Complete Project Astra System Deployment Guide

System Architecture

- **Windows UGV-Server (AWS EC2)**: Dashboard server and data receiver
- **Ubuntu Rover (rover-PC)**: Sensor data collection and transmission
- **Network**: ZeroTier rovernet (4753CF475F287023)

Phase 1: Windows UGV-Server Setup

🖥️ MACHINE: Windows UGV-Server (AWS EC2)

Start Dashboard Server

```powershell
# Navigate to project directory
cd C:\Users\Administrator\Documents\GitHub\AstraBackup\Harry\masterscriptsv4

# Start the dashboard server
python dashboard_server_rovernet.py
```

Expected Output:

```
Project Astra NZ Rovernet Dashboard Server
==========================================
Network: rovernet (4753CF475F287023)
Server IP: 172.25.77.186:8080
Waiting for rover data...
INFO: Uvicorn running on http://0.0.0.0:8080
```

Verify Dashboard Access

```powershell
# Test dashboard locally on Windows server
Invoke-WebRequest -Uri "http://172.25.77.186:8080"
```

Dashboard URL: http://172.25.77.186:8080

Phase 2: Ubuntu Rover Network Setup

🤖 MACHINE: Ubuntu Rover (rover-PC)

Join ZeroTier Network

```bash
# Leave any existing wrong network
sudo zerotier-cli leave 41d49af6c276269e

# Join the correct rovernet
sudo zerotier-cli join 4753CF475F287023

# Verify network status
sudo zerotier-cli listnetworks
```

Authorize Rover in ZeroTier Central

🌐 BROWSER ACTION (any machine):

1. Go to ZeroTier Central: https://my.zerotier.com
2. Navigate to network: `4753CF475F287023`
3. Refresh the page
4. Find the new rover device
5. Check the "Auth" checkbox
6. Note the IP address assigned to rover

Test Network Connection

```bash
# Test connection to Windows server
ping 172.25.77.186

# Test HTTP connectivity (Linux curl syntax)
curl -X GET http://172.25.77.186:8080/api/status
```

Phase 3: Ubuntu Rover Camera Preparation

🤖 MACHINE: Ubuntu Rover (rover-PC)

Stop Conflicting Processes

```bash
# Kill any competing RealSense processes
sudo pkill -f realsense
sudo pkill -f rs-
sudo pkill -f python3

# Wait for processes to fully stop
sleep 3
```

Verify Camera Hardware

```bash
# Check if RealSense is detected
lsusb | grep Intel

# Expected output:
# Bus 001 Device 003: Intel Corp. Intel(R) RealSense(TM) Depth Camera 435i
```

Test Camera Access

```bash
# Navigate to project directory
cd ~/harry/AstraBackup/Harry/masterscriptsv4/

# Activate the correct virtual environment
source ~/rover_venv/bin/activate

# Test RealSense library
python3 -c "import pyrealsense2 as rs; print('RealSense version:', rs.__version__)"
```

Phase 4: Ubuntu Rover System Startup

🤖 MACHINE: Ubuntu Rover (rover-PC)

Terminal 1: Hardware Validation

```bash
```

```
# Navigate to project directory
cd ~/harry/AstraBackup/Harry/masterscriptsv4/

# Activate virtual environment
source ~/rover_venv/bin/activate

# Run hardware check
python3 hardware_check_v4.py
```

Expected Output:

```
✅ RPLidar: Connected
✅ RealSense Camera: Connected
✅ Pixhawk: Connected
All sensors initialized successfully
```

Terminal 2: Proximity System

```
bash

# Open new terminal
cd ~/harry/AstraBackup/Harry/masterscriptsv4/

# Activate virtual environment
source ~/rover_venv/bin/activate

# Start proximity system
python3 combo_proximity_bridge_fixed_v4.py
```

Expected Output:

```
RPLidar + RealSense + Pixhawk data flowing
Mission Planner connection available on UDP:14550
8-sector proximity data active
```

Terminal 3: Data Relay to Dashboard

```
bash
```

```
# Open new terminal
cd ~/harry/AstraBackup/Harry/masterscriptsv4/

# Activate virtual environment
source ~/rover_venv/bin/activate

# Start data relay
python3 rover_data_relay.py
```

