) 구문을 통해 버튼 클릭과 onPublishButton 슬롯을 연결한다.

별도의 스레드를 만들어, 이 스레드에서 ROS2 콜백을 계속 처리하도록 한다.

소멸자 MainWindow::~MainWindow()에서는 프로그램 종료 시 실행된다.
먼저 rclcpp::shutdown();으로 ROS2를 종료하고, ROS 스레드가 실행 중이면 join()으로 안전하게 합류한다.
마지막으로 delete ui;로 UI 객체를 해제한다.

```
void MainWindow::onPublishButton()
{
    auto msg = std_msgs::msg::String();
    msg.data = ui->lineEdit->text().toStdString();
    pub->publish(msg);
}

void MainWindow::rosSpin()
{
    rclcpp::Rate rate(10);
    while (rclcpp::ok()) {
        rclcpp::spin_some(node);
        rate.sleep();
    }
}
```

onPublishButton() 함수에서는 사용자가 ui->lineEdit에 입력한 문자열을 가져온다.
toStdString()으로 QString을 std::string으로 변환하고, 이를 std_msgs::msg::String 메시지에 저장한다.
그 후 퍼블리셔를 통해 메시지를 발행한다.

rosSpin() 함수에서는 별도 스레드에서 ROS2 콜백을 처리한다.
rclcpp::Rate rate(10);로 초당 10회 주기를 설정하고, 반복문 안에서
rclcpp::spin_some(node);를 호출하여 노드의 콜백을 처리한다.
반복이 끝나면 rate.sleep();으로 주기를 맞춘다.
이 과정을 ROS2가 종료될 때까지 계속 반복한다.