1️⃣ Verify ROS 2 Jazzy is sourced

source /opt/ros/jazzy/setup.bash

(Optional but recommended)

echo "source /opt/ros/jazzy/setup.bash" >> ~/.bashrc

Check:

ros2 --version

2️⃣ Install turtlesim

sudo apt update

sudo apt install ros-jazzy-turtlesim -y

Verify installation:

ros2 pkg list | grep turtlesim

✅ Output should show:

turtlesim

3️⃣ Run turtlesim (Node)

Terminal 1 – Start turtlesim

ros2 run turtlesim turtlesim_node

You should see a **blue window with a turtle** 🐢

4️⃣ Control turtle using keyboard

Terminal 2 – Keyboard teleop

source /opt/ros/jazzy/setup.bash

ros2 run turtlesim turtle_teleop_key

Use keys:

W → forward

S → backward

A → rotate left

D → rotate right

5️⃣ Understand what's happening (Important)

List running nodes

ros2 node list

Output:

/turtlesim

/teleop_turtle

List topics

ros2 topic list

Key topic:

/turtle1/cmd_vel

This is where velocity commands are sent.

6️⃣ Move turtle using ROS 2 topic (manual command)

Move forward

ros2 topic pub /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0}, angular: {z: 0.0}}"

Rotate

ros2 topic pub /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.0}, angular: {z: 2.0}}"

🧠 This is the **same concept you'll use later for ESP8266 robot control**.

7️⃣ Spawn a new turtle

ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.0, y: 5.0, theta: 0.0, name: 'turtle2'}"

Check:

ros2 topic list

You’ll see:

/turtle2/cmd_vel

Control turtle2:

ros2 topic pub /turtle2/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 1.5}, angular: {z: 1.0}}"

8️⃣ Clear screen & reset

Clear drawing

ros2 service call /clear std_srvs/srv/Empty

Reset turtle

ros2 service call /reset std_srvs/srv/Empty

9️⃣ Write a simple Python node (AUTO movement)

Create workspace (if not exists)

mkdir -p ~/ros2_ws/src

cd ~/ros2_ws

Create Python package

cd src

ros2 pkg create turtle_controller --build-type ament_python --dependencies rclpy geometry_msgs

---

turtle_controller/move_turtle.py

```
import rclpy

from rclpy.node import Node

from geometry_msgs.msg import Twist


class TurtleMover(Node):

    def __init__(self):
```

```
        super().__init__('turtle_mover')
        self.publisher_ = self.create_publisher(Twist, '/turtle1/cmd_vel', 10)
        self.timer = self.create_timer(0.5, self.move_turtle)

    def move_turtle(self):
        msg = Twist()
        msg.linear.x = 2.0
        msg.angular.z = 1.0
        self.publisher_.publish(msg)

def main():
    rclpy.init()
    node = TurtleMover()
    rclpy.spin(node)
    rclpy.shutdown()

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

Update setup.py

```
entry_points={
    'console_scripts': [
        'move_turtle = turtle_controller.move_turtle:main',
    ],
},
```

Build & run

```
cd ~/ros2_ws
```

colcon build

source install/setup.bash

ros2 run turtle_controller move_turtle

🐢 Turtle moves automatically in a curve.

🔟 **Mapping this to REAL ROBOT (Important for you)**

Turtlesim	**Real Robot (ESP8266)**
/cmd_vel	Motor speed topic
Twist.linear.x	Forward/backward
Twist.angular.z	Left/right
Teleop	Mobile app / joystick
Python node	Robot control node

👉 **Next logical step** for you:

- Replace turtlesim with **ESP8266 + motors**
- Same /cmd_vel topic
- ESP subscribes via serial/WiFi

⚠️ **Common errors & fixes**

❌ **turtlesim not found**

sudo apt install ros-jazzy-turtlesim

❌ **ros2 command not found**

source /opt/ros/jazzy/setup.bash

❌ **Node not running**

Check:

ros2 node list