# Test Suite - README

## Where Are The Test Files?

All test files have been created as **downloadable artifacts** in this conversation. You can find them by scrolling through the artifacts panel (usually on the right side of the Claude interface).

## Created Test Files

### ✅ Available Now (Download These)

1. **test_config.py** - Tests configuration module
2. **test_helpers.py** - Tests helper functions
3. **test_telescope_hardware.py** - Tests telescope with real hardware
4. **test_api_management.py** - Tests management API endpoints
5. **test_api_camera.py** - Tests camera API endpoints
6. **run_all_tests.sh** - Automated test runner script

### 📝 Referenced in Testing Guide (Create Manually)

These are simple mock tests - you can copy the code from the "Complete Testing Guide" artifact:

- `test_telescope_mock.py`
- `test_camera_zwo_mock.py`
- `test_camera_zwo_hardware.py`
- `test_api_telescope.py`
- `test_integration.py`

## Installation Steps

### 1. Create Tests Directory

```bash
cd ~/onstepx-alpaca
mkdir -p tests
cd tests
```

### 2. Download Test Files

From the artifacts panel in this conversation, download each test file:

- Click the download icon (⬇️) on each artifact
- Save to `~/onstepx-alpaca/tests/`

Or manually copy/paste the code into new files:

```bash
cd ~/onstepx-alpaca/tests

# Create test files
nano test_config.py          # Copy from artifact
nano test_helpers.py         # Copy from artifact
nano test_telescope_hardware.py  # Copy from artifact
nano test_api_management.py  # Copy from artifact
nano test_api_camera.py      # Copy from artifact
nano run_all_tests.sh        # Copy from artifact

# Make test runner executable
chmod +x run_all_tests.sh
```

### 3. Verify File Structure

```bash
cd ~/onstepx-alpaca
tree -L 2
```

Should show:

```
~/onstepx-alpaca/
├── config.py
├── alpaca_helpers.py
├── telescope.py
├── camera_zwo.py
├── camera_touptek.py
├── filterwheel.py
├── focuser.py
├── main.py
├── venv/
└── tests/
    ├── test_config.py
    ├── test_helpers.py
    ├── test_telescope_hardware.py
    ├── test_api_management.py
    ├── test_api_camera.py
    └── run_all_tests.sh
```

## Running Tests

### Quick Test (Module Tests Only - No Hardware)

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

python3 test_config.py
python3 test_helpers.py
```

**Expected:** All tests pass ✅

### Hardware Tests (Requires Connected Devices)

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

# Test telescope (OnStepX must be connected)
python3 test_telescope_hardware.py

# Test camera (ZWO must be connected)
# Create test_camera_zwo_hardware.py first, then:
# python3 test_camera_zwo_hardware.py
```

### API Tests (Requires Running Server)

#### Terminal 1 - Start Server:

```bash
cd ~/onstepx-alpaca
source venv/bin/activate
python3 main.py
```

#### Terminal 2 - Run Tests:

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

python3 test_api_management.py
python3 test_api_camera.py
```

### Full Automated Test Suite

```bash
cd ~/onstepx-alpaca/tests
./run_all_tests.sh
```

This will:

1. Run module tests ✅
2. Run mock tests (if present) ✅
3. Ask if you want hardware tests ❓
4. Start server automatically 🚀
5. Run API tests ✅
6. Stop server 🛑
7. Report results 📊

## Test Results

### Success ✅

```
✅ Configuration module PASSED
✅ Helper functions PASSED
✅ ALL TELESCOPE HARDWARE TESTS PASSED
✅ ALL MANAGEMENT API TESTS PASSED
✅ ALL CAMERA API TESTS PASSED
```

### Failure ✗

```
✗ Test failed: assertion_error
✗ Cannot connect to server
✗ Camera not detected
```

## Troubleshooting Test Issues

### "Module not found"

```bash
cd ~/onstepx-alpaca
source venv/bin/activate  # Don't forget this!
```

### "Cannot connect to server"

```bash
# Terminal 1
cd ~/onstepx-alpaca
python3 main.py

# Terminal 2
cd ~/onstepx-alpaca/tests
python3 test_api_management.py
```

### "Permission denied" on serial port

```bash
sudo usermod -a -G dialout $USER
# Log out and back in
```

### "Camera not detected"

```bash
lsusb | grep -i zwo
python3 -c "import zwoasi as asi; asi.init('/usr/local/lib/libASICamera2.so'); print(asi.get_num_cameras())"
```

## Creating Additional Tests

Use the existing tests as templates:

```python
#!/usr/bin/env python3
"""Test description"""

import sys
sys.path.insert(0, '..')  # Add parent dir to path

# Import what you need
import config
import requests

def test_something():
    print("Testing something...")
    # Your test code
    assert True, "Test should pass"
    print("✓ Test OK\n")

if __name__ == '__main__':
    test_something()
    print("✅ TESTS PASSED\n")
```

## Test Coverage

### Level 1: Modules ✅

- ☑ Configuration
- ☑ Helper functions
- ☑ Coordinate parsing
- ☑ Validation
- ☑ Error handling

### Level 2: Devices ⚠️

- ☑ Telescope (hardware test provided)
- □ Camera mock (create from guide)
- □ Camera hardware (create from guide)

### Level 3: API ✅

- ☑ Management endpoints
- ☑ Camera endpoints
- □ Telescope endpoints (create from guide)
- □ Integration (create from guide)

### Level 4: Clients 📋

- □ N.I.N.A. (manual testing)
- □ PHD2 (manual testing)
- □ SharpCap (manual testing)

## Quick Reference

| Test File | Hardware Needed | Server Needed | What It Tests |
|---|---|---|---|
| test_config.py | ❌ | ❌ | Configuration validity |
| test_helpers.py | ❌ | ❌ | Parsing, validation |
| test_telescope_hardware.py | ✅ Mount | ❌ | OnStepX communication |
| test_api_management.py | ❌ | ✅ | Management endpoints |
| test_api_camera.py | ✅ Camera | ✅ | Camera endpoints |
| run_all_tests.sh | ❓ Optional | ✅ Auto-start | Everything |

## Summary

✅ **5 complete test files** provided as downloadable artifacts
✅ **1 automated test runner** script provided
📝 **5 additional tests** can be created from the Testing Guide
🎯 **Total coverage**: Module + Device + API + Integration levels

Download the artifacts and start testing! 🚀