

## First approach - Call script from output\_alsa.c

The first approach is like yours, but modified.

Modify the file `output_alsa.c` from <https://github.com/ralph-irving/squeezelite>

Declare this variable and string in the beginning of the file, or in the block below:

```
char filter_cmd[64];
#define FILTER_CMD_PATH "/usr/local/bin/update_filter.py"
```

This replaces the block of code that you have added:

```
// CamillaDSP engine integration START

UNLOCK;

sprintf(filter_cmd, "%s %d", FILTER_CMD_PATH, sample_rate);
int system_return = system(filter_cmd);

if (system_return == 0) {
    LOG_INFO("Sample rate changed to %u", sample_rate);
}
else {
    LOG_ERROR("Could not set sample rate %u", sample_rate);
}

// CamillaDSP engine integration STOP
```

The program `update_filter.py` is called from the code above:

```
#!/usr/bin/python3

from websocket import create_connection
import sys

sample_rate = sys.argv[1]

ws = create_connection("ws://127.0.0.1:1234")
ws.send(f"setconfigname:/etc/camilladsp_{sample_rate}.yml")
ws.send("reload")
ws.close()
```

## Second approach - Use a named pipe

If you let `squeezelite` log to a named pipe, you can extract the sample rate from the pipe, with minimal changes to `output_alsa.c`. The only change that should be necessary is the `UNLOCK` command from the first approach above. I tried a lot without the command, which did not work (Capture error: ALSA function 'snd\_pcm\_hw\_params\_set\_rate' failed with error 'EINVAL: Invalid argument'). Actually I've never tried with it, but I think the `UNLOCK` command is required to make it work.

Create a named pipe (e.g. `bash: mkfifo /var/run/squeezelog`). Alter mode and ownership according to your needs. For squeezelite to write to the pipe, start squeezelite with these parameters `-d output=info -f <path to pipe>` (use other parameters as you like).

This program will read from the pipe and call `update_filter.py`:

```
#!/usr/bin/python3
# Read the sample rate used by SqueezeLite through a named pipe

import os
import stat
import subprocess

pipe = '/var/run/squeezelog'

with open(pipe, 'r') as fifo:
    for line in fifo:
        if "opening device at:" in line:
            word_list = line.split()
            sample_rate = word_list[-1]
            subprocess.call(f"/usr/local/bin/update_filter.py
{sample_rate}", shell=True)
```

If you want log messages to be written to file, it should be easy to append the lines to a text file as you read the from the pipe.