# **ROS2 Jazzy Project Documentation**

## **Mohammed Tahrawe - University of Petra**

youtube video link (two videos):

<https://youtu.be/PmV3Epw8c8k><https://youtu.be/86J5PDN31Ds>

## **Table of Contents**

1. [Linux Setup](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#1-linux-setup)
2. [Bash Scripting](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#2-bash-scripting)
3. [TurtleSim Control](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#3-turtlesim-control)
4. [ROS2 Commands](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#4-ros2-commands)
5. [Project Files](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#5-project-files)
6. [Troubleshooting](https://claude.ai/chat/3924b55a-1a54-4adc-be54-4f368eb61463#6-troubleshooting)

## **1. Linux Setup**

### **1.1 Ubuntu Installation**

**Operating System:** Ubuntu 24.04.3 LTS (Noble)

lsb\_release -a  
# Output:  
# Distributor ID: Ubuntu  
# Description: Ubuntu 24.04.3 LTS  
# Release: 24.04  
# Codename: noble

**Screenshot Required:** ✅ (Take screenshot of the above output)

### **1.2 Filesystem Navigation Exercises (10 Exercises)**

#### **Exercise 1: Check Current Directory**

pwd  
# Output: /home/mtahr

#### **Exercise 2: Create Directory Structure**

mkdir -p Mohammed/AI  
# Creates nested directories for ROS2 workspace organization

#### **Exercise 3: Navigate to Directory**

cd mohammed/  
pwd  
# Output: /home/mtahr/mohammed

#### **Exercise 4: Create Files**

touch m.txt  
touch m.py  
ls  
# Output: m.txt m.py

#### **Exercise 5: Edit Python File**

nano m.py  
# Created a simple Python script that prints "hello word"

#### **Exercise 6: Execute Python Script**

python3 m.py  
# Output: hello word

#### **Exercise 7: Rename File**

mv m.txt mohammed.txt  
ls  
# Output: AI m.py mohammed.txt

#### **Exercise 8: Move File to Home Directory**

mv mohammed.txt ~  
cd ~  
ls  
# Output: mohammed Mohammed mohammed.txt ros2\_ws test\_file.txt

#### **Exercise 9: Navigate to ROS2 Workspace**

cd ~/ros2\_ws/  
pwd  
# Output: /home/mtahr/ros2\_ws

#### **Exercise 10: List Workspace Contents**

ls  
# Output: build draw\_shapes.sh install log spawn\_turtle.sh src turtlesim\_art.sh

### **1.3 ROS2 Workspace Directory Structure**

/home/mtahr/ros2\_ws/  
├── build/ # Build artifacts  
├── install/ # Installed packages  
├── log/ # Build and runtime logs  
├── src/ # Source code packages  
├── spawn\_turtle.sh # Turtle spawning script  
├── draw\_shapes.sh # Shape drawing script  
└── turtlesim\_art.sh # Artistic creation script

## **2. Bash Scripting**

### **2.1 Spawn Turtle Script (spawn\_turtle.sh)**

**Features:**

* ✅ Automatically sources ROS2 Jazzy
* ✅ Accepts turtle name and position as arguments
* ✅ Includes comprehensive error handling
* ✅ Supports single and multiple turtle spawning modes

**Usage Examples:**

#### **Single Turtle Spawn**

./spawn\_turtle.sh turtle2 2.0 2.0  
./spawn\_turtle.sh turtle3 8.0 8.0 1.57

**Output:**

[INFO] ROS2 jazzy is already sourced.  
[INFO] TurtleSim node detected.  
[INFO] Spawning turtle 'turtle2' at position (2.0, 2.0) with angle 0.0...  
[INFO] Successfully spawned turtle 'turtle2'!

#### **Multiple Turtles Mode**

./spawn\_turtle.sh --multi

**Output:**

[INFO] Spawning multiple turtles in predefined positions...  
[INFO] Successfully spawned turtle 'turtle2'!  
[INFO] Successfully spawned turtle 'turtle3'!  
[INFO] Successfully spawned turtle 'turtle4'!  
[INFO] Successfully spawned turtle 'turtle5'!  
[INFO] Successfully spawned turtle 'center\_turtle'!  
[INFO] All turtles spawned successfully!

