

ROS2 강습회

대한기계학회 IT융합부문

명지대학교 최동일

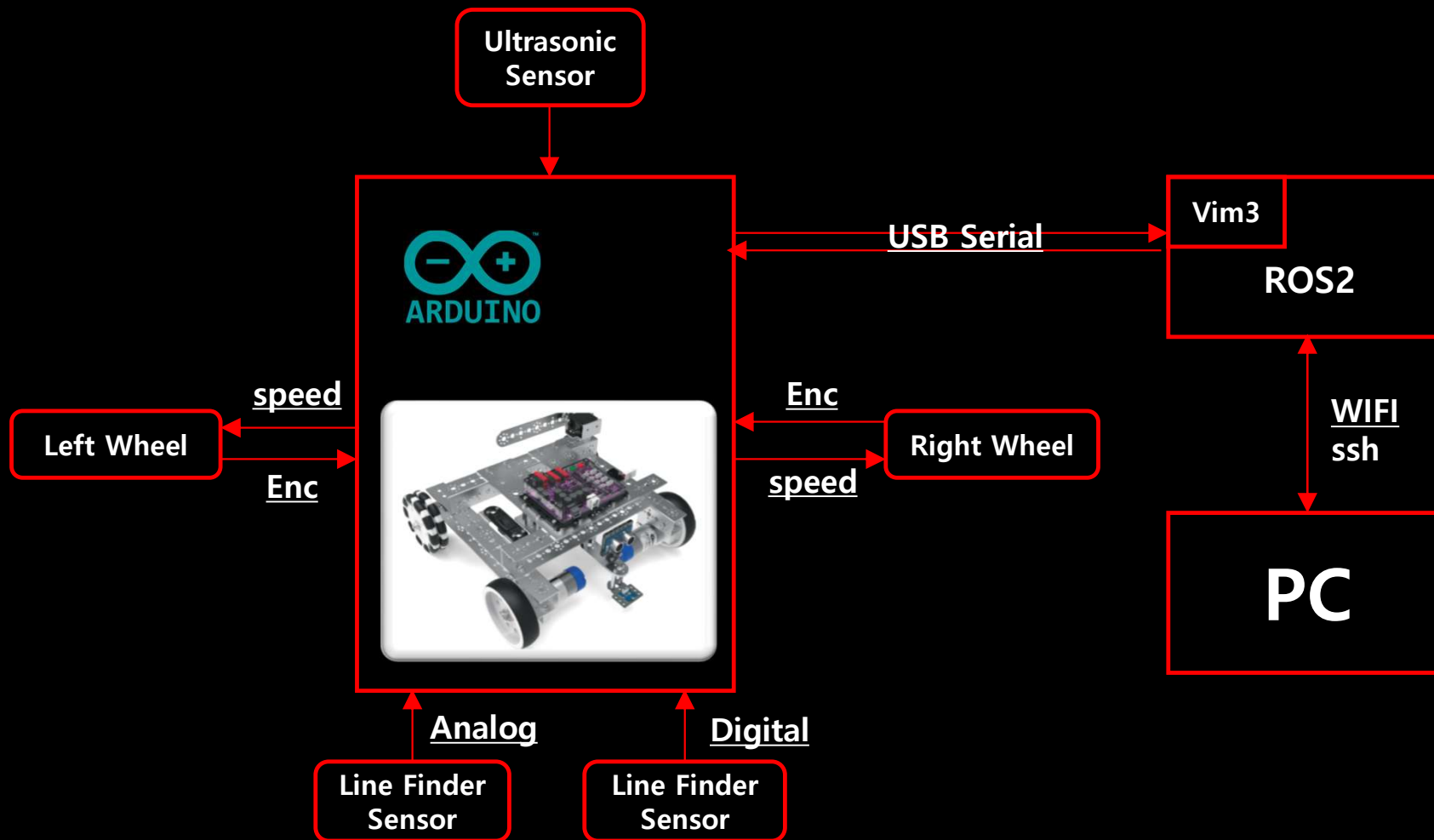
dongilc@mju.ac.kr

2023. 08. 11

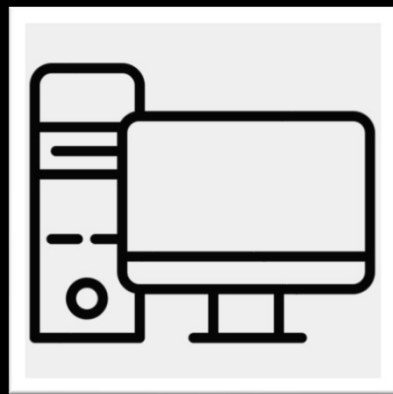
5교시

임베디드 시스템에서의 ROS2 프로그래밍 실습

Tetrix 제어흐름도



Tetrix 원격제어 (PC <-> Embedded Board)



