Explicación archivos ROS2_comienzo

robot_news_station.cpp

Este es un programa ROS2 en C++ que define un nodo llamado `RobotNewsStationNode`. Aquí está el desglose:

1. Inclusión de bibliotecas necesarias

```
#include "rclcpp/rclcpp.hpp"
#include "example_interfaces/msg/string.hpp"
```

Incluye las bibliotecas necesarias para usar ROS2 y los mensajes de tipo `String`.

2. Definición de la clase `RobotNewsStationNode`

```
class RobotNewsStationNode : public rclcpp::Node{
```

Define una nueva clase llamada `RobotNewsStationNode` que hereda de `rclcpp::Node`, lo que significa que esta clase es un nodo ROS2.

3. Constructor de la clase

```
public:
  RobotNewsStationNode() : Node("robot_news_station"){
    publisher_ =
create_publisher<example_interfaces::msg::String>("robot_news"
, 10);
    timer_ = create_wall_timer(std::chrono::milliseconds(500),
std::bind(&RobotNewsStationNode::publishNews, this));
    RCLCPP_INFO(get_logger(), "Robot News Station has been
started.");
}
```

En el constructor de la clase, se inicializa el nodo con el nombre "robot_news_station". Se crea un publicador que publica en el tema "robot_news". Se

crea un temporizador que llama a la función `publishNews` cada 500 milisegundos. Finalmente, se imprime un mensaje de información.

4. Miembros de datos privados

```
private:
  std::string robot_name_{"R2D2"};

rclcpp::Publisher<example_interfaces::msg::String>::SharedPtr
publisher_{};
  rclcpp::TimerBase::SharedPtr timer_{};
```

Declara tres miembros de datos privados: `robot_name_` que es una cadena que contiene el nombre del robot, `publisher_` que es un puntero compartido a un objeto `Publisher`, y `timer_` que es un puntero compartido a un objeto `TimerBase`.

5. Método `publishNews`

```
   void publishNews(){
       auto msg = example_interfaces::msg::String();
       msg.data = std::string("Hi, this is ") + robot_name_ +
std::string(" from the Robots News Station");
       publisher_->publish(msg);
   }
```

Define un método que crea un mensaje de tipo `String`, establece su contenido y luego lo publica utilizando el publicador.

6. Función `main`

```
   int main(int argc, char **argv){
       rclcpp::init(argc, argv);
       auto node = std::make_shared<RobotNewsStationNode>();
       rclcpp::spin(node);
       rclcpp::shutdown();
       return 0;
   }
```

En la función `main`, se inicializa ROS2, se crea una instancia de `RobotNewsStationNode`, se inicia el ciclo de eventos de ROS2 con `rclcpp::spin`, y finalmente se cierra ROS2 con `rclcpp::shutdown`.

En resumen, este programa crea un nodo ROS2 que publica un mensaje cada 500 milisegundos en el tema "robot_news".

smartphone.cpp

1. Definición de la clase `SmartphoneNode`

```
 class SmartphoneNode : public rclcpp::Node{
```

Define una nueva clase llamada `SmartphoneNode` que hereda de `rclcpp::Node`, lo que significa que esta clase es un nodo ROS2.

2. Constructor de la clase

```
public:
  SmartphoneNode() : Node("smartphone"){
    subscriber_ =
create_subscription<example_interfaces::msg::String>
("robot_news", 10,
std::bind(&SmartphoneNode::callbackRobotNews, this,
std::placeholders::_1));
    RCLCPP_INFO(get_logger(), "Smartphone has been started.");
}
```

En el constructor de la clase, se inicializa el nodo con el nombre "smartphone". Se crea un suscriptor que se suscribe al tema "robot_news". Cuando se recibe un mensaje en este tema, se llama a la función `callbackRobotNews`. Finalmente, se imprime un mensaje de información.

3. Miembro de datos privado

```
private:
rclcpp::Subscription<example_interfaces::msg::String>::SharedP
tr subscriber_;
```

Declara un miembro de datos privado `subscriber_` que es un puntero compartido a un objeto `Subscription`.

4. Método `callbackRobotNews`

```
   void callbackRobotNews(const
example_interfaces::msg::String::SharedPtr msg){
```

```
        RCLCPP_INFO(get_logger(), "%s", msg->data.c_str());
    }
```

Define un método que se llama cuando se recibe un mensaje en el tema al que se suscribe el nodo. Este método imprime el contenido del mensaje.

En resumen, este programa crea un nodo ROS2 que se suscribe al tema "robot_news" e imprime cualquier mensaje que reciba en este tema.