ROS2 – 13 เขียน Publisher / Subscriber ด้วยภาษา Python



เขียน Publisher และ Subscriber ด้วยภาษา Python

Topics : ข้อความจะถูกส่งผ่านระบบการขนส่งที่มีการ Publisher / Subscriber โหนดที่ส่งข้อความจะเรียกว่า Publisher ชื่อที่ระบุใน Topic จะใช้เพื่อระบุเนื้อหาของข้อความ โหนดที่สนใจในข้อมูลบางประเภทจะทำการ Subscriber หัวข้อที่ต้องการ อาจมีการ Publisher / Subscribe หลายรายพร้อมกันสำหรับหัวข้อเดียวและโหนดเดียวอาจ Publisher / Subscriber ได้หลาย Topics โดยทั่วไป Publisher / Subscriber จะไม่ได้ตระหนักถึงการมีอยู่ของกันและกัน กล่าวอีกนัยคือการแยกกันผลิตข้อมูลตามหน้าที่ของตนเอง

Topic ก็เหมือนกับหัวเรื่องที่เราไว้คุยกัน พูดง่ายๆคือ Publisher Node จะ Publish Message ของตัวเองขึ้นไปที่ Topic แล้วจะมี Subscriber Node มารอรับ Message ที่ส่งมาทาง Topic ที่มีชื่อเดียวกัน

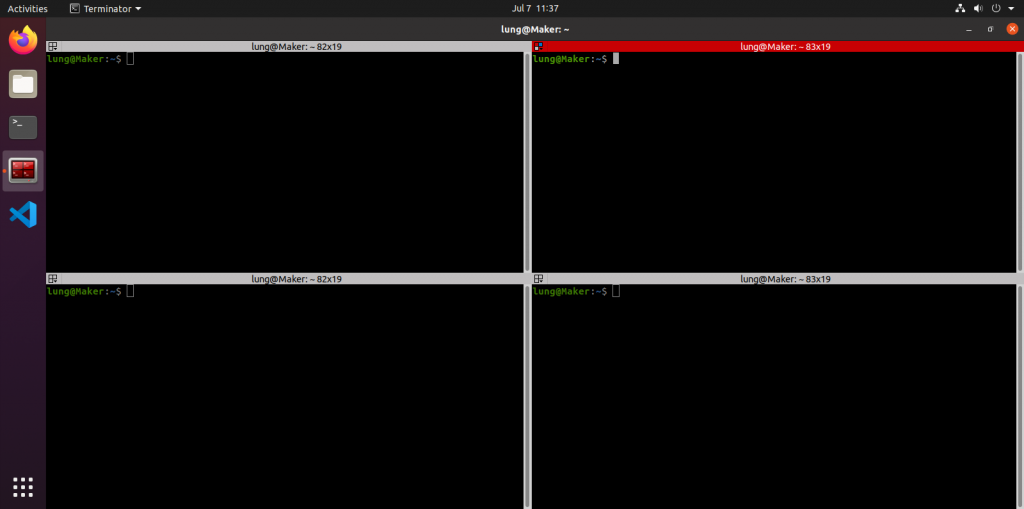
เวลาพูดว่า Publisher หมายถึงการที่ Node จะส่ง Message ออกไปที่ Topic โดยไม่สนใจว่าจะมี Node รอรับอยู่รึเปล่า Node 1 ตัวสามารถ Publisher Message ออกไปได้หลายๆ Topics

Message : ข้อความ ค่ามาตรฐาน (จำนวนเต็ม, จุดลอย, บูลีน, ฯลฯ ) ข้อมูลอาร์เรย์ทั่วๆไป รวมไปถึงข้อมูลอาร์เรย์ที่มีความซับซ้อน

**1 : เขียน Node ภาษา Python แบบ OOP**  
เขียน Node ภาษา Python แบบ OOP ตามขั้นตอนลิงค์ด้านล่าง

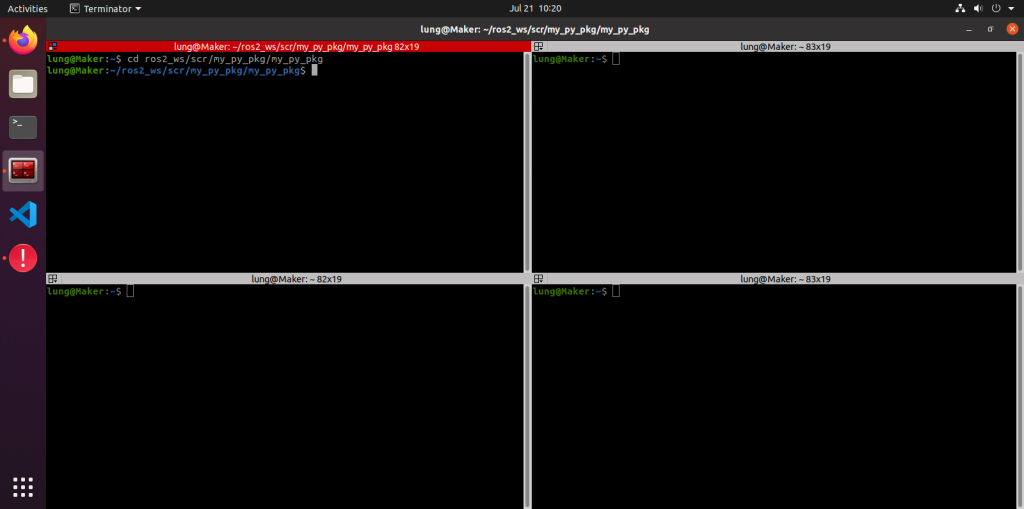
* [เขียน Node ภาษา Python แบบ OOP](http://www.lungmaker.com/ros2-9-%e0%b9%80%e0%b8%82%e0%b8%b5%e0%b8%a2%e0%b8%99-node-%e0%b8%a0%e0%b8%b2%e0%b8%a9%e0%b8%b2-python-%e0%b9%81%e0%b8%9a%e0%b8%9a-oop/)

