Lesson 7 The Definition and Use Of Topics Message

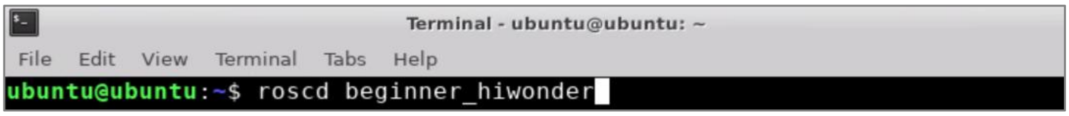
1. Customize Topic Message

Note: Before customizing the topic message, the workspace and package need to be created first. The specific operation steps can be viewed in "ROS Basic Lessons/Lesson 3 Create Workspace and Package.

The specific operation steps for customizing topic massages are as following:

1) Open the terminal.

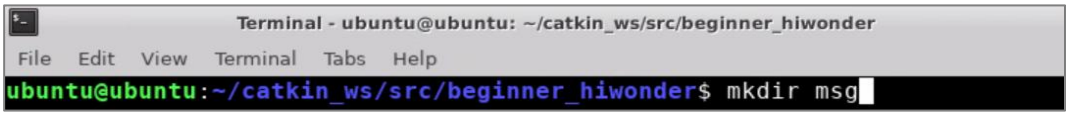
2) Enter "roscd beginner_hiwonder" command to locate to the package directory and press "Enter".


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ roscd beginner_hiwonder
```

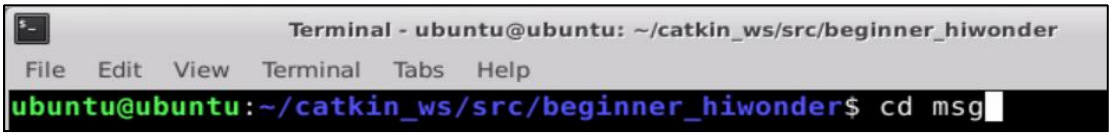
Note: If there is a prompt "**No such package/stack 'beginner_hiwonder**" appears, it means that the package does not exist in the environment variable ROS_PACKAGE_PATH. The specific solution can be viewed in "ROS Basic Lessons/Lesson 3 Create Workspace and Package". After the problem is solved, please repeat the current step.

3) Enter "mkdir msg" command and press "Enter". Then create a new folder "msg" for storing text files.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ mkdir msg
```

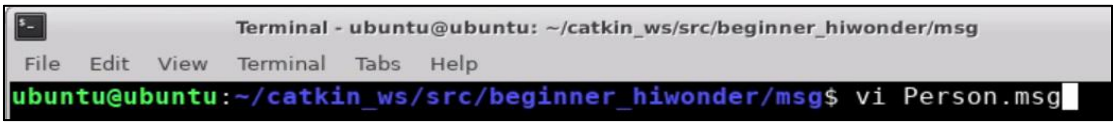
4) Enter "cd msg" command and press "Enter".



Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ cd msg
```

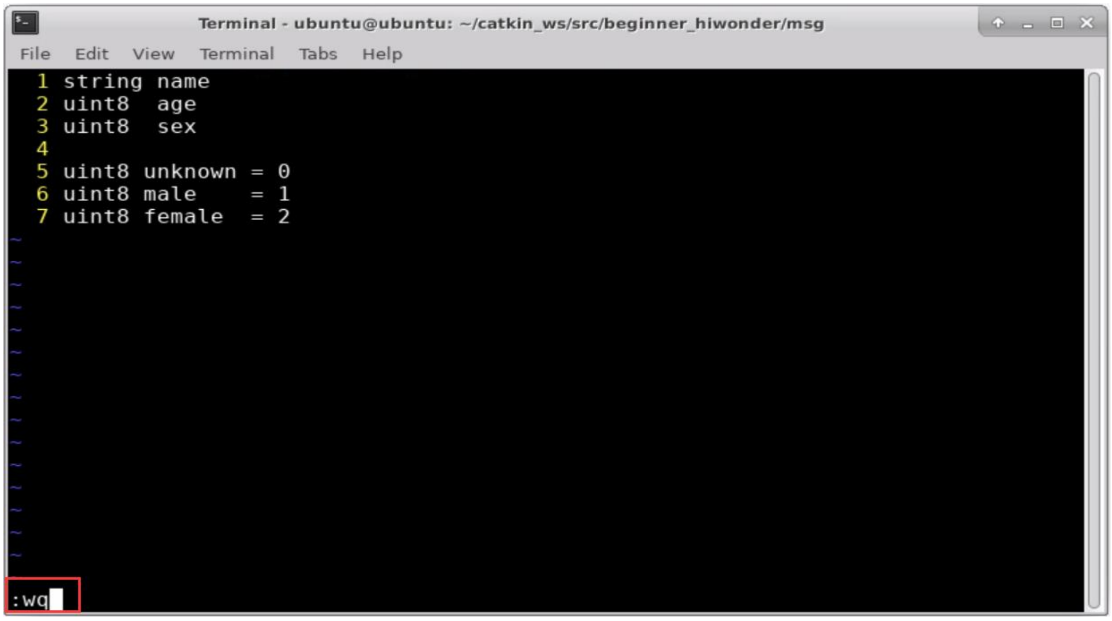
5) Enter “vi Person.msg” command to edit program and copy the following program. If want to modify, you can press “i” key. After modifying, press “Esc” and enter “:wq” to save and exit.



Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/msg
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/msg$ vi Person.msg
```

```
string name

uint8   age

uint8   sex



uint8 unknown = 0

uint8 male      = 1

uint8 female    = 2
```

