

Atlas Processor Connection Fix Documentation

Overview

This document describes the fixes applied to resolve Atlas processor connection issues and implement proper third-party control protocol communication.

Date: October 19, 2025

Branch: `fix-atlas-connection-protocol`

Reference: ATS006993-B-AZM4-AZM8-3rd-Party-Control.pdf

Issues Identified

1. Incorrect Parameter Names ❌

Problem: Input gain controls were using incorrect parameter names

- **Used:** `Input1Gain`, `Input2Gain`, etc.
- **Should be:** `SourceGain_0`, `SourceGain_1`, etc. (0-based indexing)

Impact: Input gain commands were failing because the Atlas processor didn't recognize the parameter names.

2. Mock Data in AI Analyzer ❌

Problem: `atlas-ai-analyzer.ts` was returning hardcoded mock data

- Prevented real data collection from Atlas hardware
- Made debugging difficult

Impact: System appeared to work but wasn't actually communicating with hardware.

3. Indexing Confusion ⚠️

Problem: Mixed use of 1-based (UI) and 0-based (Atlas protocol) indexing

- UI displays: "Input 1", "Zone 1", etc. (1-based)
- Atlas protocol: `SourceGain_0`, `ZoneGain_0`, etc. (0-based)

Impact: Potential off-by-one errors in zone/input mapping.

Fixes Applied

✅ 1. Corrected Parameter Names

Input Gain API (`src/app/api/audio-processor/[id]/input-gain/route.ts`)

Before:

```
param: `Input${i}Gain` // Incorrect
```

After:

```
param: `SourceGain_${i}` // Correct (0-based)
```

Changes:

- Fixed parameter names in `getInputGainSettings()` function
- Fixed parameter names in `setInputGain()` function
- Added proper 0-based indexing conversion
- Added comprehensive documentation

✓ 2. Removed Mock Data

AI Analyzer (`src/lib/atlas-ai-analyzer.ts`)

Before:

```
return {
  processorId,
  processorModel: 'AZM8',
  inputLevels: { 1: -12, 2: -18, 3: -25, 4: -30 }, // Mock data
  outputLevels: { 1: -8, 2: -10, 3: -12, 4: -15 }, // Mock data
  // ...
}
```

After:

```
return {
  processorId,
  processorModel: 'Unknown', // To be populated from database
  inputLevels: {}, // Will be populated by UDP meter subscription
  outputLevels: {}, // Will be populated by UDP meter subscription
  // ...
}
```

Note: Added TODO comments for proper UDP meter implementation.

✓ 3. Enhanced Documentation

Added comprehensive inline documentation explaining:

- 0-based vs 1-based indexing
- Proper parameter naming conventions
- Atlas protocol requirements
- TCP/UDP communication details

Atlas Protocol Reference

Message Format

All commands must be JSON-RPC 2.0 format with `\r\n` terminator:

```
{"jsonrpc": "2.0", "method": "METHOD", "params": {...}, "id": N} \r\n
```

Methods

- **set** : Set a parameter value
- **bmp** : Bump (increment/decrement) a parameter
- **get** : Get current parameter value
- **sub** : Subscribe to parameter updates
- **unsub** : Unsubscribe from parameter updates

Common Parameters

Parameter	Description	Range	Format
SourceGain_X	Input gain for source X (0-based)	-80 to 0 dB	val
SourceMute_X	Input mute for source X	0 or 1	val
SourceMeter_X	Input level meter for source X	-80 to 0 dB	val (read-only)
ZoneGain_X	Zone volume for zone X (0-based)	-80 to 0 dB	val
ZoneMute_X	Zone mute for zone X	0 or 1	val
ZoneSource_X	Zone source selection for zone X	-1 to N-1	val
ZoneMeter_X	Zone level meter for zone X	-80 to 0 dB	val (read-only)

Connection Details

- **Protocol**: TCP
- **Port**: 5321
- **Authentication**: Not required for TCP control (HTTP uses Basic Auth)
- **Timeout**: 5 seconds recommended
- **Meter Updates**: Via UDP subscription (optional)

Testing the Connection

Method 1: Using Test Script

We've provided a comprehensive test script to verify the connection:

```
cd /home/ubuntu/github_repos/Sports-Bar-TV-Controller

# Install ts-node if not already installed
npm install -g ts-node

# Run the test script
ts-node scripts/test-atlas-connection.ts <ATLAS_IP_ADDRESS>

# Example:
ts-node scripts/test-atlas-connection.ts 192.168.5.101
```

The test script will:

1. ✓ Test TCP connection on port 5321
2. ✓ Test reading zone source
3. ✓ Test reading zone volume
4. ✓ Test reading source gain
5. ✓ Test parameter subscription

Method 2: Manual Testing with netcat

```
# Connect to Atlas processor
nc 192.168.5.101 5321

# Send a command (type this and press Enter):
{"jsonrpc": "2.0", "method": "get", "params": {"param": "SourceGain_0", "fmt": "val"}, "id": 1}

# You should receive a response like:
{"jsonrpc": "2.0", "result": -20.5, "id": 1}
```