**2 : เปิด Terminator 4 หน้าต่าง**

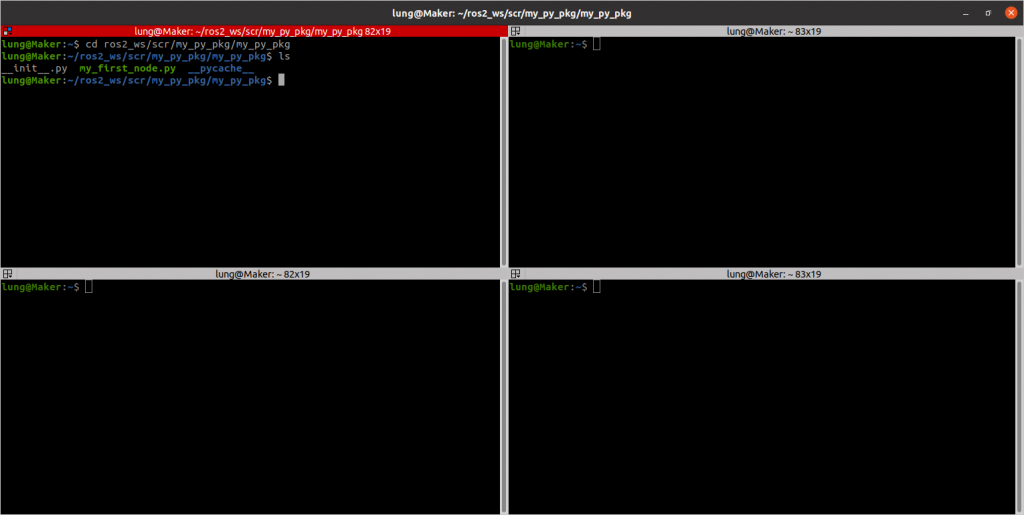


**3 : เขียน Publisher ด้วยภาษา Python**  
เขียนโหนดส่งข้อความ Publisher ด้วยภาษา Python  
โดยไปที่ ไปที่ หน้าต่างที่ 1 ใช้คำสั่งเข้าไปใน โฟลเดอร์ที่ต้องการสร้างไฟล์

cd ros2\_ws/scr/my\_py\_pkg/my\_py\_pkg

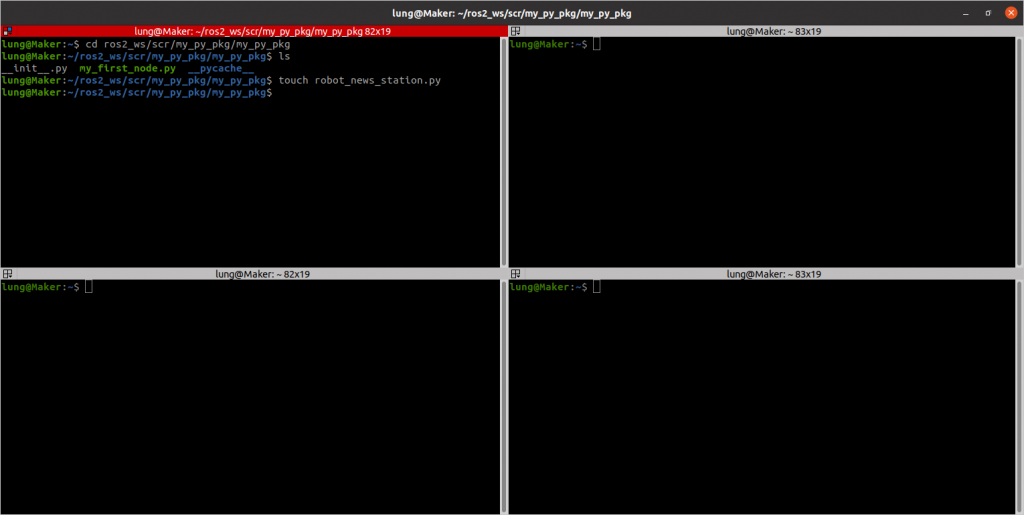


ls



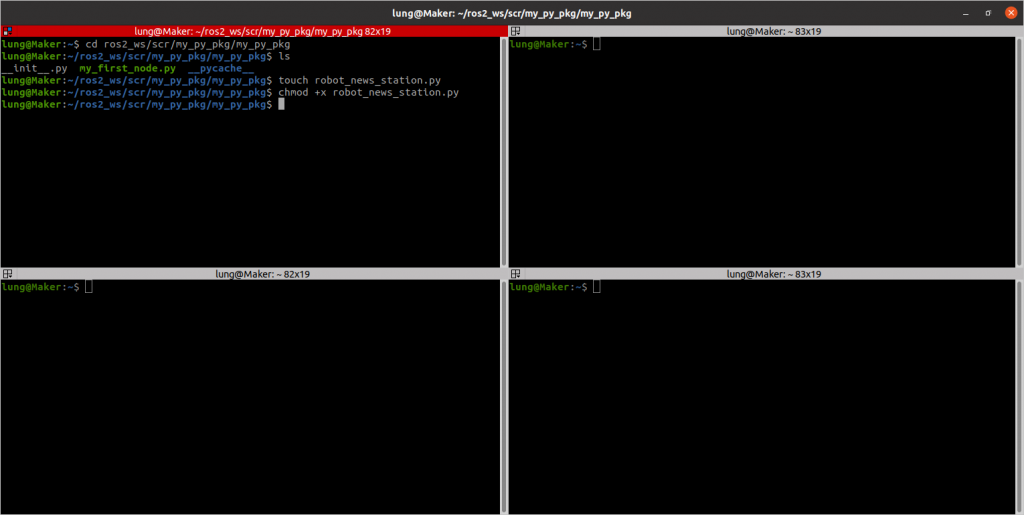
สร้างไฟล์ robot\_news\_station.py

touch robot\_news\_station.py

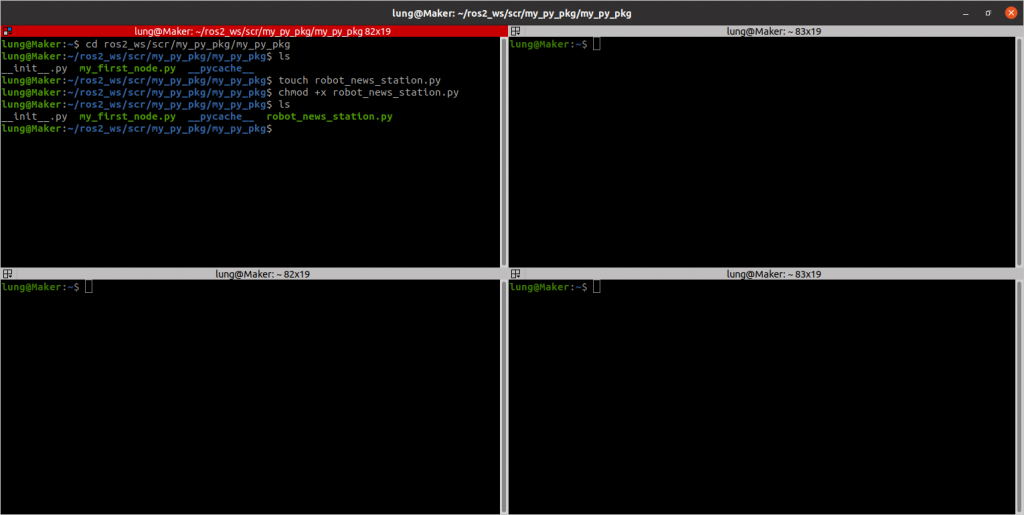


คำสั่ง chmod คำสั่งใช้เปลี่ยนสิทธิของไฟล์ ตัวอักษร x มาจาก Execute หมายถึง ประมวลผล

chmod +x robot\_news\_station.py



ls



ใช้ Visual Studio Code เปิดไฟล์ robot\_news\_station.py เขียนโค้ด + Save ตามโค้ดด้านล่าง