pc

PUB



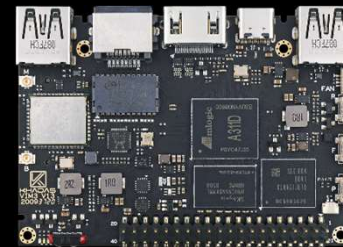
SUB

**ROS2
Topic**

SUB



PUB



Embedded Board

Tetrix 원격제어 (PC <-> Embedded Board)

```
import rclpy
from std_msgs.msg import String
import time

def talker():
    rclpy.init(args=None)
    node = rclpy.create_node('talker')
    publisher = node.create_publisher(String, 'my_topic', 10)
    msg = String()
    msg.data = 'I love MJU'

    while(True):
        print("publish : ",msg.data)
        publisher.publish(msg)
        time.sleep(1)

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

“I love MJU” 라는 값을 publish

PC Program <ros2_tutorial_2023_ch5_1_pc_pub.py>

Tetrix 원격제어 (PC <-> Embedded Board)

```
import rclpy
from std_msgs.msg import String

def callback(msg):
    print("I heard: ", msg.data)

def main(args=None):
    rclpy.init(args=args)
    node = rclpy.create_node('my_subscriber')
    subscription = node.create_subscription(String, 'my_topic', callback, 10)

    try:
        rclpy.spin(node)    # keep the node running
    except KeyboardInterrupt:
        pass

    node.destroy_node()
    rclpy.shutdown()

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

PC에서 보낸 데이터를 수신

Embedded Board Program <ros2_tutorial_2023_ch5_2_embedded_sub.py>

Tetrix 원격제어 (PC <-> Embedded Board)

[illegible]

1

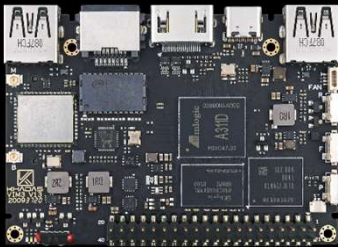
PC

[illegible]

1

Embedded Board

Tetrix 원격제어



VIM3

TX



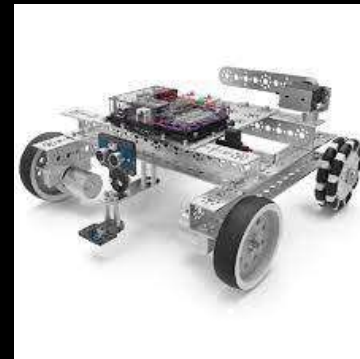
RX

USB
Serial

RX



TX

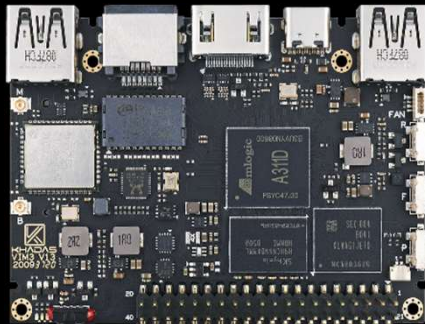


Arduino

lsusb

ls /dev/ttyUSB*

Tetrix 원격제어



VIM3

Data (VIM3 → Arduino)

- Speed L, R
- Gripper close/open
- Gripper up/down

TX



RX

**USB
Serial**

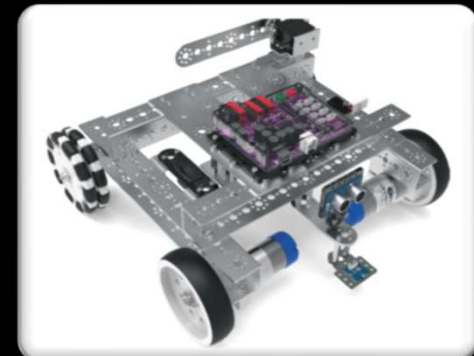
RX



TX

Data (Arduino → VIM3)

- Enc L, R
- LT1
- LT2
- Ultrasonic



Arduino

Tetrix 원격제어 (Embedded Board <-> Tetrix Prizm)

```
import serial

ser = serial.Serial('/dev/ttyUSB0', 115200)

while True:
    data = "I love MJU"

    ser.write(data.encode())
    print("publish data", data)

    response = ser.readline().decode().strip()

    print("Received data:", response)
```

