Sports Bar TV Controller - System Documentation

Server Information

Primary Server

- **URL**: http://24.123.87.42:3000/
- **IP Address**: 24.123.87.42
- **SSH Port**: 224
- **RDP Port**: 3389
- **Application Port**: 3000 (Next.js application)

Access Methods

1. **Web Application**: http://24.123.87.42:3000/
2. **SSH**: `ssh -p 224 user@24.123.87.42`
3. **RDP**: Connect to `24.123.87.42:3389` using Remote Desktop

Atlas Audio Processor Configuration

Processor Details

- **Model**: AZMP8 (Atmosphere Signal Processor with 1200W Amplifier)
- **Type**: 8-Zone Signal Processor with Integrated Amplification
- **Manufacturer**: AtlasIED
- **IP Address**: 192.168.5.101
- **Control Port**: 23 (Telnet/TCP)
- **Web Interface**: Port 80 (HTTP)

Specifications

- **Zones**: 8 independently controlled zones
- **Inputs**: 10 analog audio inputs
- 6 Mic/Line (Euroblock)
- 4 RCA (mono-summed)
- **Outputs**: 8 amplified outputs + 2 line outputs
- **Total System Power**: 1230W
- **Accessory Ports**: 4 (RJ45) for smart accessories
- **Network Control**: Dedicated Ethernet port

3rd Party Control Protocol

- **Protocol**: JSON-RPC 2.0 over TCP
- **Connection**: Telnet on port 23
- **Message Format**: `{"jsonrpc":"2.0","method":"...","params":{...}}\r\n`
- **Authentication**: 3rd Party Control must be enabled in Atlas web interface

Control Methods

The Atlas processor supports the following methods:
- **set**: Set a parameter value
- **bmp** (bump): Increment/decrement a parameter
- **sub**: Subscribe to parameter updates
- **unsub**: Unsubscribe from updates
- **get**: Get current parameter value

Parameter Format

Parameters use 0-based indexing:
- Zone 1 = `ZoneGain_0`, `ZoneMute_0`, `ZoneSource_0`
- Zone 2 = `ZoneGain_1`, `ZoneMute_1`, `ZoneSource_1`
- etc.

Example Commands

```
// Set Zone 1 volume to 50%
{"jsonrpc":"2.0","method":"set","params":{"param":"ZoneGain_0","pct":50}}

// Mute Zone 2
{"jsonrpc":"2.0","method":"set","params":{"param":"ZoneMute_1","val":1}}

// Set Zone 3 source to Source 1 (index 0)
{"jsonrpc":"2.0","method":"set","params":{"param":"ZoneSource_2","val":0}}

// Subscribe to Zone 1 gain updates
{"jsonrpc":"2.0","method":"sub","params":{"param":"ZoneGain_0","fmt":"val"}}
```

Application Architecture

Frontend

- **Framework**: Next.js 14 with App Router
- **UI Library**: React with TypeScript
- **Styling**: Tailwind CSS + Custom Components
- **State Management**: React Hooks

Backend

- **Runtime**: Node.js
- **API**: Next.js API Routes
- **Database**: PostgreSQL with Prisma ORM
- **Real-time**: TCP sockets for Atlas communication

Atlas Integration Components

1. TCP Client Library (`src/lib/atlasClient.ts`)

- Implements JSON-RPC 2.0 protocol
- Manages persistent TCP connections
- Handles command queuing and responses
- Automatic reconnection logic

2. Control API (`src/app/api/audio-processor/control/route.ts`)

- REST API for zone control
- Maps UI actions to Atlas TCP commands
- Handles authentication and validation
- Returns formatted responses

3. Frontend Components

- **AtlasProgrammingInterface**: Configuration and setup UI
- **AudioZoneControl**: Zone volume and source control
- **AtlasAIMonitor**: Real-time monitoring and AI analysis

Setup and Configuration

1. Atlas Processor Initial Setup

1. Connect to Atlas web interface at http://192.168.5.101
2. Navigate to Settings > Third Party Control
3. Enable “Third Party Control”
4. Note: Default credentials are typically admin/admin (verify with physical unit)

2. Application Configuration

1. Add processor in Audio Control Center
2. Enter processor details:
   - Name: (e.g., “Main Audio Processor”)
   - Model: AZMP8
   - IP Address: 192.168.5.101
   - Port: 80 (for web interface)
   - TCP Port: 23 (for control commands)
3. Test connection using “Test Connection” button

3. Zone Configuration

1. Define zone names and assignments
2. Configure input sources
3. Set default volumes and mute states
4. Save configuration to database