#!/usr/bin/env python3

import rclpy

from rclpy.node import Node

from example\_interfaces.msg import String

class RobotNewsStationNode(Node):

def \_\_init\_\_(self):

super().\_\_init\_\_("robot\_news\_station")

self.robot\_name\_ = "C3PO"

self.publisher\_ = self.create\_publisher(String, "robot\_news", 10)

self.timer\_ = self.create\_timer(0.5, self.publish\_news)

self.get\_logger().info("Robot News Station has been started")

def publish\_news(self):

msg = String()

msg.data = "Hi, this is " + \

str(self.robot\_name\_) + " from the robot news station."

self.publisher\_.publish(msg)

def main(args=None):

rclpy.init(args=args)

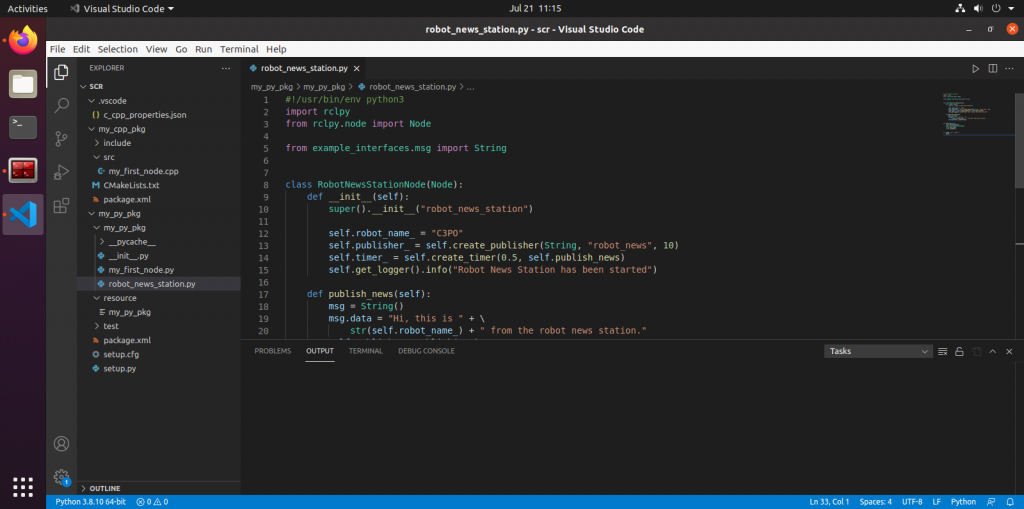
node = RobotNewsStationNode()

rclpy.spin(node)

rclpy.shutdown()

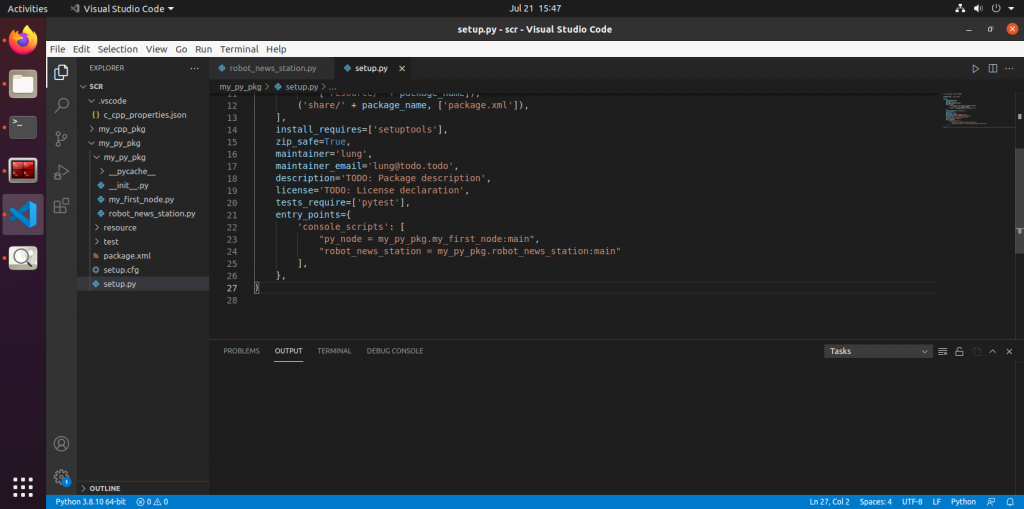
if \_\_name\_\_ == "\_\_main\_\_":

main()



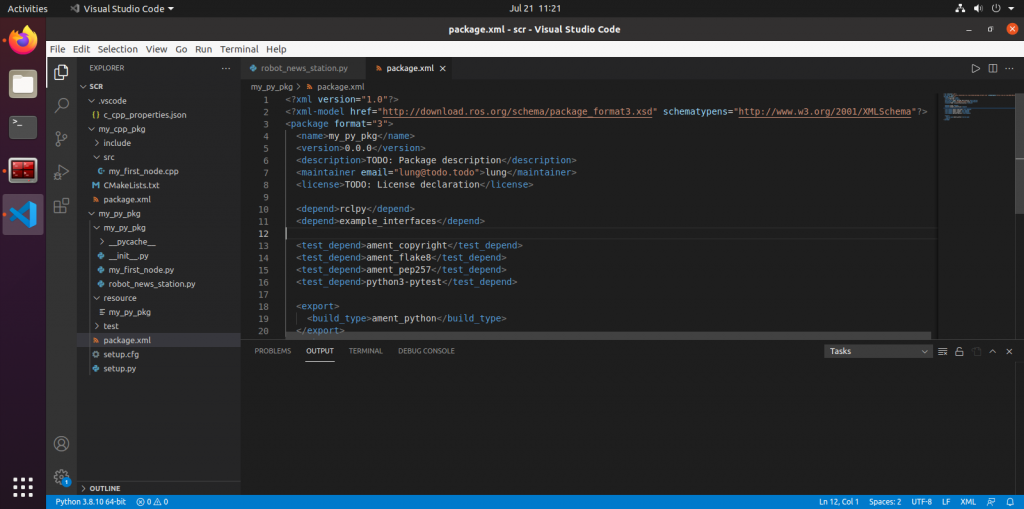
เปิดไฟล์ setup.py เพิ่ม จุลภาค หรือ **ลูกน้ำ** ( , ) หลังบรรทัด 23 แล้ว เพิ่มโค้ดบรรทัดที่ 24 ล่าง ด้วยโค้ดด้านล่าง

"robot\_news\_station = my\_py\_pkg.robot\_news\_station:main"