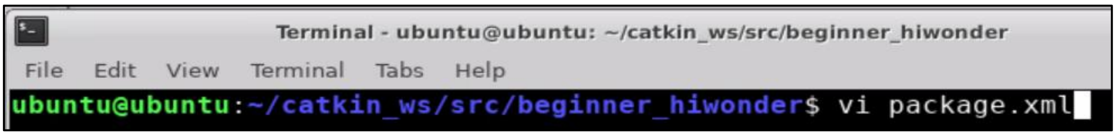


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/msg
File Edit View Terminal Tabs Help

```
string name
uint8  age
uint8  sex

uint8 unknown = 0
uint8 male    = 1
uint8 female  = 2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

6) Enter “vi package.xml” command. Then copy the following program and

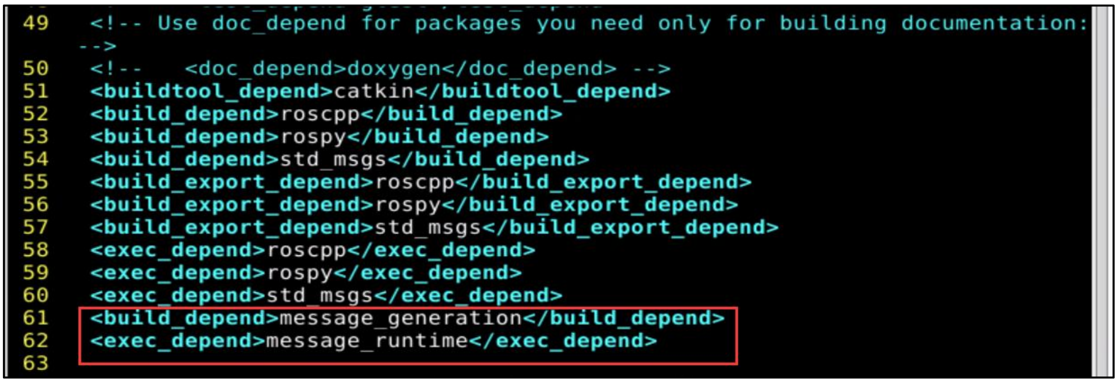
add the package dependencies in the position shown in the figure below. If want to modify, you can press “i” again. After modifying, press “Esc” and enter “:wq” to save and exit.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ vi package.xml
```

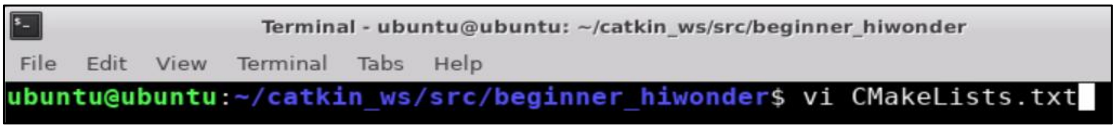
7) Add the package dependencies in the position shown in the below figure:

```
<build_depend>message_generation</build_depend>

<exec_depend>message_runtime</exec_depend>
```



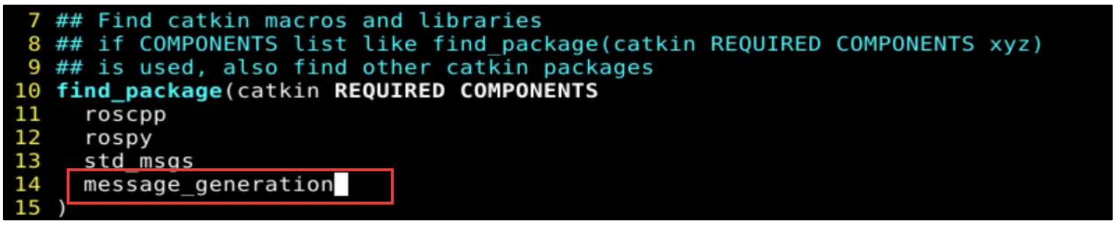
```
  <!-- Use doc_depend for packages you need only for building documentation:
-->
  <!--   <doc_depend>doxygen</doc_depend> -->
  <buildtool_depend>catkin</buildtool_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>std_msgs</build_depend>
  <build_export_depend>roscpp</build_export_depend>
  <build_export_depend>rospy</build_export_depend>
  <build_export_depend>std_msgs</build_export_depend>
  <exec_depend>roscpp</exec_depend>
  <exec_depend>rospy</exec_depend>
  <exec_depend>std_msgs</exec_depend>
  <build_depend>message_generation</build_depend>
  <exec_depend>message_runtime</exec_depend>
```

