

Some Notes about Sound on the Raspberry Pi

The sound configuration on the Raspberry Pi can be quite confusing. Here is an attempt at clarification.

First, the default sound system has a stereo speaker output, but no microphone input. The output can be directed to the 3.4 mm pin connector next to the HDMI output or to a line in the HDMI connector. By default the HDMI connector is not used for sound output since it might also be used for DVI output which has no sound line. A configuration change is necessary to redirect the sound from the 3.4 mm pin when video with sound is to be sent through the HDMI connector.

The following lines in `/boot/config.txt` deal with this possibility:

```
# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# DMT (computer monitor) modes
# hdmi_drive=2
```

I normally use a HDMI-to-VGA converter that provides a jack for the sound output so I want the sound going out the HDMI connector and therefore remove the “#” from the start of the `hdmi_drive=2` line in the above excerpt.

The sound configuration can also be changed through the desktop sound icon and through the desktop menu. If you open Menu/Preferences/Audio Device Settings, a window opens which lets you choose a “Sound Card” (a.k.a. sound device) from a selection widget at the top. A Select Controls Button opens a window with check boxes for each control on that sound device. When checked, the Playback, Capture, and Switches tabs show the controls that can then be adjusted as desired. When a USB sound device, like a GoMic is plugged in, a “Sampson GoMic” sound device may be selected and its controls exposed. The GoMic has a speaker jack so it has both Speaker and Capture controls and a switch that make both active at once.

The top line of the desktop has a speaker icon which lets you select the volume, or if right-clicked, lets you switch between built-in sound going out the 3.4 mm pin (Analog) connector or the HDMI, or instead activate the USB sound device (my GoMic). A bottom choice opens the Audio Device Settings window mentioned above.

When under command line control, the setting above in `/boot/config.txt` selects where the built-in sound is sent, but it gets more complicated when there are multiple sound devices. In any case, the behavior is controlled by a file `/home/pi/.asoundrc` containing the following lines when the built-in devices is the only one available:

```
pcm.!default {
    type hw
    card 0
}

ctl.!default {
    type hw
    card 0
}
```

Access to the built-in sound driver is via entries in `/dev/snd/*` which has

```
ls -l /dev/snd/*
crw-rw----+ 1 root audio 116,  0 Jul 19 15:32 /dev/snd/controlC0
crw-rw----+ 1 root audio 116, 16 Jul 19 16:26 /dev/snd/pcmC0D0p
crw-rw----+ 1 root audio 116, 17 Jul 19 15:32 /dev/snd/pcmC0D1p
crw-rw----+ 1 root audio 116,  1 Jul 19 15:32 /dev/snd/seq
crw-rw----+ 1 root audio 116, 33 Jul 19 15:32 /dev/snd/timer
```

These appear when the `snd_bcm2835` kernel module is loaded (usually the case). The `C0` in the filenames of the first three listings here refer to “Card 0”. `D0` and `D1` refer to the left and right channels of stereo output and the `p` at the end of `pcmC0D0p` and `pcmC0D1p` indicates playback. The `controlC0` entry allows access to volume controls, etc., for that card.

To play a file on the built-in sound system, we can use the `aplay` command as follows

```
aplay ~/Programming/Java/originals/sounds/startup.wav
```

This works for a number of simple sound formats and sampling speeds only because the build-in sound driver can make the necessary conversions.

Conversions must be explicitly specified for the GoMic USB device. When it is plugged in, the `/dev/snd` directory gains new entries:

```
ls -l /dev/snd/*
crw-rw----+ 1 root audio 116,  0 Jul 19 15:17 /dev/snd/controlC0
crw-rw----+ 1 root audio 116, 32 Jul 19 15:17 /dev/snd/controlC1
crw-rw----+ 1 root audio 116, 16 Jul 19 16:26 /dev/snd/pcmC0D0p
crw-rw----+ 1 root audio 116, 17 Jul 19 15:17 /dev/snd/pcmC0D1p
crw-rw----+ 1 root audio 116, 56 Jul 19 15:17 /dev/snd/pcmC1D0c
crw-rw----+ 1 root audio 116, 48 Jul 19 16:34 /dev/snd/pcmC1D0p
crw-rw----+ 1 root audio 116,  1 Jul 19 15:17 /dev/snd/seq
crw-rw----+ 1 root audio 116, 33 Jul 19 15:17 /dev/snd/timer
```

We now have Card 1 and its ..c1.. entries. The GoMic has a capture entry pcmC1D0c and a single playback entry pcmC1D0p.

If we now try to play a file with the GoMic set as default by right clicking on the speaker icon at the top right and clicking on GoMic, we get an error message

```
aplay startup.wav
Playing WAVE 'startup.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono
aplay: set_params:1239: Channels count non available
```

The GoMic needs help with conversions. This is fixed by adding lines to ~/.asoundrc as follows:

```
pcm.!default {
    type hw
    card 1
}

ctl.!default {
    type hw
    card 1
}

pcm.GoMic {
    type asym
    playback.pcm {
        type plug
        slave.pcm "hw:1,0"
    }
    capture.pcm {
        type plug
        slave.pcm "hw:1,0"
    }
}
```

We then use the aplay command with the -D option as follows:

```
aplay -D GoMic startup.wav
Playing WAVE 'startup.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono
```

which works because the GoMic entry is invoked which adds the necessary conversions.

A useful program is speaker-test:

```
speaker-test -D plughw:0,0 -c 2 -t wav
```

sends "Front Left" and "Front Right" messages to HDMI speakers separately.

```
speaker-test -D plughw:1,0 -c 2 -t wav
```

sends "Front Left" and "Front Right" messages to GoMic earphone outputs separately.

This last could also be done using

```
speaker-test -D GoMic -c 2 -t wav
```