Conch Shell Phone — Quick Guide

Wiring

Connect a momentary push-button switch: - One leg to **GPIO 17** (physical pin 11) - Other leg to **GND** (physical pin 9) - No resistor needed

First Time Setup

```
cd ~
git clone https://github.com/stueydubs/niko-s-project.git
cd niko-s-project/conch
bash install.sh
```

Put your audio files in the `audio/` folder: - `ring.mp3` (the ringing sound) - `01.mp3` through `30.mp3` (the 30 tracks)

Testing in Thonny

Start testing

1. Stop the background service first:

```
sudo systemctl stop conch.service
```

2. Kill any leftover audio:

```
killall vlc
```

3. Open Thonny, File > Open > `/home/nikoniko/niko-s-project/conch/conch.py`
4. Click the green Run button

While testing

- Press **Enter** in the Shell panel to answer the phone (stop ring, play track)
- Press **Enter** again to stop the current track
- The silence timers are long (15-25 min) — to speed up testing, temporarily change line 31:

```
{"file": "01.mp3", "silence_min": 0.05, "silence_max": 0.05},
```

(0.05 minutes = 3 seconds)

Stop testing

- Click the Stop button in Thonny, or press Ctrl+C
- If audio keeps playing after stopping:

```
killall vlc
```

When done testing

- Change any silence timers back to the real values
- Save the file

Running for Real (Production)

Start the service

```
sudo systemctl start conch.service
```

The conch will now run automatically, including after reboots and power cuts.

Stop the service

```
sudo systemctl stop conch.service
killall vlc
```

Restart after editing conch.py

```
sudo systemctl restart conch.service
```

Check if it's running

```
sudo systemctl status conch.service
```

Watch the logs live

```
tail -f /home/nikoniko/niko-s-project/conch/conch.log
```

See which track is next

```
cat /home/nikoniko/niko-s-project/conch/track_state.txt
```

(0 = track 01, 1 = track 02, etc.)

Reset back to track 1

```
echo 0 > /home/nikoniko/niko-s-project/conch/track_state.txt
sudo systemctl restart conch.service
```

Pulling Updates from GitHub

If changes are pushed from another computer:

```
cd /home/nikoniko/niko-s-project && git reset --hard && git pull
sudo systemctl restart conch.service
```

Editing Silence Timers

Open `conch.py` in Thonny or nano. Find `TRACK_CONFIG` (around line 31). Each track has:

```
{"file": "01.mp3", "silence_min": 15, "silence_max": 20},
```

- `silence_min` and `silence_max` are in **minutes**
- The actual silence is a random value between min and max
- After editing, restart the service

Troubleshooting

Problem	Fix
Audio keeps playing after stopping script	`killall vlc`
Service won't start	`sudo journalctl -u conch.service` to see errors
No sound at all	Check audio output: `raspi-config` > System Options > Audio
Ring plays but no tracks	Check audio files exist in `audio/` folder
Script crashes on startup	Check the log: `tail -20 /home/nikoniko/niko-s-project/conch/conch.log`
Want to start fresh	`echo 0 > /home/nikoniko/niko-s-project/conch/track_state.t`
Forgot which track we're on	`cat /home/nikoniko/niko-s-project/conch/track_state.t`

How It Works

1. **Silent** — waits for a random time (from the track's silence config)
2. **Ringing** — plays `ring.mp3` on loop until someone picks up (presses button)
3. **Playing** — plays the current track, then advances to the next one
4. Back to **Silent** — repeat forever

The current track number is saved to disk, so it survives power cuts and reboots.