8) Enter “vi CMakeLists.txt” and press “i” to modify “CMakeLists.txt” file.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

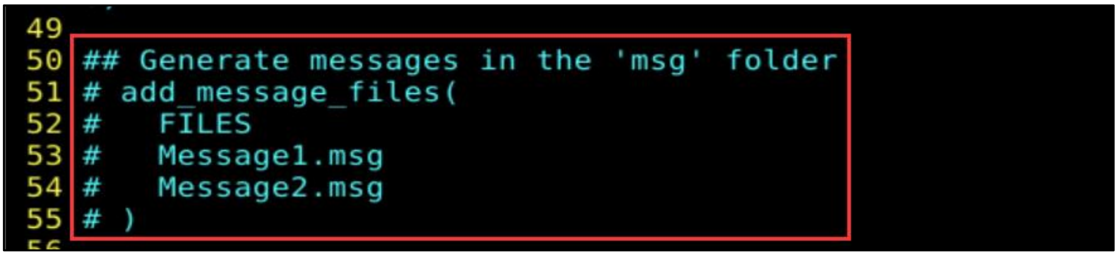
```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ vi CMakeLists.txt
```

9) Add the required compiation option “message_generation” in the position shown in the figure below.

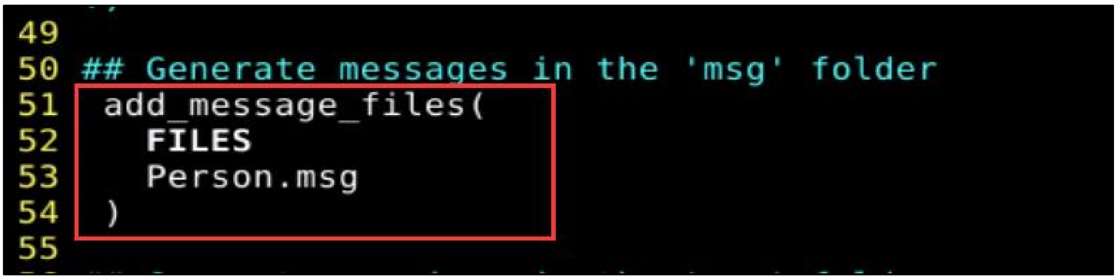


```
## Find catkin macros and libraries
## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
## is used, also find other catkin packages
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
  message_generation
)
```

10) Locate the code shown in the figure below. Then uncomment the framed code and add the required compilation option “Person.msg”.

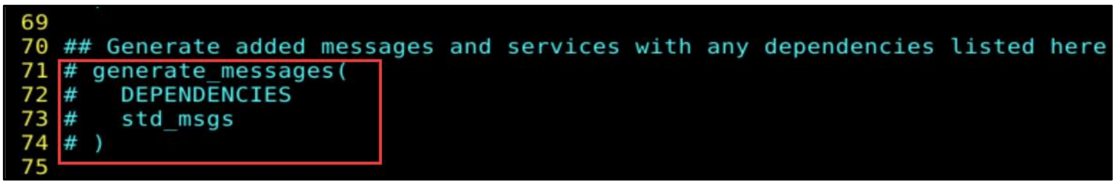


```
## Generate messages in the 'msg' folder
# add_message_files(
#   FILES
#   Message1.msg
#   Message2.msg
# )
```

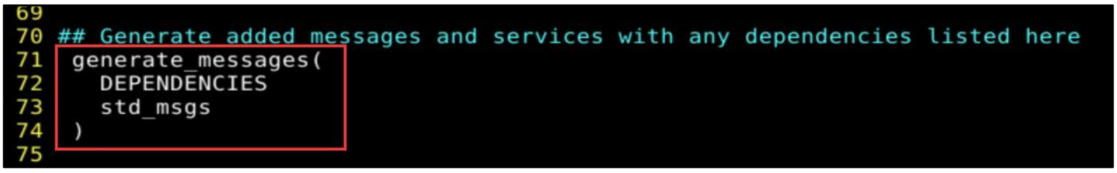


```
## Generate messages in the 'msg' folder
 add_message_files(
   FILES
   Person.msg
 )
```

11) Find the code shown in the figure below. Then uncomment the code in red box and ensure that the required compilation options take effect.