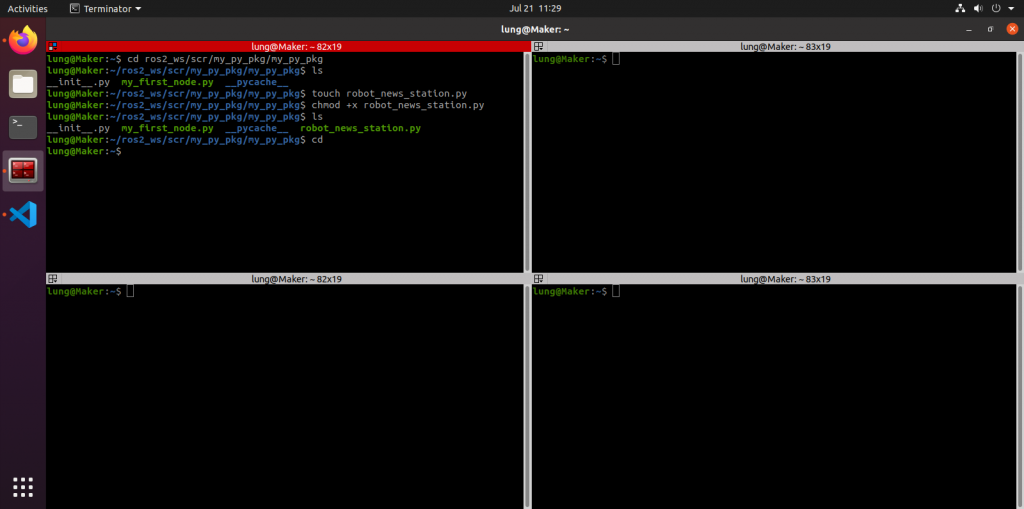
เปิดไฟล์ package.xml เพิ่มโค้ดบรรทัดที่ 11 ด้วยโค้ดด้านล่าง

