

1.创建包

首先在 ~/ros2_ws/src 目录创建一个名为 mynode 的包：

```
cd ~/ros2_ws/src
```

```
ros2 pkg create --build-type ament_cmake mynode
```

2.编写发布者节点代码

新建一个CPP文件，命名为my_first_node.cpp

```

#include <chrono>
#include <functional>
#include <memory>
#include <string>

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/string.hpp"

using namespace std::chrono_literals;

/* This example creates a subclass of Node and uses std::bind() to register a
 * member function as a callback from the timer. */

class MinimalPublisher : public rclcpp::Node
{
public:
    MinimalPublisher()
    : Node("minimal_publisher"), count_(0)
    {
        publisher_ = this->create_publisher<std_msgs::msg::String>("topic", 10);
        timer_ = this->create_wall_timer(
            500ms, std::bind(&MinimalPublisher::timer_callback, this));
    }
private:
    void timer_callback()
    {
        auto message = std_msgs::msg::String();
        message.data = "Hello, world! " + std::to_string(count_++);
        RCLCPP_INFO(this->get_logger(), "Publishing: '%s'", message.data.c_str());
        publisher_->publish(message);
    }
    rclcpp::TimerBase::SharedPtr timer_;
    rclcpp::Publisher<std_msgs::msg::String>::SharedPtr publisher_;
    size_t count_;
};

int main(int argc, char * argv[])
{
    rclcpp::init(argc, argv);
    rclcpp::spin(std::make_shared<MinimalPublisher>());
    rclcpp::shutdown();
    return 0;
}

```

2.1代码分析

2.2修改package.xml

进入ros2_ws/src/mynode目录并打开package.xml，按照之前教程要求填写description， maintainer和license。如果你并不想开源你的代码，可以忽略此步。

```
<description>TODO: Package description</description>
<maintainer email="nxrobo@todo.todo">nxrobo</maintainer>
<license>TODO: License declaration</license>
```

在编译工具依赖ament_cmake后

```
<buildtool_depend>ament_cmake</buildtool_depend>
```

添加下列依赖项：

```
<depend>rclcpp</depend>
<depend>std_msgs</depend>
```

改写完毕后注意记得保存文档!

2.3修改CmakeLists.txt

打开 CMakeLists.txt，在 find_package(ament_cmake REQUIRED) 下添加两行：

```
find_package(rclcpp REQUIRED)
find_package(std_msgs REQUIRED)
```

然后在添加可执行文件需要编译的源文件，并命名为 talker或者其他你认为合适的名字

```
add_executable(talker src/my_first_node.cpp)
ament_target_dependencies(talker rclcpp std_msgs)
```

3.编写订阅者节点代码

再次回到ros2_ws/src/cpp_pubsub/src目录下，并建立一个subscriber_member_function.cpp的文件，并写入以下内容：

```
#include <memory>

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/string.hpp"
using std::placeholders::_1;

class MinimalSubscriber : public rclcpp::Node
{
public:
    MinimalSubscriber()
    : Node("minimal_subscriber")
    {
        subscription_ = this->create_subscription<std_msgs::msg::String>(
            "topic", 10, std::bind(&MinimalSubscriber::topic_callback, this, _1));
    }

private:
    void topic_callback(const std_msgs::msg::String & msg) const
    {
        RCLCPP_INFO(this->get_logger(), "I heard: '%s'", msg.data.c_str());
    }
    rclcpp::Subscription<std_msgs::msg::String>::SharedPtr subscription_;
};

int main(int argc, char * argv[])
{
    rclcpp::init(argc, argv);
    rclcpp::spin(std::make_shared<MinimalSubscriber>());
    rclcpp::shutdown();
    return 0;
}
```