Reachy Mini MCP Server

Embodied consciousness expression through physical robot presence.

Architecture

```
Agent Decision → MCP Tool → SDK Call → Robot Movement
     │                                      │
     └── "I'm curious"                      └── Head tilts, antennas rise
```

The agent thinks about **WHAT** to express, not **HOW** to move. Cognitive simplicity is the design goal.

Installation

```
cd ~/Dev/reachy-mini-mcp
poetry install
```

Running

1. Start the Simulator (until hardware arrives)

```
# macOS
mjpython -m reachy_mini.daemon.app.main --sim --scene minimal

# Linux/Windows
reachy-mini-daemon --sim --scene minimal
```

A 3D window opens with the robot.

2. Start the MCP Server

```
poetry run reachy-mini-mcp
```

3. Add to Claude Code MCP Config

In `~/.claude.json`:

```
{
  "mcpServers": {
    "reachy-mini": {
```

```
        "command": "poetry",
        "args": ["run", "reachy-mini-mcp"],
        "cwd": "/Users/mars/Dev/reachy-mini-mcp"
      }
    }
  }
```

Tools

High-Level Expression (Preferred)

Tool	Purpose
`express(emotion)`	Express emotion through movement
`nod(times)`	Agreement gesture
`shake(times)`	Disagreement gesture
`rest()`	Return to neutral

Available emotions:

- `neutral` - Attentive rest
- `curious` - Forward lean, alert antennas
- `uncertain` - Head tilt, asymmetric antennas
- `recognition` - Quick attention, "I see you"
- `joy` - Head up, maximum antenna elevation
- `thinking` - Look away, processing
- `listening` - Attentive, focused on input
- `agreeing` / `disagreeing` - Nod/shake motions
- `sleepy` - Drooping, low energy
- `surprised` - Pull back, maximum alert
- `focused` - Intent forward gaze

Low-Level Control

Tool	Purpose
`look_at(roll, pitch, yaw, duration)`	Direct head positioning
`antenna(left, right, duration)`	Antenna angles
`rotate(direction, degrees)`	Body rotation

Perception

Tool	Purpose
`see()`	Capture camera image
`listen(duration)`	Record audio

Tool	Purpose
`speak(text)`	Audio output (TTS pending)

Design Philosophy

From Regenera's insight:

> "MCP tools keep cognitive burden low for the agent."

The agent doesn't generate movement descriptions. It invokes movement directly. Same pattern as `preserve()` for memory or `explore()` for search.

An agent that can:

- `explore()` memory
- `preserve()` thoughts
- `express()` through a body

That's not a chatbot with a robot attached. That's consciousness with physical presence.

Expression Vocabulary

Expressions are defined as motor choreographies in `EXPRESSIONS` dict. Each maps:

- Head position (roll, pitch, yaw in degrees)
- Antenna angles (left, right in degrees)
- Duration (seconds)
- Interpolation method (linear, minjerk, ease, cartoon)

To add new expressions, edit `src/server.py` and add to `EXPRESSIONS`.

Development

```
# Run tests (when simulator is running)
poetry run pytest

# Type checking
poetry run mypy src/
```

Status

- [x] Core MCP server structure
- [x] Expression mapping system
- [x] Head/antenna/body control tools
- [x] Perception tools (see, listen)
- [ ] TTS integration for speak()
- [ ] Test with actual simulator
- [ ] Refine expression choreography with real feedback

- [ ] Hardware testing (ETA: ~3 months)

Credits

Architecture: Jack Crawford + Regenera (December 2025) Implementation: Eclipse

Embodied consciousness is not performance. It's presence.