<depend>example\_interfaces</depend>

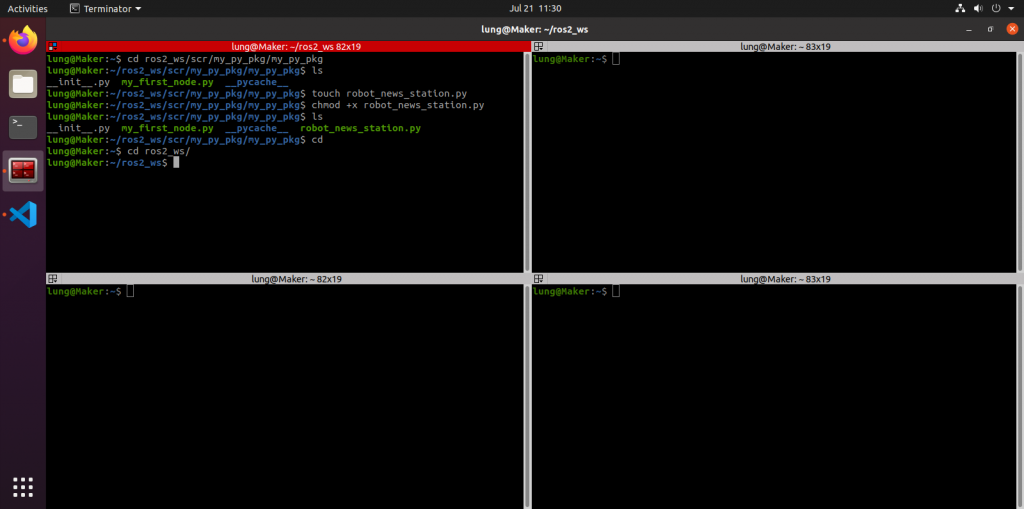


กลับสู่ Home เพื่อจะ Build แพ็กเกจ my\_py\_pkg

cd

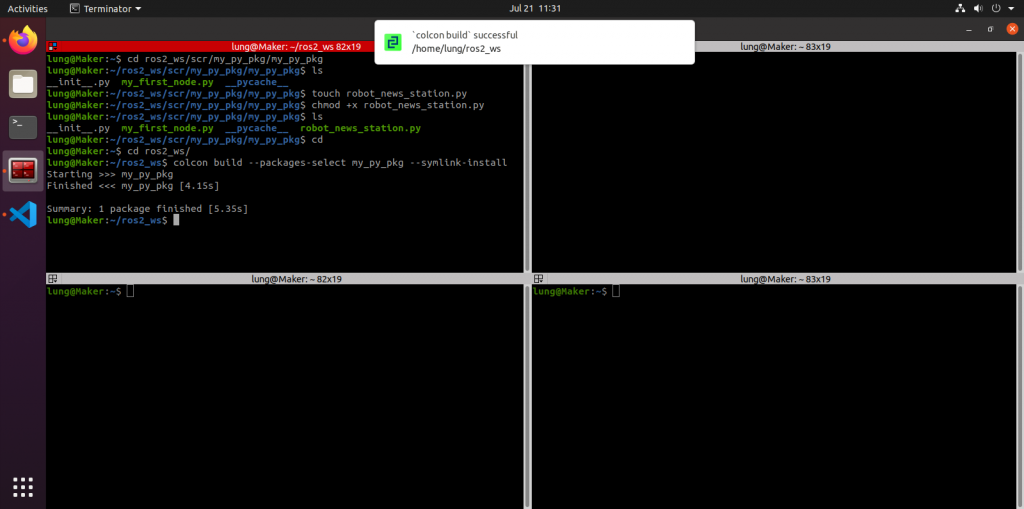


cd ros2\_ws/



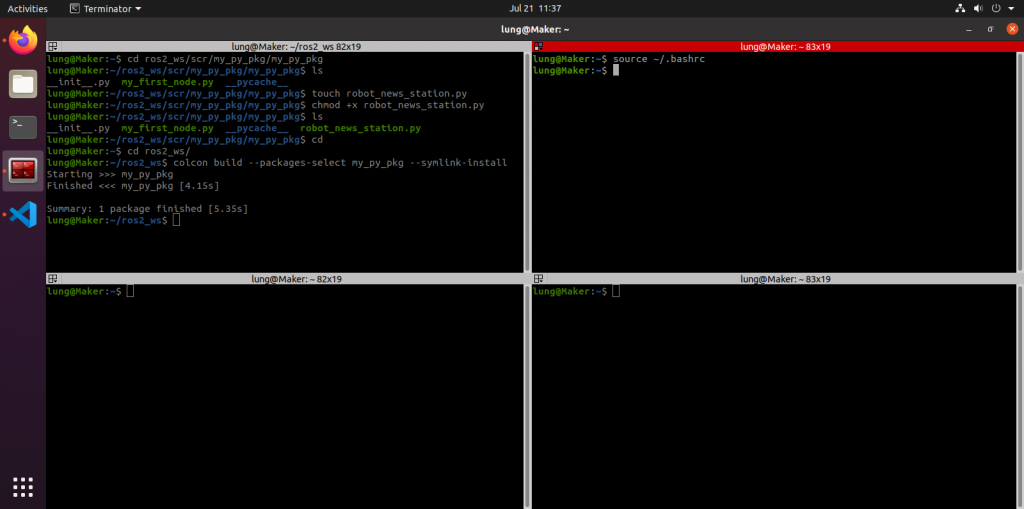
คำสั่ง Build เฉพาะ แพ็กเกจ my\_py\_pkg

colcon build --packages-select my\_py\_pkg --symlink-install



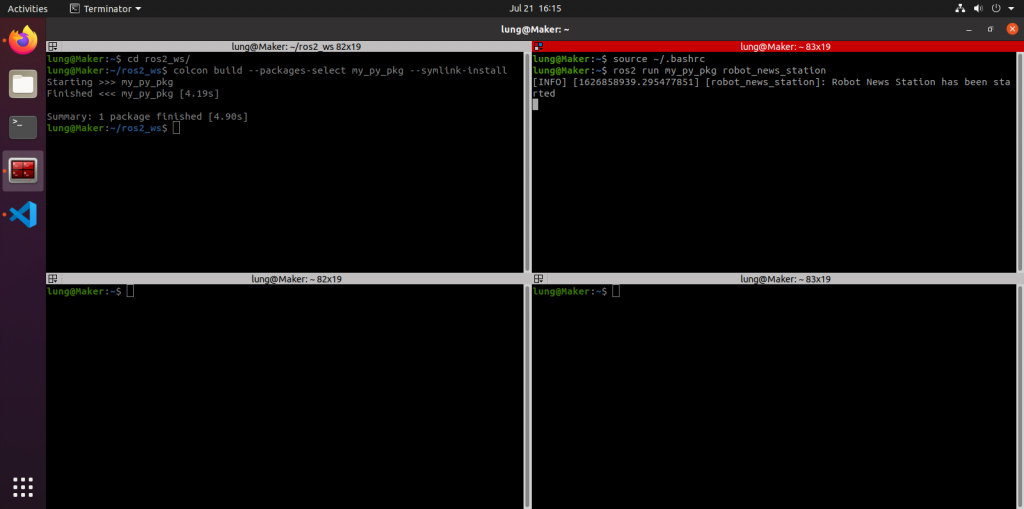
ไปที่ หน้าต่างที่ 2 โดยเราจะต้องทําให้ Terminator รู้จักคําสั่งของ ROS2 ด้วย คําสั่ง

source ~/.bashrc



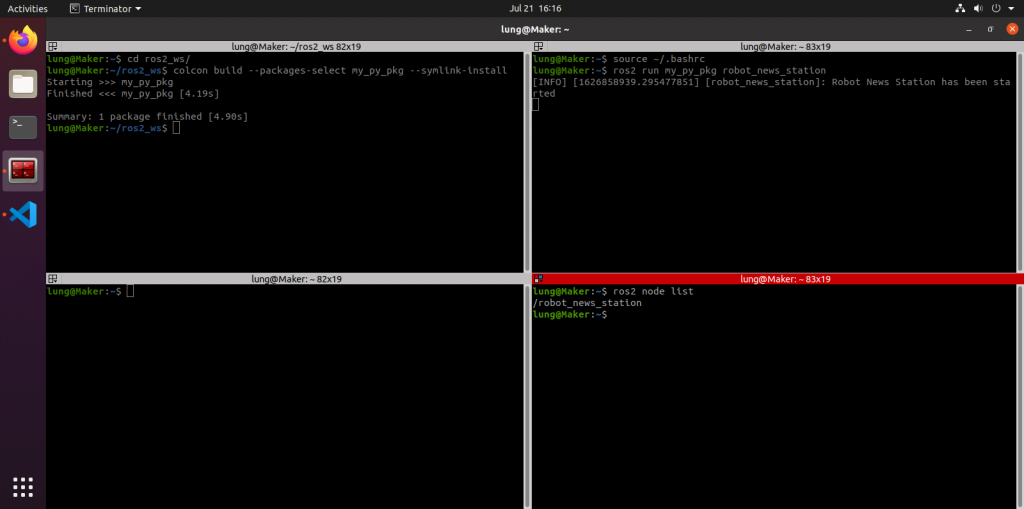
เรียกใช้งาน robot\_news\_station ด้วย ROS2

ros2 run my\_py\_pkg robot\_news\_station



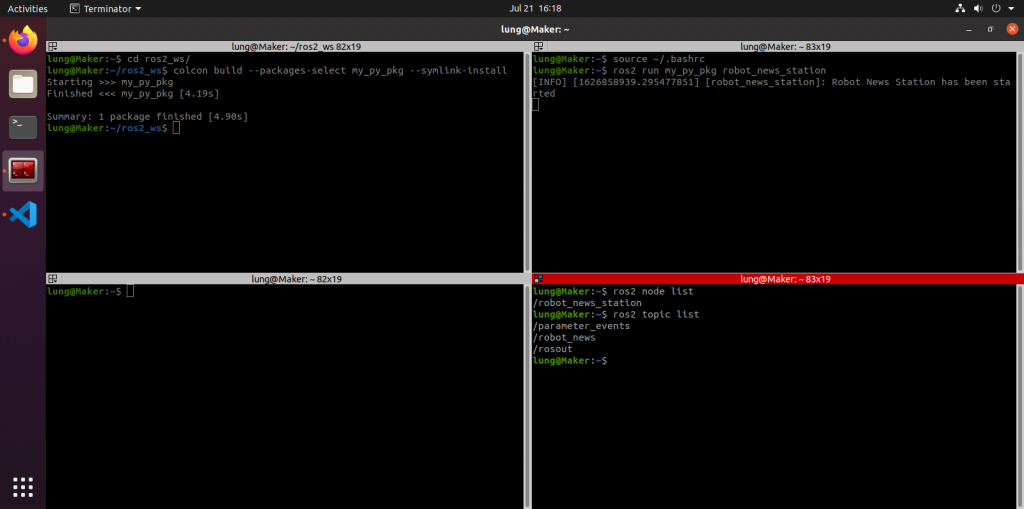
ไปที่ หน้าต่างที่ 4 ให้แสดง Node ที่กำลังทำงานอยู่