**Script Capabilities:**

* Validates ROS2 environment
* Checks if TurtleSim node is running
* Error handling for missing arguments
* Supports optional theta (angle) parameter
* Multi-turtle spawning with predefined positions

### **2.2 Shape Drawing Script (draw\_shapes.sh)**

**Features:**

* Interactive menu system
* Direct command-line shape selection
* Color-coded shapes
* Error handling

**Usage Examples:**

#### **Interactive Mode**

./draw\_shapes.sh

**Menu Output:**

================================================  
 TurtleSim Shape Drawing Menu  
================================================  
1) Draw Square (RED)  
2) Draw Circle (BLUE)  
3) Draw Star (YELLOW)  
4) Draw Figure-8 (GREEN)  
5) Draw All Shapes  
6) Clear Screen  
7) Exit  
================================================

#### **Direct Command Mode**

./draw\_shapes.sh turtle1 square  
./draw\_shapes.sh turtle1 circle  
./draw\_shapes.sh turtle1 star  
./draw\_shapes.sh turtle1 figure8

**Output:**

[INFO] Starting shape drawing with turtle1  
[INFO] Drawing SQUARE in RED...  
[INFO] Square completed!

### **2.3 Art Creation Script (turtlesim\_art.sh)**

**Features:**

* Multi-turtle coordination
* Complex artistic compositions
* Background color customization
* Detailed progress reporting

**Usage:**

./turtlesim\_art.sh

**Output:**

╔══════════════════════════════════════╗  
 ║ 🎨 TurtleSim Art Creator 🎨 ║  
 ║ Flower Garden 🌸 ║  
 ╚══════════════════════════════════════╝  
  
[INFO] Starting TurtleSim Art Creation...  
[STEP] Setting sky blue background...  
[STEP] Spawning artist turtles...  
[STEP] Drawing sun with sun\_artist  
[STEP] Drawing grass with grass\_artist  
[STEP] Drawing flower with flower1 at (3.0, 7.0)  
[STEP] Drawing butterfly with butterfly  
[INFO] Art creation complete! 🎨  
  
╔════════════════════════════════════════╗  
║ 🌸 Flower Garden Masterpiece 🌸 ║  
║ ║  
║ Features: ║  
║ • Sky blue background ║  
║ • Bright sun with rays ☀️ ║  
║ • Three colorful flowers 🌺 ║  
║ • Green grass 🌱 ║  
║ • Flying butterflies 🦋 ║  
║ • White clouds ☁️ ║  
║ Created with 6 turtles! 🐢 ║  
╚════════════════════════════════════════╝

## **3. TurtleSim Control**

### **3.1 Launch TurtleSim**

ros2 run turtlesim turtlesim\_node

### **3.2 Keyboard Control**

ros2 run turtlesim turtle\_teleop\_key

**Controls:**

* Arrow keys: Move turtle
* Rotation: Left/Right arrows
* Linear movement: Up/Down arrows

### **3.3 Shape Drawing Demonstrations**

#### **✅ Square (RED)**

./draw\_shapes.sh turtle1 square

* Color: Red (R:255, G:0, B:0)
* Pattern: 4 sides with 90-degree turns
* Status: ✅ Completed

#### **✅ Circle (BLUE)**

./draw\_shapes.sh turtle1 circle

* Color: Blue (R:0, G:0, B:255)
* Pattern: Continuous circular motion
* Status: ✅ Completed

#### **✅ Star (YELLOW)**

./draw\_shapes.sh turtle1 star

* Color: Yellow (R:255, G:255, B:0)
* Pattern: 5-pointed star
* Status: ✅ Completed

#### **✅ Figure-8 (GREEN)**

./draw\_shapes.sh turtle1 figure8

* Color: Green (R:0, G:255, B:0)
* Pattern: Infinity symbol/figure-8
* Status: ✅ Completed

### **3.4 Pen Color Changes**

**Attempted Commands:**

ros2 service call /turtle1/set\_pen turtlesim/srv/SetPen "{r:255, g:0, b:0, width:3, off:0}"  
# Error: Failed to populate field: 'SetPen\_Request' object has no attribute 'r:255'  
  
ros2 service call /turtle1/set\_pen turtlesim/srv/SetPen "{r: 255, g: 0, b: 0, width: 3, off: 0}"  
# Error: Failed to populate field: attribute name must be string, not 'bool'

**Solution Implemented:** Pen color changes are integrated directly into the shape drawing scripts, which use proper ROS2 service call formatting programmatically.