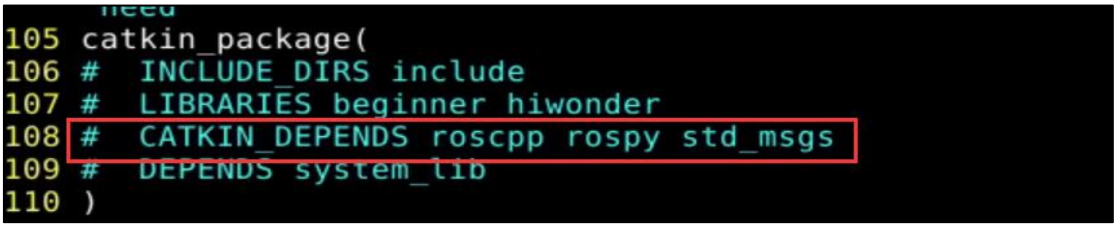


```
## Generate added messages and services with any dependencies listed here
# generate_messages(
#   DEPENDENCIES
#   std_msgs
# )
```

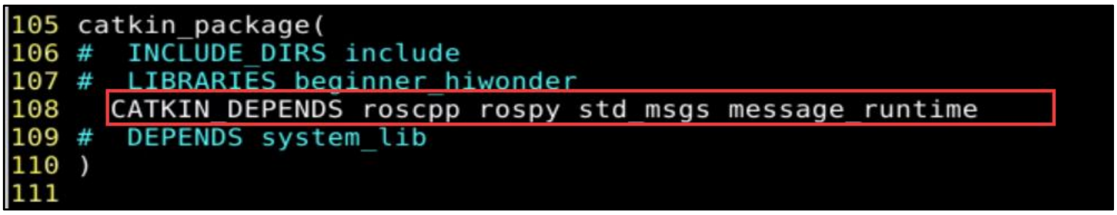


```
## Generate added messages and services with any dependencies listed here
 generate_messages(
   DEPENDENCIES
   std_msgs
 )
```

12) Find the code shown in the figure below. Then uncomment the code in red box and add the required compilation option “message_runtime”.

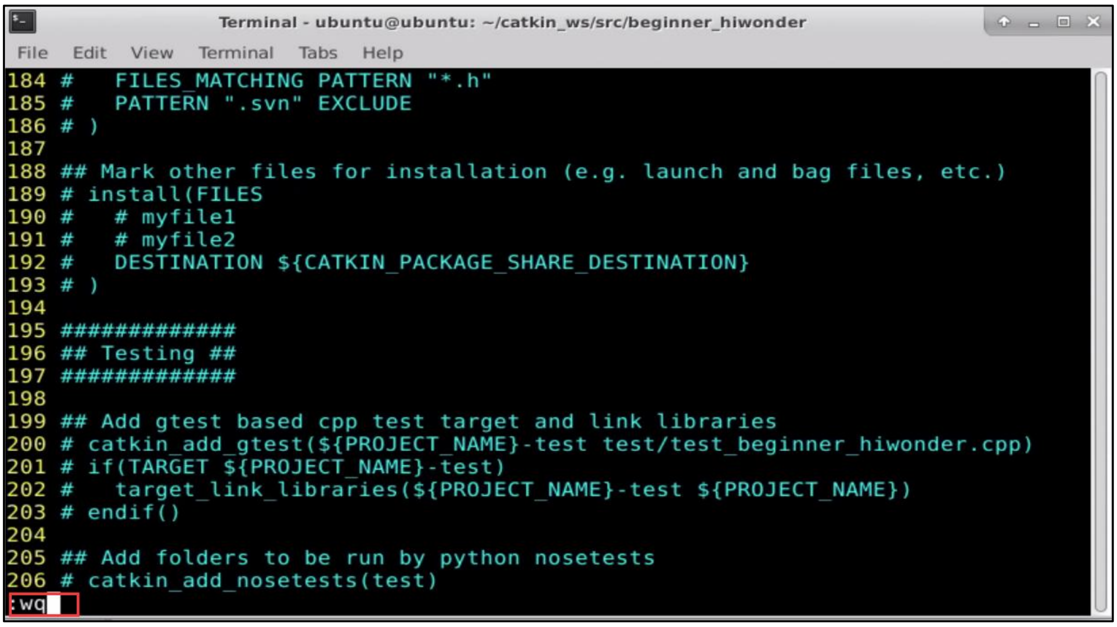


```
catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES beginner_hiwonder
#  CATKIN_DEPENDS roscpp rospy std_msgs
#  DEPENDS system_lib
)
```



```
catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES beginner_hiwonder
   CATKIN_DEPENDS roscpp rospy std_msgs message_runtime
#  DEPENDS system_lib
)
```

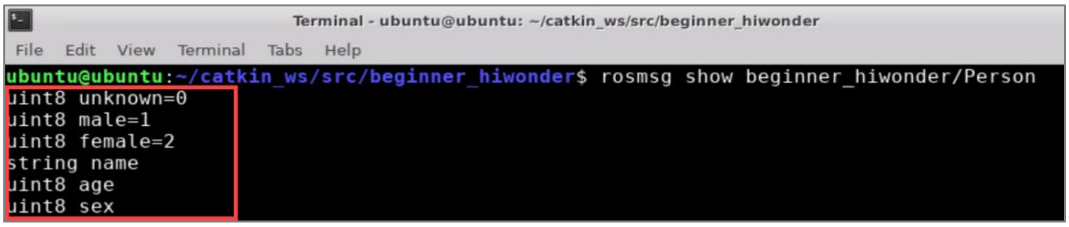
13) After modifying, press “Esc” and enter “:wq” to save and exit.



Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

```
#   FILES_MATCHING PATTERN "*.h"
#   PATTERN ".svn" EXCLUDE
# )

## Mark other files for installation (e.g. launch and bag files, etc.)
# install(FILES
#   # myfile1
#   # myfile2
#   DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
# )

#############
## Testing ##
#############

## Add gtest based cpp test target and link libraries
# catkin_add_gtest(${PROJECT_NAME}-test test/test_beginner_hiwonder.cpp)
# if(TARGET ${PROJECT_NAME}-test)
#   target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
# endif()

## Add folders to be run by python nosetests
# catkin_add_nosetests(test)
:wq
```

14) Enter the command “rosmsg show beginner_hiwonder/Person” and press “Enter” to check whether the message written can be recognized by system. When the words shown in red box appear, it means that they are recognized successfully.



Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

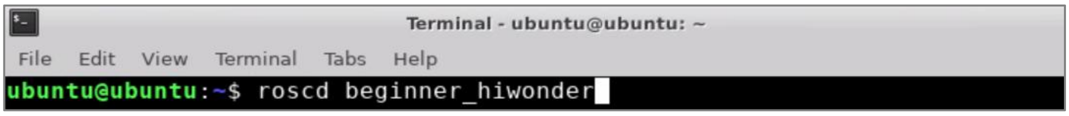
```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ rosmsg show beginner_hiwonder/Person
uint8 unknown=0
uint8 male=1
uint8 female=2
string name
uint8 age
uint8 sex
```

2. The Use of Topic Message

2.1 Create Publisher and Subscriber Code

1) Open the terminal.

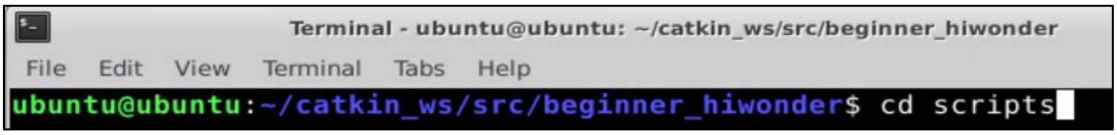
2) Enter “roscd beginner_hiwonder” command to locate to the package directory and press “Enter”.



Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

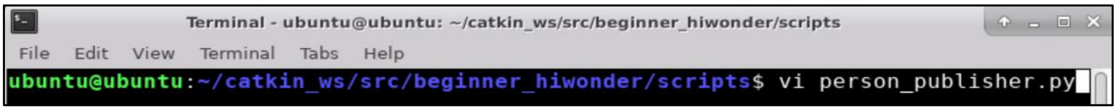
```
ubuntu@ubuntu:~$ roscd beginner_hiwonder
```

3) Enter “cd scripts” command and press “Enter” to come to the folder “scripts” where Python scripts are stored.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help

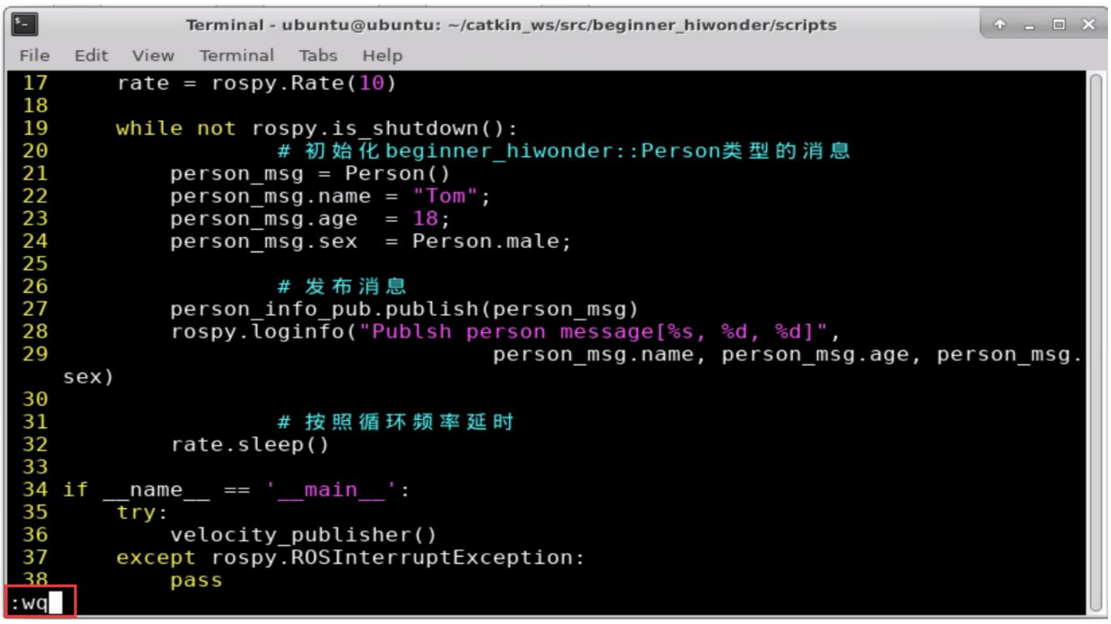
```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ cd scripts
```