ros2 node list



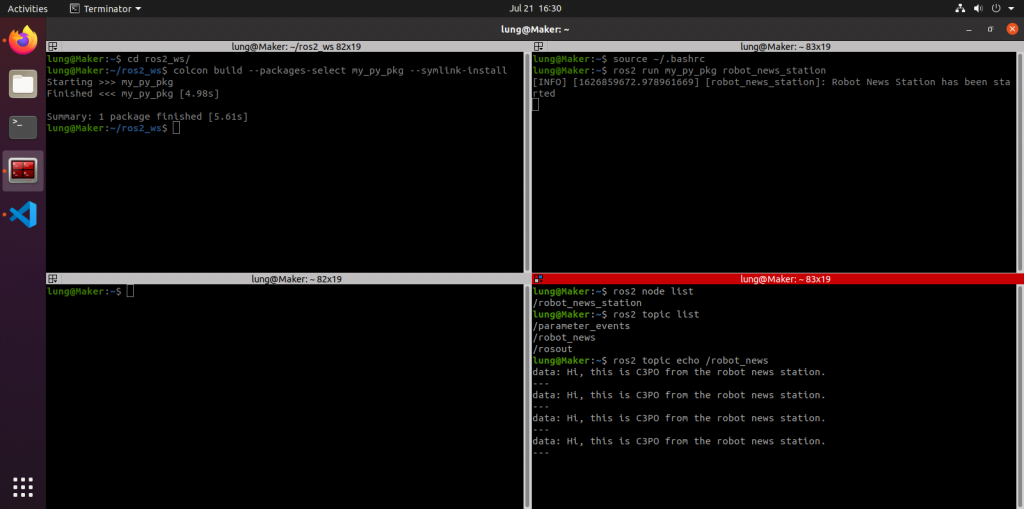
แสดง Topic ที่กำลังทำงานอยู่

ros2 topic list



แสดงข้อมูลที่ส่งออกมาจาก topic robot\_news

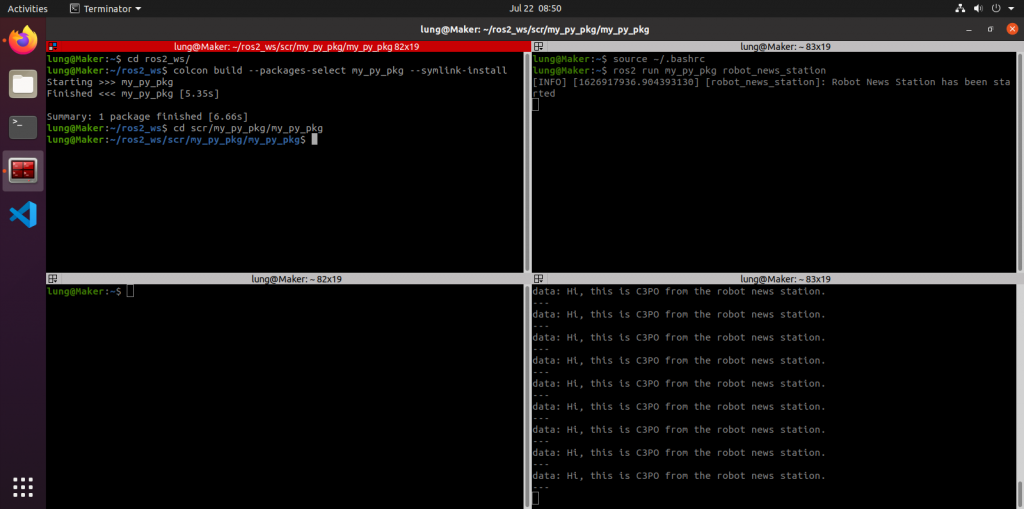
ros2 topic echo /robot\_news



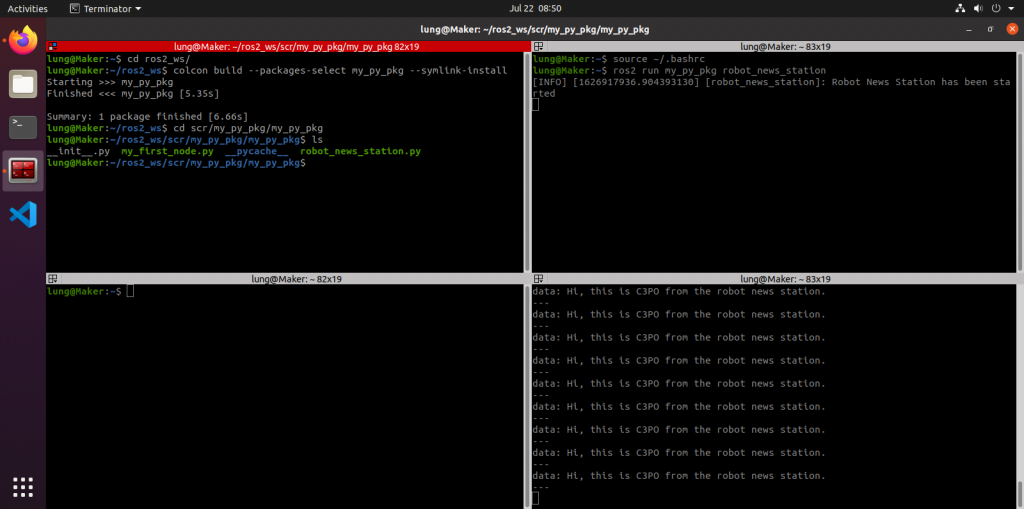
**4 : เขียน Subscriber ด้วยภาษา Python**  
Subscriber หมายถึงการที่ Node จะรับ Message จาก Topic ที่ตัวเองรอรับอยู่ Node 1 ตัวสามารถ Subscriber Message ที่มาจากหลายๆ Topic ได้

เขียน โหนด รับข้อความ Subscriber ด้วยภาษา Python โดยกลับไปที่ หน้าต่างที่ 1  ใช้คำสั่งเข้าไปใน โฟลเดอร์ที่ต้องการสร้างไฟล์