“I love MJU” 라는 값을 publish

Embedded Board Program <ros2_tutorial_2023_ch5_3_embedded_tx.py>

Tetrix 원격제어 (Embedded Board <-> Tetrix Prizm)

```
void setup() {  
  Serial.begin(115200);  
  Serial.setTimeout(10);  
}
```

```
void loop() {  
}
```

Embedded Board에서의 값을 수신하고, 그 값을 다시 송신함

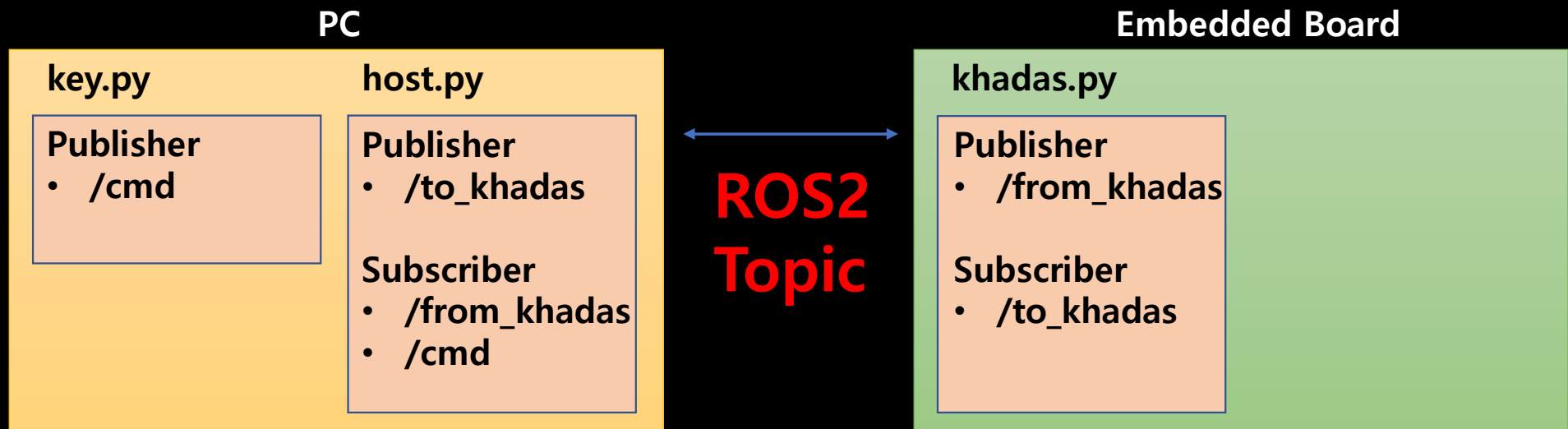
```
void serialEvent(){  
  String data = Serial.readStringUntil('\n');  
  Serial.println(data);  
}
```

Arduino Program <ros2_tutorial_2023_ch5_4_Arduino_loopback.ino>

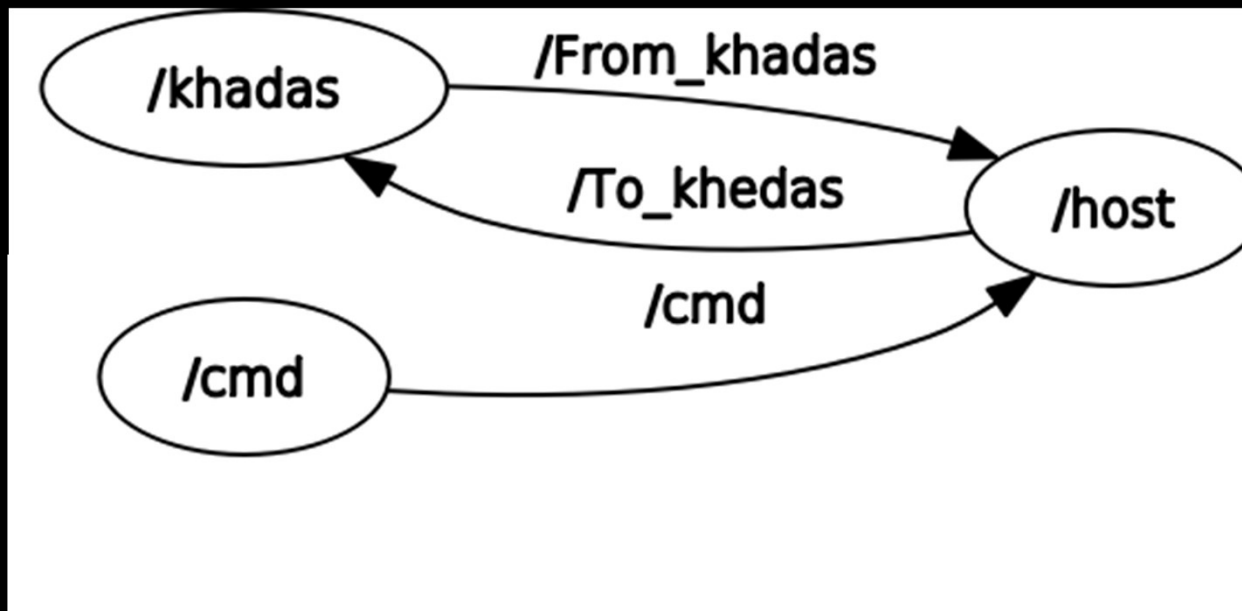
Tetrix 원격제어 (Embedded Board <-> Tetrix Prizm)