4) Enter “vi person_publisher.py” command to edit progam, and then copy the following program. If want to modify, you can press “i”. After modifying, press “Esc” and enter “:wq” to save and exit.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help

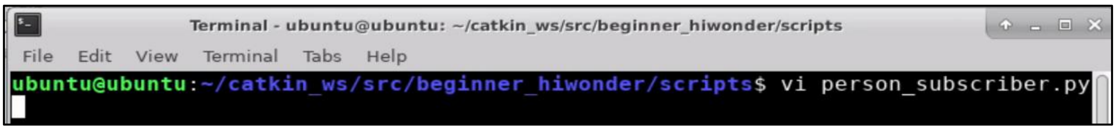
```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ vi person_publisher.py
```

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-



#This routine will pulish /person_info topic and customize message type beginner_hiwonder::Person.



import rospy

from beginner_hiwonder.msg import Person



def velocity_publisher():

    # Initialize ROS node

    rospy.init_node('person_publisher', anonymous=True)

    # Create a publisher and publish a topic named /person_info. The message type is beginner_hiwonder::Person and the queue length is 10.
    person_info_pub = rospy.Publisher('/person_info', Person, queue_size=10)

    #set the loop rate
    rate = rospy.Rate(10)

    while not rospy.is_shutdown():
        # Initialize the message of beginner_hiwonder::Person type
        person_msg = Person()
        person_msg.name = "Tom";
        person_msg.age  = 18;
        person_msg.sex  = Person.male;

        # publish message
        person_info_pub.publish(person_msg)
        rospy.loginfo("Publsh person message[%s, %d, %d]",
                person_msg.name, person_msg.age, person_msg.sex)

        # Delay on basis of the loop rate
        rate.sleep()

if __name__ == '__main__':
    try:
        velocity_publisher()

    except rospy.ROSInterruptException:

        pass
```



```
    rate = rospy.Rate(10)

    while not rospy.is_shutdown():
                # 初始化beginner_hiwonder::Person类型的消息
        person_msg = Person()
        person_msg.name = "Tom";
        person_msg.age  = 18;
        person_msg.sex  = Person.male;

                # 发布消息
        person_info_pub.publish(person_msg)
        rospy.loginfo("Publsh person message[%s, %d, %d]",
                                person_msg.name, person_msg.age, person_msg.
sex)

                # 按照循环频率延时
        rate.sleep()

if __name__ == '__main__':
    try:
        velocity_publisher()
    except rospy.ROSInterruptException:
        pass
:wq
```

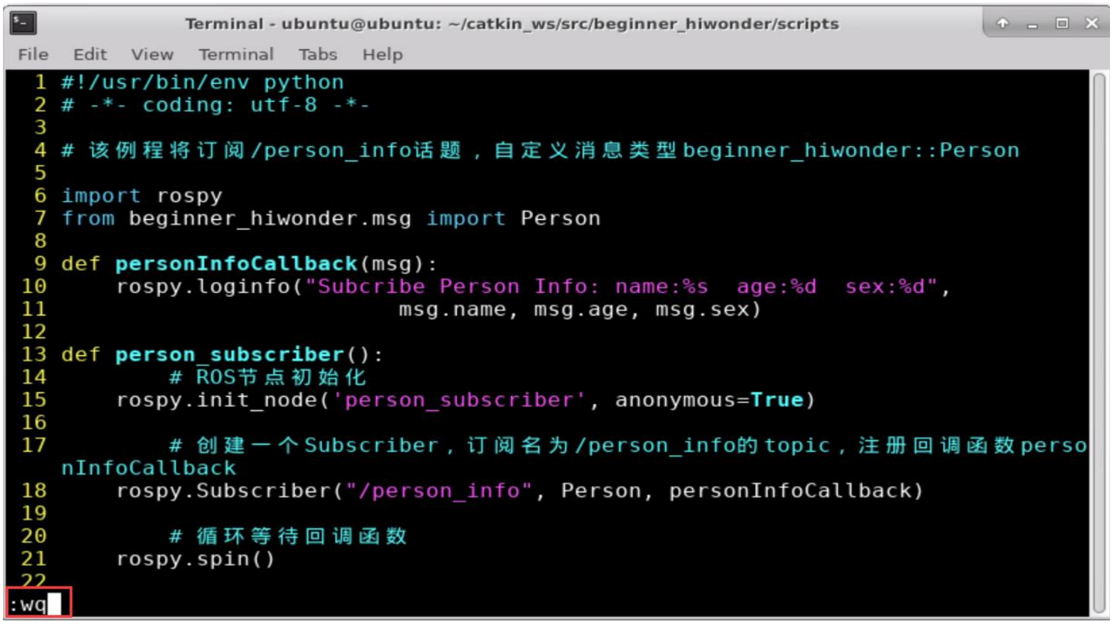
5) Enter “vi person_subscriber.py” to edit program, and copy the following program. If want to modify, you can press “i”. After modifying, press “Esc” and enter “:wq” to save and exit.