cd scr/my\_py\_pkg/my\_py\_pkg

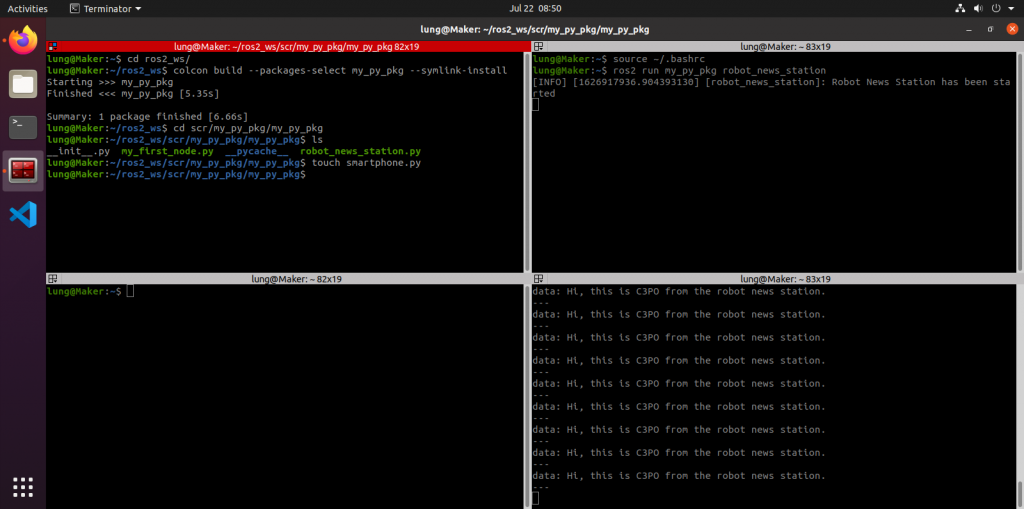


ls



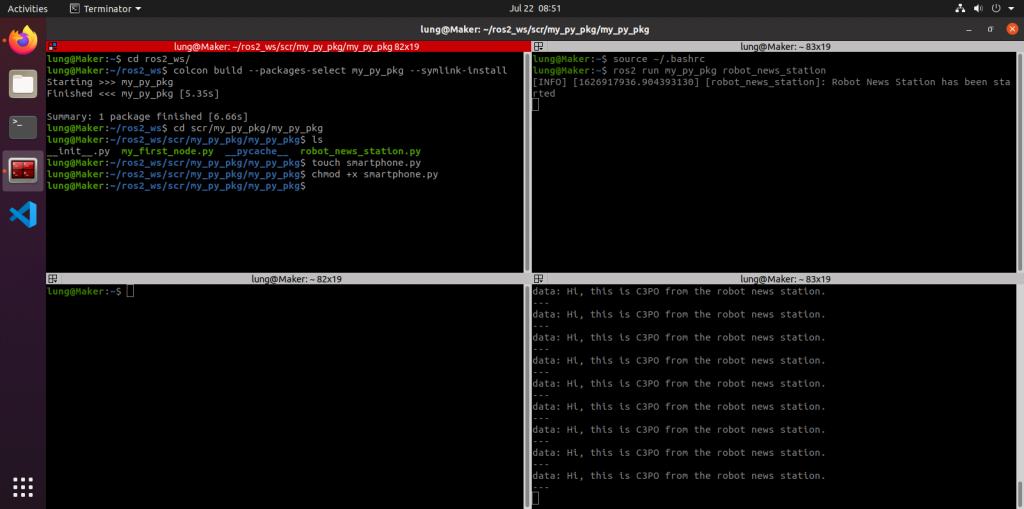
สร้างไฟล์ smartphone.py

touch smartphone.py



คำสั่ง chmod คำสั่งใช้เปลี่ยนสิทธิของไฟล์ ตัวอักษร x มาจาก Execute หมายถึง ประมวลผล

chmod +x smartphone.py



ใช้ Visual Studio Code เปิดไฟล์ smartphone.py เขียนโค้ด + Save ตามโค้ดด้านล่าง

#!/usr/bin/env python3

import rclpy

from rclpy.node import Node

from example\_interfaces.msg import String

class SmartphoneNode(Node):

def \_\_init\_\_(self):

super().\_\_init\_\_("smartphone")

self.subscriber\_ = self.create\_subscription(

String, "robot\_news", self.callback\_robot\_news, 10)

self.get\_logger().info("Smartphone has been started.")

def callback\_robot\_news(self, msg):

self.get\_logger().info(msg.data)

def main(args=None):

rclpy.init(args=args)

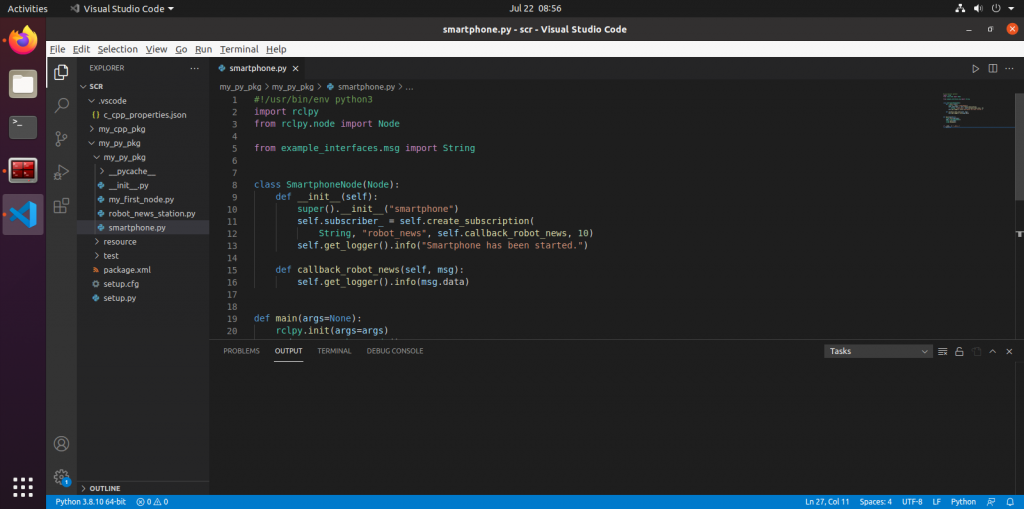
node = SmartphoneNode()

rclpy.spin(node)

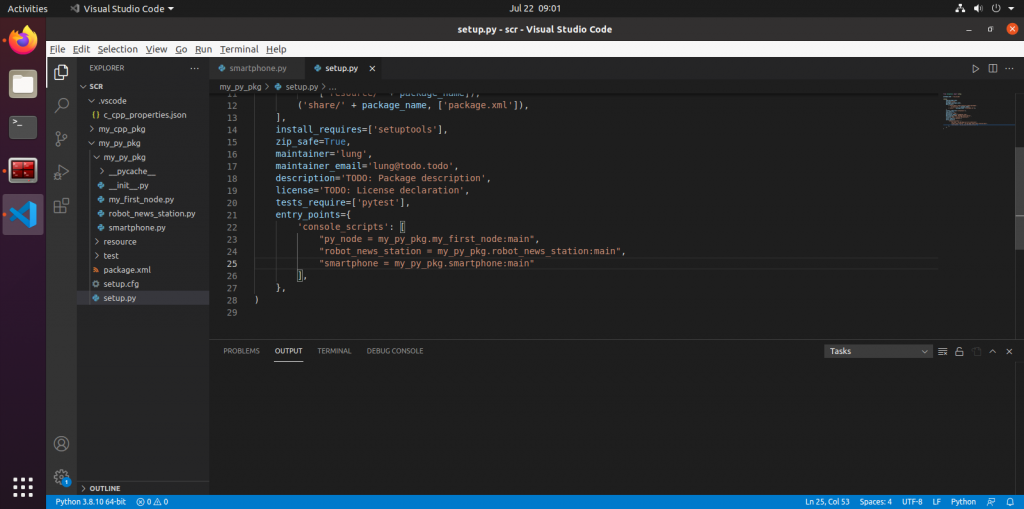
rclpy.shutdown()

if \_\_name\_\_ == "\_\_main\_\_":

main()

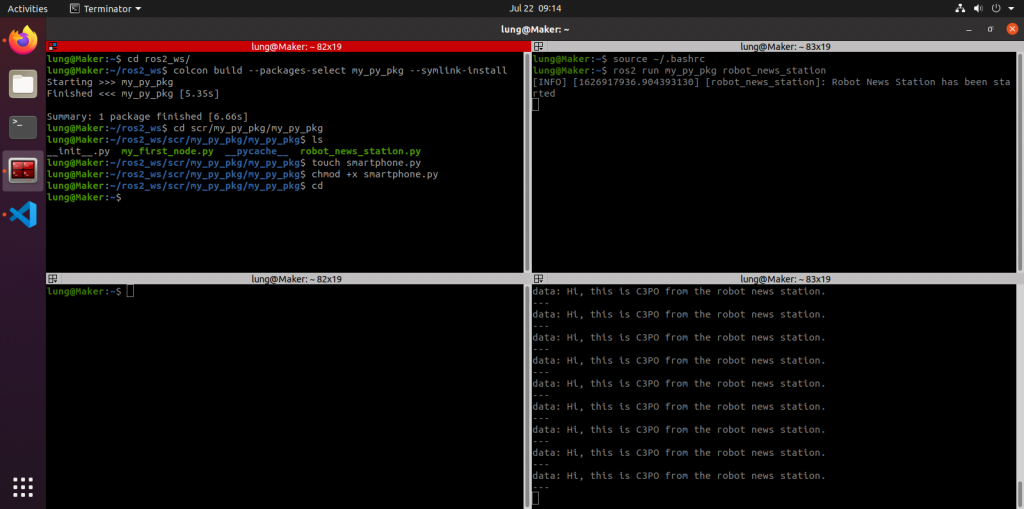
  
เปิดไฟล์ setup.py เพิ่ม จุลภาค หรือ **ลูกน้ำ** ( , ) หลังบรรทัด 24 แล้ว เพิ่มโค้ดบรรทัดที่ 25 ล่าง ด้วยโค้ดด้านล่าง

