🤖 Robot Controller Demo - Quick Start

What This Is

A **hardware-free** demo version to test the web interface and wireless communication before connecting to actual GPIO pins. Think of it as a "flight simulator" for your robot!

What It Does

✅ **Simulates all robot movements** with realistic timing
✅ **Random obstacle detection** (5% chance per movement)
✅ **Live position tracking** with visual trail
✅ **Real-time sensor readings** (simulated distances)
✅ **Statistics tracking** (distance, rotations, commands)
✅ **WebSocket communication** (same as real version)

Quick Setup (3 Steps)

1. Install Dependencies

```bash
pip3 install flask flask-socketio python-socketio eventlet
```

2. Create File Structure

```bash
# Create directory
mkdir robot_demo
cd robot_demo

# Create templates folder
mkdir templates

# Save files:
# - demo_controller.py (main folder)
# - demo.html (inside templates folder)
```

3. Run It!

```bash
python3 demo_controller.py
```

Access the Demo

On the same computer:

```
http://localhost:5000
```

From your phone/tablet (on same WiFi):

```
http://YOUR_PI_IP:5000
```

Find your IP:

```bash
hostname -I
# Example output: 192.168.1.105
# Then use: http://192.168.1.105:5000
```

What You'll See

Left Panel - Control Interface

- 🎮 **Direction pad** - Arrow buttons for movement
- 🎚️ **Slider** - Adjust distance/angle (0.5 to 3.0)
- 💃 **Action buttons** - Dance, Hi, Stop
- 📡 **Sensor readings** - Front/rear distances
- ⌨️ **Keyboard controls** - Arrow keys or WASD

Right Panel - Visualization

- 📊 **Live map** - Watch robot move in 2D space
- 🎨 **Trail tracking** - See path history
- 📈 **Statistics** - Position, distance, rotations
- 🔄 **Reset button** - Start over

Testing Scenarios

Test 1: Basic Movement ✅

1. Click UP arrow (forward)
   → Watch robot move on map
   → See distance counter increase

2. Adjust slider to 2.0
3. Click LEFT arrow
   → Robot turns 2.0 degrees
   → See angle change

4. Try keyboard: W, A, S, D keys

Test 2: Obstacle Detection ⚠️

1. Keep clicking forward
2. Eventually you'll hit a "simulated obstacle"
3. Watch for orange warning message
4. Robot stops automatically
5. Check logs for obstacle detection

Test 3: Complex Moves 💃

1. Click "Dance" button
   → Forward → Back → Turns
   → All movements tracked on map

2. Click "Say Hi" button
   → Left → Right → Center

Test 4: Multi-Device Control 📱

1. Open demo on computer browser
2. Open same URL on phone (same WiFi)
3. Both can control robot
4. Both see same position updates
5. Commands queue if sent simultaneously

Test 5: Wireless Communication Speed 🚀

1. Watch the console logs
2. Notice timestamps on commands
3. Typical response: <50ms
4. Much faster than Google Sheets polling!

What to Watch For

✅ Working Correctly:

- Status shows "Connected - Demo Mode"
- Buttons respond when clicked
- Robot animates smoothly on canvas
- Distance readings change every second
- Statistics update in real-time
- Keyboard shortcuts work

❌ Common Issues:

Can't access from phone:

```bash
# Check firewall
sudo ufw allow 5000

# OR disable temporarily
sudo ufw disable
```

Port already in use:

```bash
# Find what's using port 5000
sudo lsof -i :5000

# Kill it
sudo kill -9 <PID>
```

Module not found:

```bash
# Reinstall dependencies
pip3 install --upgrade flask flask-socketio python-socketio eventlet
```

Understanding the Simulation

Movement Timing

- **Linear speed**: 0.7 m/s (same as your real robot)
- **Rotation speed**: 270°/s (same as your real robot)
- **Example**: Moving 1.4 meters takes 2 seconds (1.4 / 0.7)

