CEC TV Control Integration Guide

Overview

HDMI-CEC (Consumer Electronics Control) integration allows you to control TV power, input switching, and audio directly over HDMI connections. This system uses the Pulse-Eight USB CEC adapter to send commands from your server to TVs via the Wolfpack matrix.

Hardware Requirements

1. **Pulse-Eight USB CEC Adapter** (P8-USBCECv1)
   - USB connection to server
   - HDMI passthrough connection

2. **Wolfpack Matrix**
   - One input dedicated to CEC adapter
   - HDMI connections to all TVs

3. **TVs with CEC Support**
   - All modern TVs support HDMI-CEC
   - May be labeled as: Anynet+ (Samsung), Bravia Sync (Sony), Simplink (LG), etc.

Installation

1. Software Installation

LibCEC has been installed on your system:

```
# Already installed:
- libcec6
- libcec-dev
- cec-utils
- python3-cec
```

2. Hardware Connection

```
Server → USB Cable → CEC Adapter → HDMI → Wolfpack Matrix Input
Matrix Outputs → HDMI Cables → TVs
```

Connection Steps:
1. Connect CEC adapter USB port to server USB port
2. Connect CEC adapter HDMI input to server video output (optional for passthrough)
3. Connect CEC adapter HDMI output to Wolfpack matrix input (e.g., Input 10)
4. Connect Wolfpack outputs to all TVs via HDMI

3. Configuration in Sports Bar AI Assistant

Step A: Configure Matrix CEC Input

1. Navigate to **Matrix Control** page

2. In the configuration section, find **“CEC Adapter Input”** dropdown
3. Select which input channel has the CEC adapter connected (e.g., Input 10)
4. Save the configuration

Step B: Test CEC Connection

1. Navigate to **CEC Control** page (`/cec-control`)
2. The system will automatically detect the USB CEC adapter
3. Click **“Scan”** to detect connected TVs
4. You should see a list of detected devices

Features

1. Direct CEC Commands

Uses libCEC to send commands directly to TVs via the USB adapter:
- **Power On** - Wake TVs from standby
- **Power Off** - Put TVs into standby mode
- **Toggle Power** - Smart toggle based on current state
- **Mute/Unmute** - Control TV audio
- **Input Switching** - Change HDMI input on TVs
- **Volume Control** - Adjust TV volume levels

2. Matrix-Routed CEC Control

Routes CEC input through matrix for system-wide control:
- Route CEC input to specific TV outputs
- Send power commands after routing
- Control all TVs simultaneously or individually
- Configurable delays for routing stability

API Endpoints

Initialize CEC

```
GET/POST /api/cec/initialize
```

Initializes libCEC and detects USB CEC adapters.

Response:

```
{
  "success": true,
  "message": "CEC adapter initialized: /dev/ttyACM0",
  "adapters": ["/dev/ttyACM0"]
}
```

Scan for Devices

```
GET /api/cec/scan?refresh=true
```

Scans the CEC bus for connected devices.

Response:

```
{
  "success": true,
  "devices": [
    {
      "address": "0",
      "name": "TV",
      "vendor": "Samsung",
      "osdName": "Samsung TV",
      "powerStatus": "on",
      "cecVersion": "1.4"
    }
  ],
  "count": 1
}
```

Send CEC Command

```
POST /api/cec/command
Content-Type: application/json

{
  "action": "power_on",
  "tvAddress": "0",
  "params": {}
}
```

Available Actions:

- `power_on` - Turn TV on
- `power_off` - Turn TV off (standby)
- `toggle_power` - Toggle based on current state
- `set_input` - Change HDMI input (params: `{ inputNumber: 1 }`)
- `set_volume` - Set volume level (params: `{ volume: 50 }`)
- `mute` - Mute/unmute audio
- `send_key` - Send navigation keys (params: `{ key: "up" }`)
- `raw` - Send raw CEC command (params: `{ command: "10:04" }`)

Response:

```
{
  "success": true,
  "message": "Command sent successfully"
}
```

Get Power Status

```
GET /api/cec/status?tvAddress=0
```

Returns current power status of specified TV.