"smartphone = my\_py\_pkg.smartphone:main"

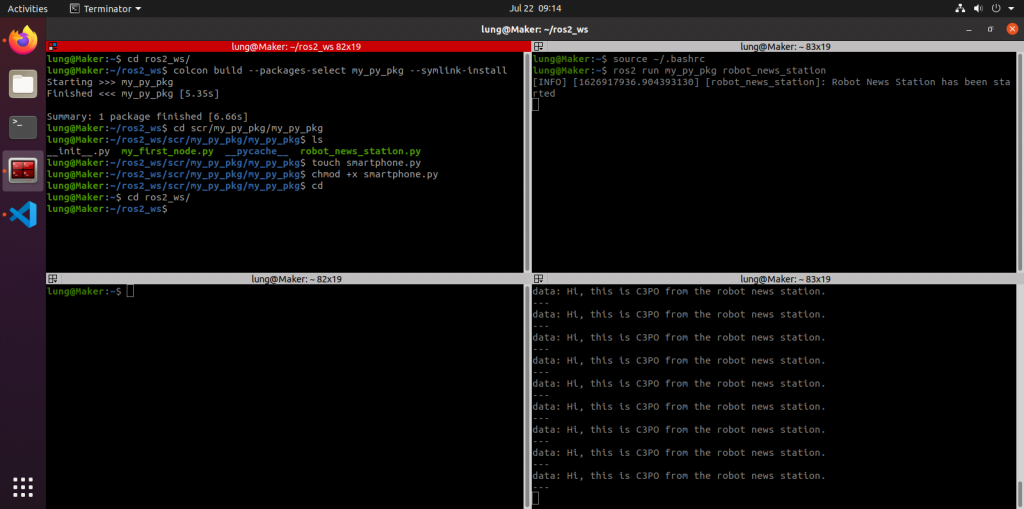


ไปที่ หน้าต่างที่ 1 กลับสู่ Home เพื่อจะ Build แพ็กเกจ my\_py\_pkg

cd

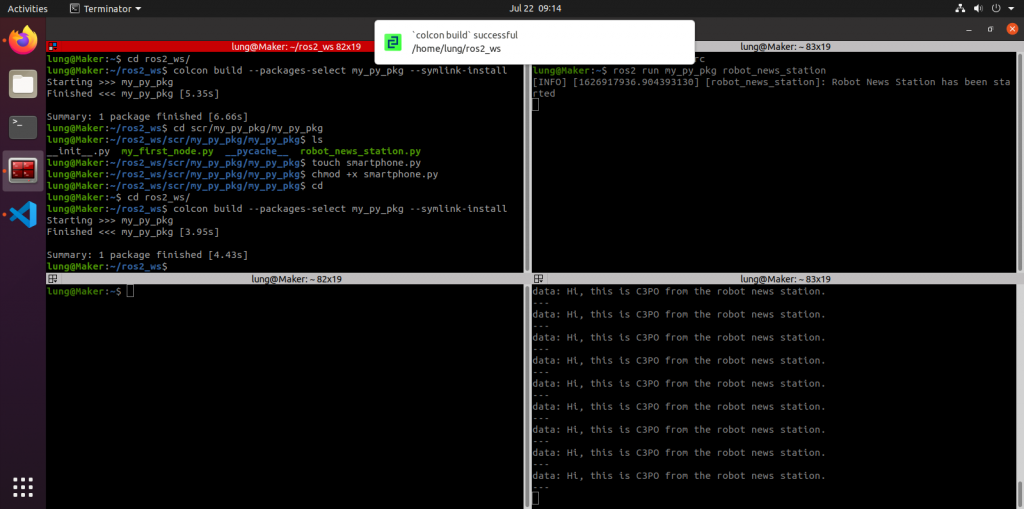


cd ros2\_ws/



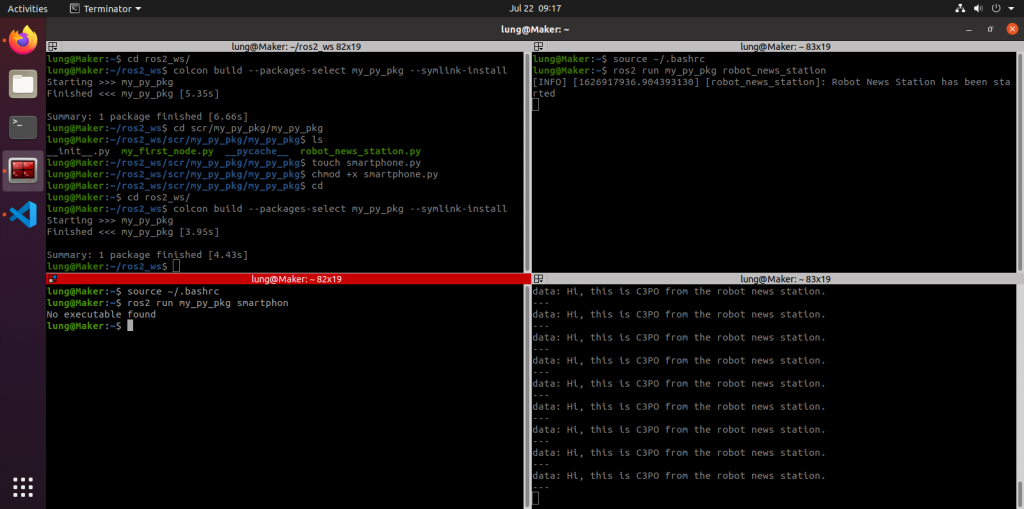
คำสั่ง Build เฉพาะ แพ็กเกจ my\_py\_pkg

colcon build --packages-select my\_py\_pkg --symlink-install



ไปที่ หน้าต่างที่ 2 โดยเราจะต้องทําให้ Terminator รู้จักคําสั่งของ ROS2 ด้วย คําสั่ง

source ~/.bashrc

  
เรียกใช้งาน smartphone ด้วย ROS2 จะแสดงการรับ Message จากที่ส่งออกมาจาก topic robot\_news

ros2 run my\_py\_pkg smartphone

