Complete Deployment Guide - OnStepX Alpaca Bridge

Project Complete!

Your Alpaca server now has **ALL planned features** implemented:

Feature	Status	Hardware	Mock
Telescope	✓ Complete	OnStepX (Network/USB)	N/A
UDP Discovery	✓ Complete	Port 32227	N/A
Camera (ZWO)	✓ Complete	ASI Cameras	N/A
Camera (ToupTek)	☑ Complete	ToupTek Cameras	N/A
FilterWheel	☑ Complete	ZWO EFW	✓ Available
Focuser	☑ Complete	ZWO EAF	✓ Available

Omplete File Structure

```
alpaca-onstepx/
                          # Configuration (devices, ports, settings)
— config.py
— alpaca_helpers.py
                              # Helper functions
                          # Flask server + all API routes
 — main.py
  — alpaca_discovery.py
                               # UDP discovery service
                           # OnStepX mount driver (network + USB)
 — telescope.py
                              # ZWO ASI camera driver
  — camera_zwo.py
  — camera_touptek.py
                               # ToupTek camera driver
  – filterwheel.py
                            # ZWO EFW + mock filter wheel
                          # ZWO EAF + mock focuser
  — focuser.py
  — test_telescope_connection.py
                                   # Test telescope connectivity
test_discovery.py
                             # Test UDP discovery
  test_filterwheel_focuser.py
                                 # Test filter wheel & focuser
                         # Python virtual environment
   - venv/
                              # Python dependencies
  – requirements.txt
  — onstepx-alpaca.service
                                # Systemd service file
```

Step-by-Step Deployment

Step 1: Prepare the Raspberry Pi

```
bash

# Update system
sudo apt update
sudo apt upgrade -y

# Install dependencies
sudo apt install python3-pip python3-venv -y

# Create project directory
mkdir -p ~/alpaca-onstepx
cd ~/alpaca-onstepx
```

Step 2: Copy All Files

Transfer all your files to the Pi:

```
# From your development machine:
scp *.py ubuntu@raspberrypi:~/alpaca-onstepx/
# Or use git if you have a repository
```

Step 3: Setup Virtual Environment

```
bash

cd ~/alpaca-onstepx

# Create virtual environment

python3 -m venv venv

# Activate it

source venv/bin/activate

# Install dependencies

pip install flask requests pyserial
```

Step 4: Configure Your Setup

Edit (config.py):

```
python
# 1. SET YOUR TELESCOPE IP
TELESCOPE_CONFIG = {
  'connection_type': 'network',
  'network': {
    'host': '192.168.1.XXX', # ← YOUR ONSTEPX IP
    'port': 9999,
  },
}
# 2. ENABLE YOUR DEVICES
DEVICES = {
  'telescope': {'enabled': True, ...},
  'camera_zwo': {'enabled': True, ...}, # If you have it
  'camera_touptek': {'enabled': True, ...}, # If you have it
  'filterwheel': {'enabled': True, ...}, # Enable
  'focuser': {'enabled': True, ...},
                                  # Enable
}
# 3. SET FILTERWHEEL MODE
FILTERWHEEL_CONFIG = {
  'mode': 'auto', # 'auto' = use ZWO if available, mock if not
  # Customize your filter names:
  'filter_names': [
    "Red", "Green", "Blue", "Luminance",
    "H-Alpha", "OIII", "SII", "Clear"
  ],
  # Set focus offsets (adjust for YOUR filters):
  'focus_offsets': [0, 0, 0, 0, 50, 30, 40, 0],
}
# 4. SET FOCUSER MODE
FOCUSER_CONFIG = {
  'mode': 'auto', # 'auto' = use ZWO if available, mock if not
}
```

Step 5: Configure Firewall

```
bash

# Allow Alpaca HTTP server
sudo ufw allow 5555/tcp

# Allow UDP discovery
sudo ufw allow 32227/udp

# Verify
sudo ufw status
```