Response:

```
{
  "success": true,
  "status": "on",
  "devices": [...]
}
```

Usage Examples

1. Bartender Remote Integration

The CEC control is integrated into the bartender remote (`/remote` page):
- Compact CEC controls for quick TV power management
- Works alongside existing IR and network controls
- Automatic device detection and status updates

2. Dedicated CEC Control Page

Access the full CEC control interface at `/cec-control`:
- Full power control (on/off/toggle)
- HDMI input switching
- Audio mute control
- Device scanning and status
- Detailed device information

3. Programmatic Control

Use the CEC service in your code:

```
import { cecService } from '@/lib/cec-service';

// Initialize
await cecService.initialize();

// Scan for devices
const devices = await cecService.scanDevices();

// Power on TV
const result = await cecService.powerOn('0');

// Set HDMI input 2
await cecService.setInput(2, '0');

// Toggle power
await cecService.togglePower('0');
```

Troubleshooting

CEC Adapter Not Detected

```
# Check if adapter is connected
lsusb | grep -i "pulse"

# Test CEC client
cec-client -l

# Check device permissions
ls -la /dev/ttyACM*
```

TVs Not Responding

1. **Verify TV CEC is enabled:**
   - Check TV settings for CEC/Anynet+/Bravia Sync
   - Enable the feature (usually disabled by default)

2. **Check HDMI connections:**
   - Ensure all HDMI cables support CEC
   - Try different HDMI ports on TVs

3. **Verify matrix routing:**
   - Ensure CEC input can be routed to TV outputs
   - Test by manually routing the input

Commands Timing Out

- Increase delay times in CEC configuration
- Check network latency to matrix
- Verify matrix firmware is up to date

CEC Address Reference

Standard CEC addresses:
- `0` - TV
- `1` - Recording Device 1
- `2` - Recording Device 2
- `3` - Tuner 1
- `4` - Playback Device 1
- `5` - Audio System
- `F` - Broadcast (all devices)

Most TVs use address `0`.

Integration with Existing Systems

Works With:

- ✅ IR Control (Global Cache) - CEC provides power control, IR provides channel/menu navigation
- ✅ Network Control (DirecTV, Fire TV) - CEC controls TV, network controls source devices
- ✅ Wolfpack Matrix - CEC input routing for system-wide control

- ✅ Atlas Audio System - Independent audio routing

Recommended Setup:

1. Use CEC for TV power and input switching
2. Use IR/Network for source device control (cable boxes, Fire TV, etc.)
3. Use matrix routing for video distribution
4. Use Atlas for audio distribution

Advanced Features

Batch TV Control

Control multiple TVs simultaneously:
1. Configure CEC input in matrix
2. Use system-wide power control
3. Route CEC input to multiple outputs
4. Send power commands to all

Scheduled Power Management

Create automation scripts:

```
# Power on all TVs at opening time
curl -X POST http://localhost:3000/api/cec/command \
  -H "Content-Type: application/json" \
  -d '{"action":"power_on","tvAddress":"0"}'

# Route to specific TVs and power off
# (Implement via matrix routing + CEC commands)
```

TV Status Monitoring

Monitor TV power states:
- Poll `/api/cec/status` periodically
- Track TV online/offline status
- Alert on power anomalies
- Log power events for analytics

Support

Pulse-Eight Resources

- Documentation: https://www.pulse-eight.com/
- Support: support@pulse-eight.com
- LibCEC GitHub: https://github.com/Pulse-Eight/libcec

System Logs

Check system logs for CEC activity:

```
# View CEC client logs
journalctl -u cec-client

# Check application logs
tail -f /home/ubuntu/Sports-Bar-TV-Controller/logs/cec.log
```

Future Enhancements

Potential future features:
- [ ] TV power scheduling
- [ ] Automatic TV detection and labeling
- [ ] Multi-room zone power management
- [ ] Integration with occupancy sensors
- [ ] Power usage analytics
- [ ] Remote CEC control via mobile app
- [ ] Backup/restore TV configurations

Last Updated: October 1, 2025
Version: 1.0.0