Obstacle Detection

- **5% random chance** per movement check
- Simulates ultrasonic sensor behavior
- Returns distances between 5-200 cm
- Triggers emergency stop if <20cm

Position Tracking

- Starts at (0, 0) facing up (0°)
- Updates 10 times per movement
- Tracks X, Y coordinates in meters
- Tracks angle in degrees (0-360)

Console Output Examples

Successful Command:

```
2025-01-09 10:30:15 - INFO - 📨 Received command: {'type': 'forward', 'value': 1.0}
2025-01-09 10:30:15 - INFO - 🚗 Moving forward 1.0m
2025-01-09 10:30:16 - INFO - ✅ Command executed successfully
```

Obstacle Detected:

```
2025-01-09 10:31:20 - INFO - 🚗 Moving forward 2.0m
2025-01-09 10:31:21 - WARNING - ⚠️ Obstacle detected! Stopping.
```

Client Connection:

```
2025-01-09 10:29:00 - INFO - ✅ Client connected
2025-01-09 10:35:00 - INFO - ❌ Client disconnected
```

Performance Comparison

Feature	Google Sheets	Demo (WebSockets)
Command Latency	1-2 seconds	<50ms (40x faster!)
Real-time Updates	No	Yes
Position Tracking	No	Yes
Visual Feedback	No	Yes
Multi-device	Yes	Yes
Internet Required	Yes	No

Next Steps After Testing

✅ If demo works perfectly:

1. **Set up hotspot** (from SETUP_GUIDE.md)
2. **Replace demo_controller.py** with web_controller.py
3. **Use index.html** instead of demo.html
4. **Connect GPIO** according to your pin configuration
5. **Run real robot!**

🔧 Modifications You Can Make:

Change speeds:

```python
# In demo_controller.py
self.speed = 0.7  # Change to 1.0 for faster
self.rot_speed = 270  # Change to 360 for faster turns
```

Change obstacle probability:

```python
if random.random() < 0.05:  # Change 0.05 to 0.10 for 10% chance
```

Change distance range:

```python
base_distance = random.uniform(25, 200)  # Adjust min/max
```

Troubleshooting Tips

Demo runs but no movement on canvas?

- Check browser console (F12)
- Look for JavaScript errors
- Try different browser (Chrome recommended)

Can't connect from phone?

- Verify both devices on same WiFi
- Check Pi's IP: `hostname -I`
- Try disabling firewall temporarily
- Use IP address, not hostname

Buttons don't respond?

- Check if command is executing (status bar)
- Look for errors in terminal
- Refresh browser page
- Check console logs

Learning Points

This demo teaches you:

1. **WebSocket Communication** - Real-time bidirectional data
2. **Flask Web Server** - Serving pages and handling requests
3. **Threading** - Running robot commands without blocking
4. **Canvas Animation** - Visualizing robot state
5. **Event-Driven Programming** - Responding to user actions

Files You Created

```
robot_demo/
├── demo_controller.py      # Main server (this file)
├── templates/
│   └── demo.html           # Web interface
└── demo_controller.log     # Automatically generated logs
```

Compare to Your Real Robot

Component	Demo Version	Real Version
GPIO	Simulated	RPi.GPIO
Motors	Virtual	PWM control
Sensors	Random values	Ultrasonic
Display	None	LCD with GIFs
Movement	Calculated	Physical
Communication	**SAME**	**SAME**

The **communication layer is identical** - that's what we're testing!

Ready for Real Robot?

Once this demo works smoothly:

- ✅ Web interface loads
- ✅ Commands execute
- ✅ Statistics update
- ✅ Accessible from phone
- ✅ No lag or errors

You're ready to deploy on the actual robot! The wireless communication is proven to work.

Questions? Issues? Check the logs!

bash

```
# View live logs
tail -f demo_controller.log

# Search for errors
grep ERROR demo_controller.log
```