Step 6: Test Everything

```
cd ~/alpaca-onstepx
source venv/bin/activate

# Test telescope connection
python3 test_telescope_connection.py network 192.168.1.XXX

# Test discovery
python3 test_discovery.py

# Test filter wheel and focuser
python3 test_filterwheel_focuser.py both auto
```

Step 7: Run the Server

bash
python3 main.py

Expected Output:

OnStepX Alpaca Bridge - Complete Server with Discovery
Initializing devices
[Telescope] Configured for NETWORK: 192.168.1.100:9999
✓ Telescope initialized
✓ ZWO camera initialized
✓ ToupTek camera initialized
Toup Tex carriera mittanzea
[FilterWheel] Mode: auto
✓ Filter wheel initialized
Slots: 8 Filters: Pad Creen Plus Luminance II Alpha OIII SII Clear
Filters: Red, Green, Blue, Luminance, H-Alpha, OIII, SII, Clear
[Focuser] Mode: auto
✓ Focuser initialized
Max position: 100000 steps
Step size: 1.0 microns
Device initialization complete!
=======================================
Starting UDP Discovery Service
✓ Discovery service running on UDP port 32227
Clients can now auto-discover this server!
HTTP Server starting
Host: 0.0.0.0 Port: 5555
1011.0000
Access from network: http:// <pi-ip>:5555</pi-ip>
N.I.N.A. should now auto-discover this server!
=======================================
Press Ctrl+C to stop
Tress Sar 6 to stop
* Serving Flask app 'main'
* Running on http://0.0.0.0:5555

1. Open N.I.N.A.

2. Equipment → Telescope

- Choose "ASCOM Alpaca"
- Server should auto-discover! 🧩
- Select it and click "Connect"

3. Equipment → Camera

- Choose "ASCOM Alpaca"
- Select camera and connect

4. Equipment → Filter Wheel

- Choose "ASCOM Alpaca"
- Select filter wheel and connect

5. Equipment → Focuser

- Choose "ASCOM Alpaca"
- Select focuser and connect

All devices should connect and work! 🎉





Setup as Systemd Service (Auto-Start)

Create Service File

bash

sudo nano /etc/systemd/system/onstepx-alpaca.service

ini

[Unit]

Description=OnStepX Alpaca Bridge Server

After=network.target

[Service]

Type=simple

User=ubuntu

WorkingDirectory=/home/ubuntu/alpaca-onstepx

Environment="PATH=/home/ubuntu/alpaca-onstepx/venv/bin"

ExecStart=/home/ubuntu/alpaca-onstepx/venv/bin/python3 /home/ubuntu/alpaca-onstepx/main.py

Restart=always

RestartSec=10

[Install]

WantedBy=multi-user.target

Enable and Start Service

```
# Reload systemd
sudo systemctl daemon-reload

# Enable service (start on boot)
sudo systemctl enable onstepx-alpaca

# Start service now
sudo systemctl start onstepx-alpaca

# Check status
sudo systemctl status onstepx-alpaca

# View logs
sudo journalctl -u onstepx-alpaca -f
```

Service Management Commands

```
bash
# Start
sudo systemctl start onstepx-alpaca
# Stop
sudo systemctl stop onstepx-alpaca
# Restart
sudo systemctl restart onstepx-alpaca
# Status
sudo systemctl status onstepx-alpaca
# Logs (last 100 lines)
sudo journalctl -u onstepx-alpaca -n 100
# Follow logs
sudo journalctl -u onstepx-alpaca -f
```

Configuration Quick Reference

Mock Mode (Testing Without Hardware)

```
python

# config.py
FILTERWHEEL_CONFIG = {'mode': 'mock', ...}
FOCUSER_CONFIG = {'mode': 'mock', ...}
```

Auto Mode (Use Hardware if Available)

```
python

# config.py

FILTERWHEEL_CONFIG = {'mode': 'auto', ...}

FOCUSER_CONFIG = {'mode': 'auto', ...}
```