Troubleshooting

Atlas Connection Issues

Problem: Cannot connect to Atlas processor

Solutions:

1. Verify network connectivity: `ping 192.168.5.101`
2. Check if 3rd Party Control is enabled in Atlas web interface
3. Verify firewall settings allow port 23 (telnet)
4. Check TCP port is not already in use
5. Review Atlas logs for connection attempts

Problem: Commands not executing

Solutions:
1. Verify message format includes `\r\n` terminator
2. Check parameter names match Atlas configuration
3. Ensure zone/source indices are 0-based
4. Review Atlas response messages for errors
5. Check if processor is in a locked state

Problem: Subscriptions not receiving updates

Solutions:
1. Verify subscription was successful (check response)
2. Ensure connection remains open
3. Check for UDP port 3804 if using meter subscriptions
4. Review buffer handling in TCP client

Application Issues

Problem: Processor shows “offline” status

Solutions:
1. Click “Test Connection” button
2. Verify processor IP address and ports
3. Check network connectivity between server and processor
4. Review application logs for connection errors

Problem: Configuration not saving

Solutions:
1. Check database connection
2. Verify Prisma schema is up to date
3. Run database migrations if needed
4. Check application logs for errors

Database Schema

AudioProcessor Table

- `id` : UUID (primary key)
- `name` : String
- `model` : String (e.g., “AZMP8”)
- `ipAddress` : String
- `port` : Integer (web interface port)
- `tcpPort` : Integer (TCP control port, default 23)
- `zones` : Integer (number of zones)
- `status` : Enum (online, offline, error)
- `username` : String (optional, encrypted)
- `password` : String (optional, encrypted)
- `lastSeen` : DateTime
- `createdAt` : DateTime
- `updatedAt` : DateTime

AudioZone Table

- `id`: UUID (primary key)
- `processorId`: UUID (foreign key)
- `zoneNumber`: Integer (1-based)
- `name`: String
- `volume`: Integer (0-100)
- `muted`: Boolean
- `currentSource`: String
- `createdAt`: DateTime
- `updatedAt`: DateTime

Security Considerations

1. **Credentials Storage**: Atlas credentials are encrypted in database
2. **Network Security**: Ensure firewall rules restrict access to ports 23, 80, 3000
3. **Authentication**: Implement authentication for web application access
4. **Audit Logging**: Log all control commands for accountability

Maintenance

Regular Tasks

1. **Daily**: Monitor processor status and connectivity
2. **Weekly**: Review application logs for errors
3. **Monthly**: Backup database and configuration
4. **Quarterly**: Review and update firmware if available

Log Locations

- **Application Logs**: Check server console output
- **Atlas Logs**: Available in Atlas web interface
- **Database Logs**: PostgreSQL logs (if enabled)

Automatic Hardware Query Feature

Overview

The application now automatically queries the Atlas hardware during processor creation to fetch real configuration instead of using mock/model data. This ensures that the application always displays accurate source and zone names as configured in the Atlas web interface.

How It Works

1. Processor Creation Flow

When a new Atlas processor is created:
1. The processor record is created in the database with status ‘offline’
2. The application automatically connects to the Atlas hardware via TCP (port 23)
3. It queries all source names using `SourceName_X` parameters
4. It queries all zone names using `ZoneName_X` parameters
5. It queries current zone status (source selection, volume, mute state)

6. The real hardware configuration is saved to the database
7. The processor status is updated to ‘online’ if successful

2. Hardware Query Parameters

Source Queries:

```
{"jsonrpc":"2.0","method":"get","params":{"param":"SourceName_0","fmt":"str"},"id":1}
{"jsonrpc":"2.0","method":"get","params":{"param":"SourceName_1","fmt":"str"},"id":2}
// ... continues for all sources
```

Zone Queries:

```
{"jsonrpc":"2.0","method":"get","params":{"param":"ZoneName_0","fmt":"str"},"id":10}
{"jsonrpc":"2.0","method":"get","params":{"param":"ZoneSource_0","fmt":"val"},"id":11}
{"jsonrpc":"2.0","method":"get","params":{"param":"ZoneGain_0","fmt":"pct"},"id":12}
{"jsonrpc":"2.0","method":"get","params":{"param":"ZoneMute_0","fmt":"val"},"id":13}
// ... continues for all zones
```

3. Configuration Storage

The queried hardware configuration is stored in two places:
1. **Database**: Zone records with real names and current status
2. **File System**: JSON configuration files in `data/atlas-configs/`

