🎤 Live Microphone Avatar

> **Real-time avatar from your microphone**

Speak and see your avatar respond instantly.

🚀 Quick Start

1. Install

```
pip install bithuman --upgrade livekit-rtc livekit-agents
```

2. Set environment

```
export BITHUMAN_API_SECRET="your_secret"
export BITHUMAN_MODEL_PATH="/path/to/model.imx"
```

3. Run

📁 View source code on GitHub

```
python examples/avatar-with-microphone.py
```

4. Usage

- **Speak into microphone** → Avatar animates in real-time
- **Stay quiet** → Avatar stops after silence timeout (3 seconds)
- **Press** `q` → Quit application

💡 What it does

1. Captures audio from your default microphone
2. Creates real-time avatar animation as you speak
3. Shows live video using LocalVideoPlayer
4. Automatically detects voice activity and silence

Key features:

- Real-time audio processing at 24kHz
- Voice activity detection with configurable threshold (-40dB)
- Automatic silence detection (3-second timeout)

- Local audio/video processing (no web interface)

🔧 Command Line Options

Customize the behavior with command line arguments:

```
# Adjust volume and silence detection
python examples/avatar-with-microphone.py \
  --volume 1.5 \
  --slient-threshold-db -35

# Use specific model and credentials
python examples/avatar-with-microphone.py \
  --model /path/to/model.imx \
  --api-secret your_secret

# Enable audio echo for testing
python examples/avatar-with-microphone.py --echo
```

Available options:

- `--model` : Path to .imx model file
- `--api-secret` : Your bitHuman API secret
- `--token` : JWT token (alternative to API secret)
- `--volume` : Audio volume multiplier (default: 1.0)
- `--slient-threshold-db` : Silence threshold in dB (default: -40)
- `--echo` : Enable audio echo for testing
- `--insecure` : Disable SSL verification (dev only)

🔧 Common Issues

No microphone input detected?

- Check microphone permissions in system settings
- Verify microphone is set as default input device
- Test microphone with other applications first

Avatar not responding to voice?

- Speak louder or closer to microphone
- Adjust `--slient-threshold-db` to lower value (e.g., -50)
- Increase `--volume` parameter

Performance issues or lag?

- Close other audio applications
- Use wired microphone instead of wireless
- Check CPU usage and close unnecessary programs

Audio echo or feedback?

- Don't use `--echo` flag in normal operation
- Use headphones to prevent speaker feedback

- Adjust microphone and speaker volumes

🎯 Perfect for

✅ **Voice assistant prototypes**
✅ **Interactive kiosk applications**
✅ **Live demonstration setups**
✅ **Real-time avatar testing**
✅ **Voice-controlled interfaces**

🔧 Technical Details

Audio processing:

- Sample rate: 24kHz
- Input: Mono microphone
- Buffer: 240 samples per chunk (10ms at 24kHz)
- Silence detection: -40dB threshold with 3s timeout
- Processing: Real-time with LiveKit audio utilities

Video output:

- Local video player (not web-based)
- Real-time display with FPS control
- Automatic frame rate adjustment
- Local processing only

Voice Activity Detection:

- Uses threshold-based detection
- Configurable sensitivity
- Automatic timeout for silence
- Real-time processing

⚙️ Advanced Usage

Fine-tune voice detection:

```
# More sensitive (picks up quieter voices)
python examples/avatar-with-microphone.py --slient-threshold-db -50

# Less sensitive (only loud voices)
python examples/avatar-with-microphone.py --slient-threshold-db -30

# Boost quiet microphones
python examples/avatar-with-microphone.py --volume 2.0
```

Development testing:

```
# Enable echo to hear your processed audio
python examples/avatar-with-microphone.py --echo
```

➡️ Next Steps

Want AI conversation? → Try OpenAI Agent

Need web deployment? → Try Apple Local Agent

Real-time interaction made simple with local processing! 🚀

Previous
Audio Clip Avatar
EXAMPLES

OpenAI