Method 3: Using the Web UI

1. Navigate to the Audio Control Center
2. Click on “Atlas Programming Interface”
3. Add your Atlas processor (IP: 192.168.5.101, Port: 5321)
4. Click “Test Connection”
5. If successful, you should see “Authenticated” status
6. Try adjusting input gain sliders
7. Try adjusting zone volume controls

Verifying the Fix

1. Check Logs

The system logs all Atlas communication to:

```
~/Sports-Bar-TV-Controller/log/atlas-communication.log
```

Look for:

- ✓ Connection attempts and successes
- ✓ Commands sent (with correct parameter names)

- ✓ Responses received
- ✗ Any errors or timeouts

2. Monitor Network Traffic

```
# Install tcpdump if not available
sudo apt-get install tcpdump

# Monitor traffic on port 5321
sudo tcpdump -i any port 5321 -A

# You should see JSON-RPC messages
```

3. Test Input Gain Control

```
# Using curl to test the API
curl -X POST http://localhost:3000/api/audio-processor/<PROCESSOR_ID>/input-gain \
-H "Content-Type: application/json" \
-d '{
  "inputNumber": 1,
  "gain": -15
}'

# Expected response:
# {"success":true,"inputNumber":1,"gain":-15,"message":"Input 1 gain set to -15dB"}
```

4. Test Zone Volume Control

```
# Using curl to test zone control
curl -X POST http://localhost:3000/api/audio-processor/control \
-H "Content-Type: application/json" \
-d '{
  "processorId": "<PROCESSOR_ID>",
  "command": {
    "action": "volume",
    "zone": 1,
    "value": 50
  }
}'

# Expected response:
# {"success":true,"result":{"...},"message":"volume command executed successfully"}
```

Common Issues & Troubleshooting

Issue: “Unable to connect to processor”

Possible Causes:

1. Processor is offline or not on network
2. Incorrect IP address
3. Firewall blocking port 5321
4. Network routing issue

Solutions:

1. Ping the processor: `ping 192.168.5.101`
2. Try accessing web interface: `http://192.168.5.101`
3. Check firewall: `sudo iptables -L | grep 5321`
4. Verify processor is on same network/subnet

Issue: “Command timeout”**Possible Causes:**

1. Processor is slow to respond
2. Parameter name doesn't exist in configuration
3. Network latency

Solutions:

1. Increase timeout in code (currently 5 seconds)
2. Check parameter names in Atlas web interface
3. Check network latency: `ping -c 10 192.168.5.101`

Issue: “Invalid parameter name”**Possible Causes:**

1. Parameter not configured in Atlas
2. Using wrong parameter name
3. Index out of range

Solutions:

1. Log into Atlas web interface
2. Go to Settings > Third Party Control > Message Table
3. Verify available parameter names
4. Check that indices match your configuration






Issue: “Processor shows as offline”**Possible Causes:**

1. TCP connection failing
2. Authentication required (but not provided)
3. Processor ID mismatch

Solutions:

1. Test connection manually with netcat
2. Provide credentials if using HTTP endpoints
3. Verify processor ID in database matches actual hardware

Next Steps**Immediate Actions**

1.  Test connection with test script
2.  Verify input gain controls work
3.  Verify zone volume controls work
4.  Check logs for any errors
5.  Test with actual audio to confirm hardware responds

Future Enhancements

1. Implement UDP meter subscription for real-time level monitoring
 2. Add automatic parameter discovery from Atlas configuration
 3. Implement scene recall functionality
 4. Add group/combine zone controls
 5. Enhance error handling and retry logic
 6. Add connection health monitoring
-

Files Modified

Core Files

- `src/app/api/audio-processor/[id]/input-gain/route.ts` - Fixed parameter names
- `src/lib/atlas-ai-analyzer.ts` - Removed mock data
- `src/lib/atlasClient.ts` - Already correct ✓
- `src/lib/atlas-tcp-client.ts` - Already correct ✓
- `src/lib/atlas-logger.ts` - Already correct ✓

New Files

- `scripts/test-atlas-connection.ts` - Connection test tool
 - `docs/ATLAS_CONNECTION_FIX.md` - This documentation
-

Git Commit History

```
# View commits on fix branch
git log --oneline fix-atlas-connection-protocol

# Expected output:
# a533da7 Fix Atlas protocol: Use correct parameter names (SourceGain_X) and remove
# mock data
```

Support & References

Documentation

- **Atlas Protocol PDF:** `ATS006993-B-AZM4-AZM8-3rd-Party-Control.pdf`
- **AtlasIED Support:** `support@atlasied.com`
- **AtlasIED Website:** `https://www.atlasied.com`

Code References

- **JSON-RPC 2.0 Spec:** `https://www.jsonrpc.org/specification`
 - **Node.js Net Module:** `https://nodejs.org/api/net.html`
 - **Atlas Product Page:** `https://www.atlasied.com/atmosphere`
-

Contact

For questions or issues with this fix:

- Check the logs first: `log/atlas-communication.log`
 - Run the test script: `ts-node scripts/test-atlas-connection.ts`
 - Review this documentation
 - Contact system administrator
-

Document Version: 1.0

Last Updated: October 19, 2025

Status:  Ready for Testing