[illegible]

Tetrix 원격제어



Tetrix 원격제어



Tetrix 원격제어 (ros2_tutorial_2023_ch5_4_host.py)

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist
from geometry_msgs.msg import Vector3
```

```
class MyNode(Node):
    def __init__(self):
        super().__init__('my_node')
        self.publisher = self.create_publisher(Twist, 'To_khadas', 10)
        self.subscription1 = self.create_subscription(
            Twist,
            'From_key',
            self.listener_callback_twist,
            10)
        self.subscription1 # prevent unused variable warning
        self.subscription2 = self.create_subscription(
            Vector3,
            'From_khadas',
            self.listener_callback_float,
            10)
        self.subscription2 # prevent unused variable warning

        self.subscription2 = self.create_subscription(
            Vector3,
            'From_khadas',
            self.listener_callback_float,
            10)
        self.subscription2 # prevent unused variable warning
```

Tetrix 원격제어 (ros2_tutorial_2023_ch5_4_host.py 이어서)

```
def listener_callback_twist(self, msg):
    self.publisher.publish(msg)

    self.get_logger().info('Received twist message: "%s"' % msg)

def listener_callback_float(self, msg):
    self.get_logger().info('Received float message: "%s"' %
msg.data)

def send_twist(self, linear_x, angular_z):
    msg = Twist()
    msg.linear.x = linear_x
    msg.angular.z = angular_z
    self.publisher.publish(msg)

def main(args=None):
    rclpy.init(args=args)
    node = MyNode()
    node.send_twist(0.5, 0.1)
    rclpy.spin(node)
    node.destroy_node()
    rclpy.shutdown()

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

- cmd 노드에서의 값을 subscribe함
- cmd 노드에서의 값을 다시 publish함
- 임베디드 보드에서 엔코더값을 subscribe함

Tetrix 원격제어 (ros2_tutorial_2023_ch5_5_key.py)

```
import rclpy
import time
from geometry_msgs.msg import Twist
import sys, select, termios, tty
settings = termios.tcgetattr(sys.stdin)

def getKey():
    tty.setraw(sys.stdin.fileno())
    select.select([sys.stdin], [], [], 0)
    key = sys.stdin.read(1)
    termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
    return key
```

Tetrix 원격제어 (ros2_tutorial_2023_ch5_5_key.py 이어서 1)

```
def main(args=None):
    rclpy.init(args=args)
    node = rclpy.create_node('teleop_twist_keyboard')
    pub = node.create_publisher(Twist, 'From_key', 3)
    speed=0
    angle=0

    while(1):
        key = getKey()
        if key == "w":
            print(key)
            # speed=speed+1
            if speed ==360:
                speed=360
            else:
                speed=speed+10
            print(speed)

        if key == "s":
            print(key)
            if speed ==-360:
                speed=360
            else:
                speed=speed-10
            print(speed)
```

```
        if key=="a":
            print(key)
            speed=0
            angle=angle-1
            print(angle)

        if key=="d":
            print(key)
            speed=0
            angle=angle+1
            print(angle)

        if key=="q":
            print(key)
            angle=0.0
            speed=0.0
            print(speed)
            print(angle)

        if key == "e":
            print(key)
            angle=0.0
            speed=0.0
            break
    print(1)
```

Tetrix 원격제어 (ros2_tutorial_2023_ch5_5_key.py 이어서 2)