```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ vi person_subscriber.py
```

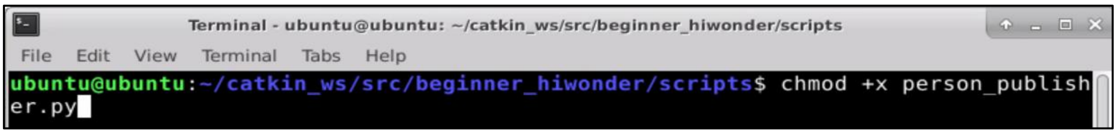
```
#!/usr/bin/env python

# -*- coding: utf-8 -*-



# This routine will subscribe to the /person_info topic and customize the message type beginner_hiwonder::Person



import rospy
```

```
from beginner_hiwonder.msg import Person

def personInfoCallback(msg):
    rospy.loginfo("Subcribe Person Info: name:%s  age:%d  sex:%d",
                  msg.name, msg.age, msg.sex)

def person_subscriber():
    # Initialize ROS node
    rospy.init_node('person_subscriber', anonymous=True)

    # Create a subscriber, subsribe the topic named /person_info and register the callback funcation personInfoCallback
    rospy.Subscriber("/person_info", Person, personInfoCallback)

    # loop and wait the callback function
    rospy.spin()

if __name__ == '__main__':
    person_subscriber()
```


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# 该例程将订阅/person_info话题，自定义消息类型beginner_hiwonder::Person

import rospy
from beginner_hiwonder.msg import Person

def personInfoCallback(msg):
    rospy.loginfo("Subcribe Person Info: name:%s  age:%d  sex:%d",
                         msg.name, msg.age, msg.sex)

def person_subscriber():
        # ROS节点初始化
    rospy.init_node('person_subscriber', anonymous=True)

        # 创建一个Subscriber，订阅名为/person_info的topic，注册回调函数personInfoCallback
    rospy.Subscriber("/person_info", Person, personInfoCallback)

        # 循环等待回调函数
    rospy.spin()

:wq
```

