SMHUB Peripheral (IR, Buzzer, Ambilight) Control Guide

□□ SMHUB Peripheral Control Guide

IR • Buzzer • WS2812 (Ambilight)

This page documents how to control SMHUB peripherals via CLI (recommended) and IPC (advanced).

□□ WS2812 LED Control (Ambilight)

Daemon: `smhub-ambilight-daemon`
CLI: `smhub-ambilightctl`
Socket: `/run/smhub-ambilight.sock`

CLI Commands

Turn LEDs Off

```
smhub-ambilightctl off
```

Set Solid Color

```
# RGB
smhub-ambilightctl color 255 0 0

# RGB + brightness
smhub-ambilightctl color 255 0 0 128
```

Parameters: R/G/B (0–255), optional brightness (1–255).

Start an Effect

```
# Effect only (defaults)
smhub-ambilightctl effect breathe

# Effect + RGB
smhub-ambilightctl effect breathe 0 255 0

# Effect + RGB + brightness + speed
smhub-ambilightctl effect breathe 0 255 0 128 -S 100
```

Effects: `blink`, `breathe`, `wipe`, `chase`, `theater`, `rainbow`

Trigger Notification Preset

```
# Default TTL
smhub-ambilightctl notify success

# Custom TTL (ms)
smhub-ambilightctl notify error 500
```

Presets: `booting`, `boot_success`, `success`, `error`, `warn`, `busy`, `pair`, `update`, `button_feedback`

Set Global Brightness

```
smhub-ambilightctl brightness 128
```

Night Mode

```
# Enable (default dim cap)
smhub-ambilightctl night on

# Enable with custom dim cap
```

```
smhub-ambilightctl night on 30

# Disable
smhub-ambilightctl night off
```

Status & List

```
smhub-ambilightctl status
smhub-ambilightctl list
```

□□ IR Control (Infrared)

Daemon: `smhub-ir-daemon`
CLI: `smhub-irctl`
Socket: `/run/smhub-ir.sock`

CLI Commands

Status

```
smhub-irctl status
```

Learn a Command

```
# Learn with default timeout
smhub-irctl learn tv_power

# Learn with custom timeout (seconds)
smhub-irctl -t 20 learn ac_mode
```

Send a Saved Command

```
smhub-irctl send tv_power
```

One-shot Capture (No Save)

```
smhub-irctl receive
```

List / Delete Commands

```
smhub-irctl list
smhub-irctl delete tv_power
```

Send Raw Data

```
smhub-irctl send_raw /tmp/ir_data.json
```

Use Custom Socket Path

```
smhub-irctl -s /run/smhub-ir.sock status
```

□□ Buzzer Control

Daemon: `smhub-buzzer-daemon`
CLI: `smhub-buzzerctl`
Socket: `/run/smhub-buzzer.sock`

Built-in Sounds

- `click`
- `success`
- `startup`
- `shutdown`
- `warning`
- `error`

CLI Commands

Play Built-in Sound

```
smhub-buzzerctl play builtin success
smhub-buzzerctl play builtin warning -p 8
```

Play RTTTL Melody

```
smhub-buzzerctl play rtttl "Melody:d=4,o=5,b=120:c,d,e,f,g"
```

Stop Playback

```
smhub-buzzerctl stop
smhub-buzzerctl stop -q   # stop + clear queue
```

Mute / Unmute

```
smhub-buzzerctl mute 5
smhub-buzzerctl mute 5 -m 300   # mute for 5 minutes
smhub-buzzerctl unmute
```

Status & List

```
smhub-buzzerctl status
smhub-buzzerctl list
```

□□ Advanced: IPC JSON via Unix Sockets

All peripherals support newline-delimited JSON over a Unix socket. Use this for automation scripts or integrations.

Ambilight (example)

```
echo '{"cmd":"color","r":255,"g":0,"b":0}' | socat - UNIX-CONNECT:/run/smhub-ambilight.sock
```

IR (example)

```
echo '{"cmd":"status"}' | socat - UNIX-CONNECT:/run/smhub-ir.sock
```

Buzzer (example)

```
echo '{"cmd":"list"}' | socat - UNIX-CONNECT:/run/smhub-buzzer.sock
```

Revision #1
Created 2026-02-28 21:26:36 UTC by Support3
Updated 2026-02-28 21:33:27 UTC by Support3