## **4. ROS2 Commands**

### **4.1 Services Demonstration**

#### **Spawn Service**

ros2 service call /spawn turtlesim/srv/Spawn "{x: 2.0, y: 2.0, theta: 0.0, name: 'turtle2'}"

**Response:**

turtlesim.srv.Spawn\_Response(name='turtle2')

#### **Set Pen Service**

ros2 service call /turtle1/set\_pen turtlesim/srv/SetPen \  
 "{r: 255, g: 0, b: 0, width: 3, off: 0}"

### **4.2 Multiple Turtle Spawning (✅ 6 Turtles)**

#### **Spawned Turtles:**

1. **turtle2** - Position: (2.0, 2.0, 0.0)
2. **turtle3** - Position: (9.0, 2.0, 0.0) / (8.0, 8.0, 1.57)
3. **turtle4** - Position: (2.0, 9.0, 0.0)
4. **turtle5** - Position: (9.0, 9.0, 0.0)
5. **center\_turtle** - Position: (5.5, 5.5, 0.0)
6. **turtle1** - Default (original turtle)

**Command Used:**

./spawn\_turtle.sh --multi

### **4.3 Background Color Change**

**Implementation:** Background color changing is integrated into turtlesim\_art.sh script:

[STEP] Setting sky blue background...  
ros2 param set /turtlesim background\_r 135  
ros2 param set /turtlesim background\_g 206  
ros2 param set /turtlesim background\_b 235

**Custom Color:** Sky Blue (R:135, G:206, B:235)

### **4.4 Topics Demonstration**

#### **List Active Topics**

ros2 topic list

**Expected Output:**

/parameter\_events  
/rosout  
/turtle1/cmd\_vel  
/turtle1/color\_sensor  
/turtle1/pose

#### **Monitor Turtle Pose**

ros2 topic echo /turtle1/pose

#### **Publish Velocity Commands**

ros2 topic pub /turtle1/cmd\_vel geometry\_msgs/msg/Twist \  
 "{linear: {x: 2.0}, angular: {z: 1.8}}"

### **4.5 Parameters Demonstration**

#### **List Parameters**

ros2 param list

#### **Get Background Color**

ros2 param get /turtlesim background\_r  
ros2 param get /turtlesim background\_g  
ros2 param get /turtlesim background\_b

#### **Set Background Color**

ros2 param set /turtlesim background\_r 135  
ros2 param set /turtlesim background\_g 206  
ros2 param set /turtlesim background\_b 235

## **5. Project Files**

### **5.1 File Permissions**

chmod +x spawn\_turtle.sh  
chmod +x draw\_shapes.sh  
chmod +x turtlesim\_art.sh

### **5.2 Workspace Structure**

~/ros2\_ws/  
├── spawn\_turtle.sh # Turtle spawning automation  
├── draw\_shapes.sh # Shape drawing with color control  
├── turtlesim\_art.sh # Complex artistic creations  
├── src/ # Source packages (if any)  
├── build/ # Compiled artifacts  
├── install/ # Installation files  
└── log/ # Operation logs

## **6. Troubleshooting**

### **6.1 Common Issues**

#### **Issue 1: Service Call Syntax Error**

**Problem:**

Failed to populate field: 'SetPen\_Request' object has no attribute 'r:255'

**Solution:** Use proper YAML syntax in bash scripts with correct spacing and quoting.

#### **Issue 2: Turtle Already Exists**

**Problem:**

response: turtlesim.srv.Spawn\_Response(name='')

**Solution:** Turtle with that name already exists. Use different name or clear TurtleSim first.

### **6.2 Best Practices**

1. **Always source ROS2:**

source /opt/ros/jazzy/setup.bash

1. **Check TurtleSim is running:**

ros2 node list | grep turtlesim

1. **Use error handling in scripts**
2. **Clear screen between demonstrations:**

ros2 service call /clear std\_srvs/srv/Empty