由于消息会在不同的进程之间传递，为了能够在不同的项目之间共享消息类型，避免同一个类型的消息在多个项目之间重复定义，也为了消息类型的名称更加清楚，现于Share项目里创建了SharedMessageTypes.cpp/.h两个文件(Share/TinyROS/目录下)，大家以后都将消息类型写在里面，就可以在不同的项目之间共享。以下是详细说明：

1. 所有使用TinyROS的项目都会引用Share项目，因此会自然地引用新增的两个文件。但是新增的两个文件不被包含在TinyROS.h头文件中，因此需要在自己的代码中单独包含SharedMessageTypes.h



1. 大家的消息类型都会往这里写，因此需要遵循以下编辑规范。首先注意，这些内容放在RoboTax命名空间下。
2. 使用编译器指令：一对#pragma region/endregion ，把一种消息相关的一段代码括起来，这样就可以使用折叠功能，折叠前后效果如图：

文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

注意在#pragma region 后面直接添加文本，给这块区域命名

1. 每段代码内部使用注释注明编辑者、消息的内容等。也可以改别人写的，注明编辑者即可。
2. 常见使用场景1：仅仅为类型起一个别名：

例如，使用SimpleObjectMessage<float>传输小车的当前朝向角度（一个浮点数），可以使用using语句为它起一个别名



之后，就可以使用CarDirectionMessage这个名称，与SimpleObjectMessage<float>等价。（这句话是在RoboTax命名空间下的，在外面访问的话需要加命名空间限定）。使用类型的别名可以让含义更加明显，建议给SimpleObjectMessage起别名。一个类型可以有多个别名，即使是同一个类型，在不同的环境下可能有不同的含义，因此可以起不同的别名，它们全都是等价的。

1. 常见使用场景2：定义自己的结构体，使用SimpleObjectMessage传输

如图，直接在SharedMessageTypes头文件中定义结构体就可以，然后推荐为此SimpleObjectMessage起别名。这样，所有的项目都能够使用这个类型，而无需每个项目都定义一遍这个结构体。

文本

描述已自动生成

1. 常见使用场景3：定义自己的消息类型

可以按照TinyROS使用说明中的步骤，继承自Message以实现自己的消息类型，这时，头文件里写类型的声明，源文件里写类里面各个方法的实现。但如果要写模板类，就直接全写在头文件里。

1. 提醒：请各位注意好自己所负责的模块，考虑好自己会使用到什么样类型的消息。消息肯定不是单方的，因此请与通信的另一方做好沟通，共同确立消息的内容，然后再编写消息类型。在编辑文件之前，先看看现有的类型里是不是已经有了所需的，