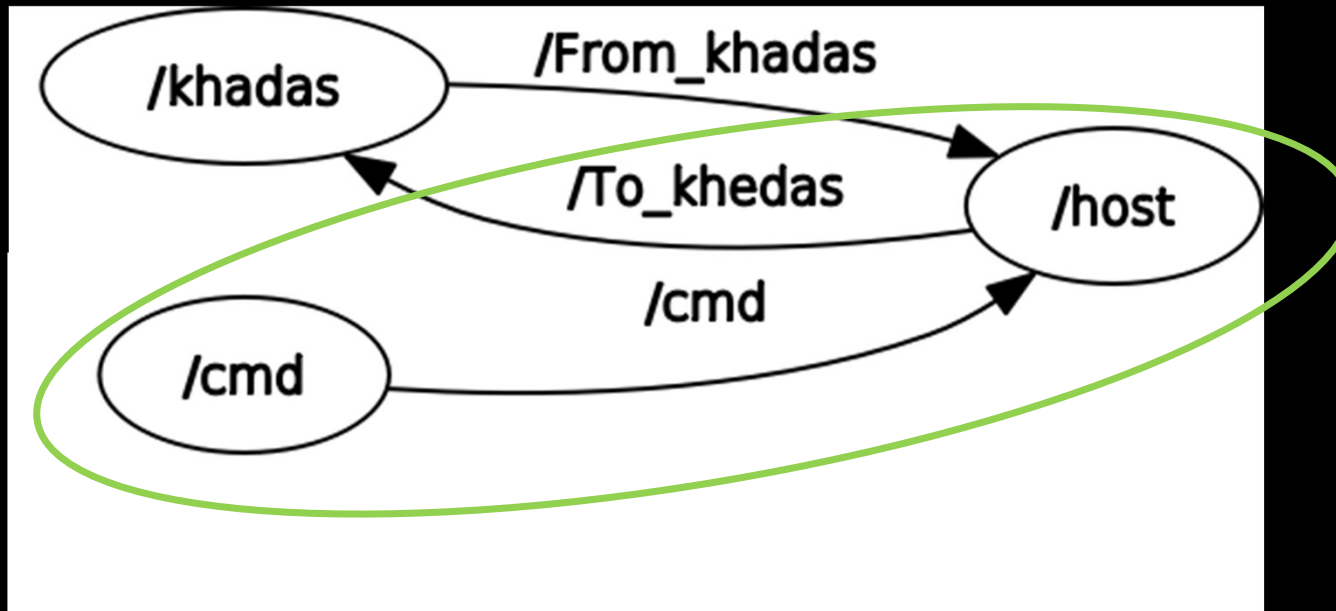
```
twist = Twist()
speed=float(speed)
twist.linear.x = speed; twist.linear.y = 0.0; twist.linear.z = 0.0
twist.angular.x = 0.0; twist.angular.y = 0.0; twist.angular.z = float(angle)
pub.publish(twist)
```

```
termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
```

```
main()
```

- 사용자의 명령을 입력받고, 그 값을 publish함

Tetrix 원격제어



Tetrix 원격제어 (ros2_tutorial_2023_ch5_6_khadas.py)

```
import rclpy
import serial
```

```
from rclpy.node import Node
from geometry_msgs.msg import Twist
from geometry_msgs.msg import Vector3
```

```
class serial_node(Node):

    def __init__(self, port_name):
        super().__init__('serial_node')

        # Subscriber init
        self.subscription = self.create_subscription(Twist,
            'To_khadas',
            self.listener_callback,
            10)

        self.subscription # prevent unused variable warning

        # Init Serial Communication
        self.Serial_ = serial.Serial(
            port=port_name,
            baudrate=115200
        )

        # Publisher init
        self.publisher_ = self.create_publisher(Vector3,
            'From_khadas', 10)

        self.i = 0
```

Tetrix 원격제어 (ros2_tutorial_2023_ch5_6_khadas.py 이어서 1)

```
def listener_callback(self, msg):
    # Pyserial TX (Arudino -> ROS2)
    self.i += 0.3
    robot = Twist()
    enco = Vector3()

    robot.linear.x = msg.linear.x
    robot.angular.z = msg.angular.z

    self.serial_write(robot.linear.x, robot.angular.z)

    data = self.serial_read()
    print(data)
    if data != -1:
        enco.x = float(data[0])
        enco.y = float(data[1])
        print(2)
        self.publisher_.publish(enco)
    else:
        print("err")
```

```
def serial_write(self, data1, data2):

    datafame = '$'+str(data1)+','+str(data2)
    print(datafame)
    self.Serial_.write(datafame.encode())

def serial_read(self):
    response = self.Serial_.readline()
    data = response[:len(response)-2].decode('utf-8')
    print(data)
    data = data.split(',')
    return data
```

Tetrix 원격제어 (ros2_tutorial_2023_ch5_6_khadas.py 이어서 2)

```
def main(args=None):
    rclpy.init(args=args)

    Test_node = serial_node("/dev/ttyUSB0")

    rclpy.spin(Test_node)

    Test_node.destroy_node()
    rclpy.shutdown()

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

감사합니다.