Expected Output:

```
Connecting to dashboard server: 172.25.77.186:8080
✅ Dashboard connection established
📡 Transmitting sensor data...
🎥 Camera feed active
📊 Telemetry data flowing
```

Phase 5: System Verification

🖥️ MACHINE: Windows UGV-Server - Dashboard Check

```
powershell

# Verify dashboard is receiving data
Invoke-WebRequest -Uri "http://172.25.77.186:8080/api/status"
```

Browser Verification:

- Open: http://172.25.77.186:8080
- Should show:
  - ✅ Live LiDAR radar display
  - ✅ RealSense camera feed
  - ✅ Telemetry data updates
  - ✅ Real-time sensor status

🤖 MACHINE: Ubuntu Rover - Mission Planner (Optional)

```
bash

# Mission Planner connection available at:
# UDP: [rover-ip]:14550
# Enable proximity display: Ctrl+F → Proximity
```

Troubleshooting Guide

🤖 UBUNTU ROVER - Camera Issues

If No Camera Feed on Dashboard:

```bash
# Check RealSense detailed status
cd ~/harry/AstraBackup/Harry/masterscriptsv4/
source ~/rover_venv/bin/activate

python3 -c "
import pyrealsense2 as rs
try:
    ctx = rs.context()
    devices = ctx.query_devices()
    print(f'Found {len(devices)} RealSense devices')
    for i, dev in enumerate(devices):
        print(f'Device {i}: {dev.get_info(rs.camera_info.name)}')
        print(f'Serial: {dev.get_info(rs.camera_info.serial_number)}')
except Exception as e:
    print(f'RealSense error: {e}')
"
```

Camera Reset Procedure:

```bash
# Emergency camera reset
sudo modprobe -r uvcvideo
sudo modprobe uvcvideo
sleep 5

# Restart camera processes
sudo pkill -f python3
cd ~/harry/AstraBackup/Harry/masterscriptsv4/
source ~/rover_venv/bin/activate
python3 hardware_check_v4.py
```

🖥️ WINDOWS SERVER - Network Issues

Test Dashboard Server:

```powershell
```

```
# Check if server is running
netstat -an | findstr :8080

# Test local access
curl http://localhost:8080/api/status

# Test external access
curl http://172.25.77.186:8080/api/status
```

🌐 BOTH MACHINES - ZeroTier Issues

Check ZeroTier Status:

```
bash

# Ubuntu Rover
sudo zerotier-cli info
sudo zerotier-cli listnetworks

# Check IP assignment
ip addr show zt[tab]
```

```
powershell

# Windows Server
zerotier-cli info
zerotier-cli listnetworks

# Check IP assignment
ipconfig | findstr "172.25"
```

Quick Start Commands Summary

🖥️ Windows UGV-Server:

```
powershell

cd C:\Users\Administrator\Documents\GitHub\AstraBackup\Harry\masterscriptsv4
python dashboard_server_rovernet.py
```

🤖 Ubuntu Rover (3 terminals):

```
bash
```

```
# Terminal 1 - Hardware Check
cd ~/harry/AstraBackup/Harry/masterscriptsv4/
source ~/rover_venv/bin/activate
python3 hardware_check_v4.py

# Terminal 2 - Proximity System
cd ~/harry/AstraBackup/Harry/masterscriptsv4/
source ~/rover_venv/bin/activate
python3 combo_proximity_bridge_fixed_v4.py

# Terminal 3 - Data Relay
cd ~/harry/AstraBackup/Harry/masterscriptsv4/
source ~/rover_venv/bin/activate
python3 rover_data_relay.py
```

Startup Sequence Checklist

- □ **Step 1**: Start Windows dashboard server
- □ **Step 2**: Authorize rover in ZeroTier Central
- □ **Step 3**: Test network connectivity (ping both directions)
- □ **Step 4**: Kill competing processes on Ubuntu rover
- □ **Step 5**: Run hardware check on Ubuntu rover
- □ **Step 6**: Start proximity system on Ubuntu rover
- □ **Step 7**: Start data relay on Ubuntu rover
- □ **Step 8**: Verify dashboard shows all data including camera feed

Success Criteria: Dashboard at http://172.25.77.186:8080 shows live camera feed, LiDAR data, and telemetry updates.