Configuration file format:

```
{
  "processorId": "clxxx...",
  "ipAddress": "192.168.5.101",
  "port": 23,
  "model": "AZMP8",
  "inputs": [
    {
      "id": "source_0",
      "number": 1,
      "name": "Matrix 1 (M1)",
      "type": "atlas_configured",
      "parameterName": "SourceName_0",
      "queriedFromHardware": true
    }
  ],
  "outputs": [
    {
      "id": "zone_0",
      "number": 1,
      "name": "Main Bar",
      "type": "zone",
      "parameterName": "ZoneName_0",
      "currentSource": 0,
      "volume": 75,
      "muted": false,
      "queriedFromHardware": true
    }
  ],
  "queriedAt": "2024-10-19T12:34:56.789Z",
  "source": "hardware_query_auto"
}
```

4. API Endpoints

Create Processor with Auto-Query:

```
POST /api/audio-processor
{
  "name": "Main Processor",
  "model": "AZMP8",
  "ipAddress": "192.168.5.101",
  "port": 80,
  "tcpPort": 23,
  "zones": 8,
  "description": "Main audio processor"
}
```

Response includes hardware configuration:

```
{
  "processor": {
    "id": "clxxx...",
    "name": "Main Processor",
    "status": "online",
    "inputs": 9,
    "outputs": 5,
    "hardwareQuerySuccess": true
  },
  "hardwareConfig": {
    "sources": 9,
    "zones": 5,
    "queriedAt": "2024-10-19T12:34:56.789Z",
    "inputs": [...],
    "outputs": [...]
  },
  "message": "Processor created and hardware configuration queried successfully"
}
```

Manual Hardware Query:

```
POST /api/atlas/query-hardware
{
  "processorId": "clxxx..."
}
```

Skip Auto-Query (Optional):
To skip automatic hardware query during creation:

```
POST /api/audio-processor
{
  ...,
  "skipHardwareQuery": true
}
```

5. Error Handling

If hardware query fails during processor creation:
- The processor is still created in the database

- Status remains ‘offline’
- Model-based default values are used temporarily
- User receives a warning message
- User can manually trigger hardware query later

Benefits

1. **No Mock Data**: Always displays actual hardware configuration
2. **Real-Time Accuracy**: Zone and source names match Atlas web interface
3. **Automatic Setup**: No manual configuration needed
4. **Current Status**: Displays actual zone states (source, volume, mute)
5. **Easy Updates**: Re-query hardware anytime to sync changes

Implementation Files

Key Components:

- `src/app/api/audio-processor/route.ts` - Auto-query on processor creation
- `src/app/api/atlas/query-hardware/route.ts` - Manual hardware query endpoint
- `src/lib/atlas-hardware-query.ts` - Hardware query logic
- `src/lib/atlasClient.ts` - TCP client with JSON-RPC 2.0 support
- `src/lib/atlas-tcp-client.ts` - Legacy TCP client (deprecated)

Testing the Integration

1. **Create New Processor:**
   - Navigate to Audio Control Center
   - Click “Add Processor”
   - Fill in processor details
   - Click “Create”
   - Verify hardware configuration is automatically fetched

2. **Verify Real Data:**
   - Check that source names match Atlas web interface
   - Check that zone names match Atlas configuration
   - Verify current zone states are accurate

3. **Manual Query:**
   - Open processor settings
   - Click “Query Hardware”
   - Verify configuration updates

4. **Monitor Logs:**
   - Check console for `[Atlas Query]` messages
   - Verify successful connection and parameter queries
   - Review any error messages

Reference Documentation

Atlas Documents

1. **ATS007275-Atmosphere-Data-Sheet_RevE.pdf**: Full specifications
2. **ATS006190F-AZM4-AZM8-Data-Sheet.pdf**: Model-specific details
3. **ATS006993-B-AZM4-AZM8-3rd-Party-Control.pdf**: TCP control protocol

AtlasIED Resources

- **Website**: https://www.atlasied.com
- **Support**: support@atlasied.com
- **Phone**: (800) 876-3333

Version History

v1.1.0 (2024-10-19)

- **Major Enhancement**: Automatic hardware query on processor creation
- Removed mock/model data dependencies
- Implemented real-time hardware configuration fetching
- Added automatic zone creation with real names from Atlas
- Enhanced error handling for hardware query failures
- Updated system documentation with new feature details
- Added configuration file storage for hardware queries

v1.0.0 (2024-10-18)

- Initial system documentation
- Atlas AZMP8 integration completed
- TCP control protocol implemented
- Fixed rendering errors in AtlasProgrammingInterface
- Updated TCP port from 3804 to 23 (correct telnet port)
- Added defensive null checks for array rendering

Document Last Updated: October 19, 2024
Maintained By: System Administrator
Next Review Date: November 19, 2024