Wolfpack Matrix AI Knowledge Base

Overview

This knowledge base provides comprehensive information about Wolfpack modular matrix switchers for AI-powered analysis and optimization.

Matrix Switcher Capabilities

Supported Models

- **36x36 Matrix**: 36 inputs, 36 outputs, full crosspoint switching
- **Modular Design**: Expandable card cage architecture
- **Protocol Support**: TCP (port 5000) and UDP (port 4000) control
- **Scene Management**: Save and recall routing configurations
- **Real-time Control**: Immediate switching with status feedback

Communication Protocols

TCP Control (Port 5000)

- **Advantages**: Reliable delivery, error checking, ordered packets
- **Best For**: Critical routing operations, configuration changes
- **Characteristics**: Higher latency but guaranteed delivery
- **Recommended**: For automated systems requiring reliability

UDP Control (Port 4000)

- **Advantages**: Lower latency, faster response times
- **Best For**: Real-time switching, live event management
- **Characteristics**: No delivery guarantee but faster execution
- **Recommended**: For interactive control applications

Command Reference

Basic Switching Commands

Input to All Outputs

- **Command**: `YAll.` (where Y = input number)
- **Example**: `1ALL.` switches input 1 to all outputs
- **Use Case**: Emergency broadcasts, global content distribution
- **Response**: `OK` on success, `ERR` on failure

One-to-One Mapping

- **Command**: `All1.`
- **Function**: Maps input 1→output 1, input 2→output 2, etc.
- **Use Case**: System reset, default configuration
- **Best Practice**: Use during maintenance or setup

Single Input to Output

- **Command**: `YXZ.` (Y=input, Z=output)
- **Example**: `1X2.` switches input 1 to output 2
- **Most Common**: Primary switching operation
- **Validation**: Always verify with status query after switching

Multi-Output Switching

- **Command**: `YXZ&Q&W.` (Y=input, Z,Q,W=outputs)
- **Example**: `1X2&3&4.` switches input 1 to outputs 2, 3, and 4
- **Efficiency**: Single command for multiple operations
- **Limitation**: Maximum outputs per command varies by model

Scene Management Commands

Save Scene

- **Command**: `SaveY.` (Y = scene number)
- **Example**: `Save2.` saves current routing to scene 2
- **Capacity**: Typically 8-16 scenes depending on model
- **Best Practice**: Document scene contents for operators

Recall Scene

- **Command**: `RecallY.` (Y = scene number)
- **Example**: `Recall2.` restores saved scene 2
- **Speed**: Fastest way to implement complex routing changes
- **Use Case**: Event-based configurations, emergency scenarios

System Control Commands

Buzzer Control

- **Commands**: `BeepON.` and `BeepOFF.`
- **Function**: Enable/disable audible feedback
- **Recommendation**: Turn off in noise-sensitive environments
- **Default**: Usually enabled on factory reset

Status Query

- **Command**: `Y?.` (Y = input number)
- **Example**: `1?.` returns routing status for input 1
- **Response**: Lists all outputs receiving this input
- **Usage**: Troubleshooting, verification, status monitoring

Performance Characteristics

Switching Speed

- **Typical**: <100ms for single operation
- **Optimal**: <50ms under ideal network conditions
- **Factors**: Network latency, protocol choice, system load

Network Requirements

- **Bandwidth**: Minimal (<1KB per command)

- **Latency**: <10ms recommended for real-time control
- **Stability**: Wired connections preferred over WiFi
- **Redundancy**: Consider backup control paths for critical systems

Common Issues and Solutions

Connection Problems

“Cannot Connect” Errors

- **Causes**: Incorrect IP address, network issues, matrix power state
- **Solutions**:
  1. Verify IP address configuration
  2. Test network connectivity with ping
  3. Check matrix power status and network cables
  4. Verify firewall settings on control device

Intermittent Disconnections

- **Causes**: Network instability, IP conflicts, power issues
- **Solutions**:
  1. Switch to wired connection if using WiFi
  2. Check for IP address conflicts
  3. Implement connection retry logic
  4. Monitor power supply stability

Command Execution Issues

“ERR” Responses

- **Causes**: Invalid commands, out-of-range values, syntax errors
- **Solutions**:
  1. Verify command syntax (always end with period)
  2. Check input/output range (1-36 for 36x36 matrix)
  3. Ensure proper case formatting
  4. Test with simple commands first

Slow Response Times

- **Causes**: Network congestion, protocol choice, system overload
- **Solutions**:
  1. Switch from UDP to TCP or vice versa
  2. Reduce network traffic during operations
  3. Implement command queuing to avoid overload
  4. Check matrix CPU usage and memory

Configuration Issues

Duplicate Labels

- **Problem**: Multiple inputs/outputs with same name
- **Impact**: Operator confusion, routing errors
- **Solution**: Use unique, descriptive labels for all channels

Missing Layout Mapping

- **Problem**: Outputs not mapped to physical TV locations
- **Impact**: Inefficient operations, wrong content on TVs

- **Solution**: Implement location-based labeling system

Optimization Strategies

Routing Efficiency

Group Similar Operations

- Use multi-output commands when possible
- Batch scene recalls for complex changes
- Minimize single-operation commands during busy periods

Strategic Scene Usage

- Create scenes for common configurations
- Use descriptive scene names
- Update scenes when layout changes

Network Optimization

Protocol Selection

- **TCP for reliability**: Configuration changes, critical operations
- **UDP for speed**: Live switching, real-time control
- **Hybrid approach**: TCP for setup, UDP for operation

Connection Management

- Implement connection pooling
- Use persistent connections when possible
- Monitor connection health continuously

Integration Best Practices

Audio Routing Integration

- Map matrix audio outputs to Atlas audio inputs
- Use consistent naming between systems
- Monitor for audio routing conflicts
- Implement audio-follow-video logic where appropriate

Layout System Integration

- Synchronize output labels with TV positions
- Import layout data for automatic mapping
- Validate physical connections match configuration
- Update both systems when layout changes

Monitoring and Logging

- Log all switching operations with timestamps
- Monitor command success rates
- Track performance metrics (latency, errors)
- Implement alerting for system issues

AI Analysis Patterns

Normal Operation Indicators

- Command success rate >95%
- Response times <200ms
- Stable network connections
- Proper label configuration

Warning Signs

- Increasing error rates
- Growing response times
- Frequent disconnections
- Configuration inconsistencies

Critical Issues

- Connection failures
- Command rejection
- System unresponsiveness
- Hardware fault indicators

Future Enhancements

Advanced Features

- **Automatic failover**: Switch to backup inputs on signal loss
- **Load balancing**: Distribute content across multiple outputs
- **Intelligent routing**: AI-driven content distribution
- **Predictive maintenance**: Early warning for hardware issues

Integration Opportunities

- **Voice control**: Natural language matrix control
- **Mobile apps**: Remote switching capabilities
- **Analytics**: Usage patterns and optimization recommendations
- **API expansion**: RESTful interfaces for third-party integration