Hardware Only Mode (Error if Not Found)

```
python

# config.py

FILTERWHEEL_CONFIG = {'mode': 'zwo', ...}

FOCUSER_CONFIG = {'mode': 'zwo', ...}
```

Network Telescope

```
python

# config.py

TELESCOPE_CONFIG = {
    'connection_type': 'network',
    'network': {'host': '192.168.1.100', 'port': 9999},
}
```

USB Telescope

```
python

# config.py

TELESCOPE_CONFIG = {
    'connection_type': 'serial',
    'serial': {'port': '/dev/ttyUSB0', 'baudrate': 9600},
}
```

1 N.I.N.A. Integration Checklist

Equipment Setup:

☐ Telescope connects and shows coordinates
Camera(s) connect and can take exposures
☐ Filter wheel connects and can change filters
☐ Focuser connects and can move
Sequencing Features:
☐ Auto-focus works with focuser
☐ Filter changes work in sequences
Focus offsets apply when changing filters
☐ Temperature compensation (if enabled)
Advanced Features:
☐ Meridian flip works with telescope
☐ PHD2 guiding works (if telescope supports)
Platesolving works with camera
Multi-filter sequences work
Troubleshooting Guide
Issue: "No devices found" in N.I.N.A. Solution 1: Check discovery
Solution 1: Check discovery
Solution 1: Check discovery
Solution 1: Check discovery bash
bash python3 test_discovery.py
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry • In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555
bash python3 test_discovery.py Solution 2: Manual entry In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry • In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry • In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall bash sudo ufw allow 32227/udp
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry • In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall bash sudo ufw allow 32227/udp
Solution 1: Check discovery bash python3 test_discovery.py Solution 2: Manual entry • In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall bash sudo ufw allow 32227/udp sudo ufw allow 5555/tcp
bash python3 test_discovery.py Solution 2: Manual entry In N.I.N.A., enter server IP manually: 192.168.1.XXX:5555 Solution 3: Check firewall bash sudo ufw allow 32227/udp sudo ufw allow 5555/tcp Issue: Telescope won't connect

Check 2: Verify IP address

bash
ping 192.168.1.XXX

Check 3: Check OnStepX web interface

http://192.168.1.XXX

Issue: Filter wheel or focuser not working

Check 1: Verify mode in config

python
If you don't have hardware, use mock:
FILTERWHEEL_CONFIG = {'mode': 'mock', ...}

Check 2: Test independently

bash

python3 test_filterwheel_focuser.py both auto

Check 3: Check hardware detection

bash

python3 test_filterwheel_focuser.py detect

Issue: Server won't start

Check 1: Port already in use?

bash
sudo netstat -tulpn | grep 5555

Check 2: Python errors?

bash
Check imports
python3 -c "import flask"
python3 -c "import config"

Check 3: Check logs

bash sudo journalctl -u onstepx-alpaca -n 50

Issue: Service won't start on boot

Check 1: Is it enabled?

bash
sudo systemctl is-enabled onstepx-alpaca

Check 2: Check for errors

bashsudo systemctl status onstepx-alpacasudo journalctl -u onstepx-alpaca --since today

📊 Performance & Monitoring

Check Server Status

```
bash

# CPU usage
top -p $(pgrep -f main.py)

# Memory usage
ps aux | grep python3

# Network connections
sudo netstat -tulpn | grep python3

# Log file size
ls -lh /var/log/onstepx-alpaca.log
```

Expected Performance

Metric	Expected Value
CPU Usage	< 5%
Memory	~50-100 MB
HTTP Response	< 50ms
UDP Discovery	< 10ms
Filter Change	1-2 seconds
Focuser Move (10k steps)	10-15 seconds

Monitoring Commands

```
bash
# Watch logs live
sudo journalctl -u onstepx-alpaca -f
# Check service status
watch -n 5 'sudo systemctl status onstepx-alpaca'
# Monitor network
sudo iftop -i eth0
```