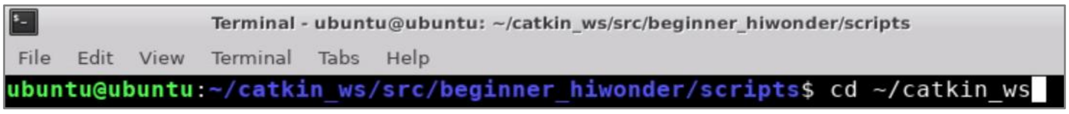
6) Enter “chmod +x person_publisher.py” command and press “Enter” to give the executable permission to the saved person_publisher.py.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ chmod +x person_publisher.py
```

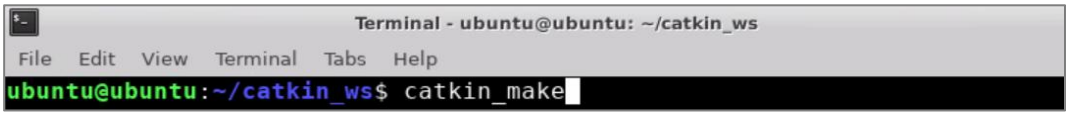
2.2 Run Publisher And Subscriber Nodes

1) Enter “cd ~/catkin_ws” command and press “Enter” to enter to catkin workspace.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help

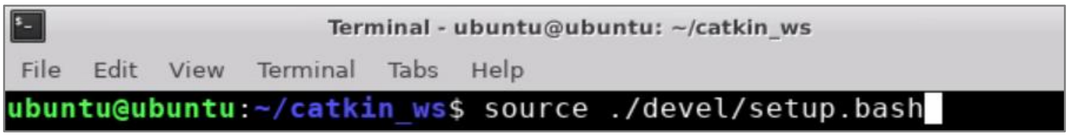
```
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ cd ~/catkin_ws
```

2) Enter “catkin_make” command and press “Enter” to build all the packages in directory.


Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help

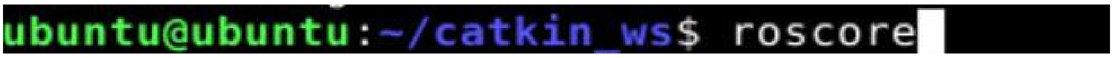
```
ubuntu@ubuntu:~/catkin_ws$ catkin_make
```

3) Enter “source ./devel/setup.bash” command and press “Enter” to refresh the workspace environment.


Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help

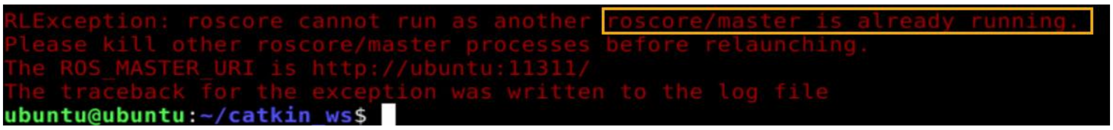
```
ubuntu@ubuntu:~/catkin_ws$ source ./devel/setup.bash
```

4) Enter “roscore” command to start node manager.



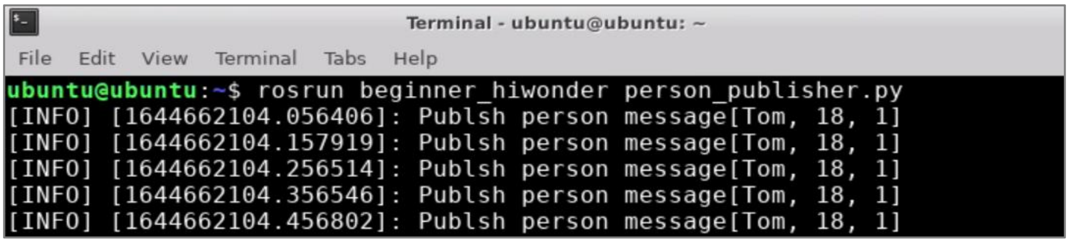
```
ubuntu@ubuntu:~/catkin_ws$ roscore
```

After starting, the prompt below will appear:



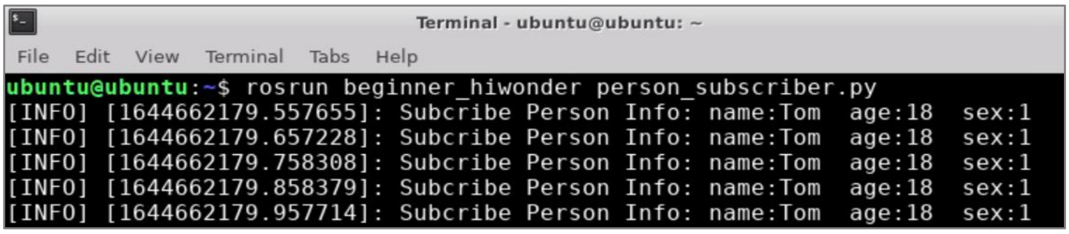
```
RLException: roscore cannot run as another roscore/master is already running.
Please kill other roscore/master processes before relaunching.
The ROS_MASTER_URI is http://ubuntu:11311/
The traceback for the exception was written to the log file
ubuntu@ubuntu:~/catkin_ws$
```

5) Enter “rosrun beginner_hiwonder person_publisher.py” command and press “Enter” to run publisher node. If want to stop running node, you can press “Ctrl+C”.


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ rosrun beginner_hiwonder person_publisher.py
[INFO] [1644662104.056406]: Publsh person message[Tom, 18, 1]
[INFO] [1644662104.157919]: Publsh person message[Tom, 18, 1]
[INFO] [1644662104.256514]: Publsh person message[Tom, 18, 1]
[INFO] [1644662104.356546]: Publsh person message[Tom, 18, 1]
[INFO] [1644662104.456802]: Publsh person message[Tom, 18, 1]
```

6) Open a new terminal. Enter “rosrun beginner_hiwonder person_subscriber.py” command and press “Enter” to run the subscriber node. If want to stop running node, you can press “Ctrl+C”.


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ rosrun beginner_hiwonder person_subscriber.py
[INFO] [1644662179.557655]: Subcribe Person Info: name:Tom  age:18  sex:1
[INFO] [1644662179.657228]: Subcribe Person Info: name:Tom  age:18  sex:1
[INFO] [1644662179.758308]: Subcribe Person Info: name:Tom  age:18  sex:1
[INFO] [1644662179.858379]: Subcribe Person Info: name:Tom  age:18  sex:1
[INFO] [1644662179.957714]: Subcribe Person Info: name:Tom  age:18  sex:1
```

① The publisher node needs to be started first, and then the subscriber node can subscribe message.

② If need to receive the publisher messages completely, you can start the subscriber node first and then the publisher node.