Recurity Considerations

Current Status

- V Discovery is read-only (safe)
- 1 No authentication required
- All commands accepted

Recommendations

For Home Network (Private WiFi):

- Current setup is fine
- Use firewall to restrict access

For Public/Shared Network:

- Add authentication layer
- Use VPN for remote access
- Consider nginx reverse proxy with SSL

Firewall Rules:

```
bash
# Only allow from specific IP range
sudo ufw allow from 192.168.1.0/24 to any port 5555
sudo ufw allow from 192.168.1.0/24 to any port 32227
```

Backup & Recovery

Backup Configuration

```
bash

# Backup all config files

cd ~/alpaca-onstepx

tar -czf alpaca-backup-$(date +%Y%m%d).tar.gz \
    config.py \
    *.py \
    requirements.txt

# Move to safe location

mv alpaca-backup-*.tar.gz ~/backups/
```

Restore from Backup

```
cd ~/alpaca-onstepx
tar -xzf ~/backups/alpaca-backup-YYYYMMDD.tar.gz
sudo systemctl restart onstepx-alpaca
```

Quick Reset

```
bash

# Stop service
sudo systemctl stop onstepx-alpaca

# Reset to defaults
cd ~/alpaca-onstepx
git reset --hard # If using git

# Or manually restore config.py
cp config.py.backup config.py

# Restart
sudo systemctl start onstepx-alpaca
```

Future Enhancements

Possible Additions:

1. Additional Hardware Support

- Pegasus Astro focusers
- Moonlite focusers
- QHYCFW filter wheels
- Manual filter wheel option

2. Advanced Features

- Web-based configuration interface
- Real-time monitoring dashboard
- Automatic focus offset calculation
- Temperature logging and graphs
- Session logging

3. Integration

- MQTT support for home automation
- Prometheus metrics export
- REST API documentation (Swagger)
- Mobile app for monitoring

4. Safety Features

- Automatic parking on disconnect
- Weather integration (stop on bad weather)
- Collision detection
- Emergency stop button

Architecture Supports Easy Extension!

All drivers follow the same pattern:

```
python

class NewDevice(BaseClass):
    def connect(self): ...
    def disconnect(self): ...
    # Device-specific methods

def create_device(mode='auto', ...):
    if mode == 'new_brand':
        return NewDevice(...)
```



Configuration:
☐ OnStepX IP address set correctly
All devices enabled in config.py
☐ Filter names customized
☐ Focus offsets measured and set
☐ Mode set for each device (auto/mock/zwo)
Testing:
☐ Telescope connects and reads position
☐ Discovery works (test_discovery.py)
☐ Filter wheel changes positions
☐ Focuser moves and reads temperature
☐ All tests pass
Production:
☐ Systemd service installed
☐ Service starts on boot
☐ Firewall configured
☐ Logs rotating properly
☐ Backup created
N.I.N.A. Integration:
Server auto-discovered
☐ All devices connect
☐ Can take test exposures
☐ Filters change correctly
Focuser moves correctly
Ready for imaging!
🎉 You're Ready!
Your complete OnStepX Alpaca Bridge is now:
• V Fully functional with all devices
• V Production ready with systemd service
• V Auto-discovering via UDP
• V Flexible with mock/real hardware modes
• Extensible for future additions

Clear skies and great imaging! 🌟 🔭

Support & Resources

Documentation:

- ASCOM Alpaca: https://ascom-standards.org/Developer/Alpaca.htm
- OnStepX: https://onstep.groups.io
- N.I.N.A.: https://nighttime-imaging.eu

Testing:

- All test scripts included
- Mock modes for safe testing
- Comprehensive error messages

Community:

- OnStep forums for mount questions
- ASCOM forums for API questions
- N.I.N.A. forums for integration help

Need help? Check the test scripts first - they'll diagnose most issues!