# Configuring the AudioMoth USB Microphone from the Command Line

**theteam@openacousticdevices.info**

**21st July 2024**

The AudioMoth USB Microphone is normally configured using a desktop app after the switch has manually been moved to the USB/OFF position. This application note describes (i) an update in version 1.3.0 of the AudioMoth-USB-Firmware that also supports temporarily changing the configuration settings whilst the device is acting as a microphone with the switch in the DEFAULT and CUSTOM positions, and (ii) a command line tool that allows AudioMoth USB Microphones to be temporarily reconfigured in this way.

## 1   Firmware Update

Previous versions of the AudioMoth-USB-Microphone firmware enumerate the device as a USB microphone when the switch is in the DEFAULT or CUSTOM positions, and as a USB HID device when in the USB/OFF position. As with other variants of the AudioMoth firmware, the USB/OFF position is used to configure and re-flash the device.

From version 1.3.0 onwards, when the switch is in the DEFAULT or CUSTOM positions, the AudioMoth-USB-Microphone firmware enumerates the device as a composite USB device with both USB microphone and USB HID interfaces. This makes no difference to its use as a USB microphone, or to its normal use when the switch is in the USB/OFF position. However, the additional USB HID interface described here provides an alternative mechanism to temporarily change the configuration settings without having to manually adjust the switch position.

## 2   Command Line Tool

The AudioMoth-USB-Microphone command line tool provides a mechanism to temporarily change the configuration of the AudioMoth USB Microphone whilst it is enumerated as a microphone. It is called with one or more arguments (in lower or upper case). The first argument specifies one of seven commands given by: **LIST**, **CONFIG**, **UPDATE**, **LED**, **READ**, **RESTORE**, **PERSIST**, **FIRMWARE** and **BOOTLOADER**[1].

---

[1]Note that the **FIRMWARE** and **BOOTLOADER** commands require firmware version 1.3.1 and greater.

**LIST** returns a list of the connected AudioMoth USB Microphones along with their device IDs. The **LIST** command does not accept any additional arguments.

**CONFIG** sets a temporary configuration. The AudioMoth will ignore the current switch position and set the configuration specified by the settings that follow (see Section 2.1). All unspecified settings are set to their default value. The AudioMoth will flash both LED to show that it is applying the temporary configuration[2]. If the sample rate is changed, the AudioMoth will disconnect and re-enumerate as USB microphones do not support dynamic changes of sample rate.

**UPDATE** sets a temporary configuration using just the settings that control the gain of AudioMoth's software-controlled amplifier (specifically, **GAIN** and **LOWGAINRANGE** as described in Section 2.1). All other existing settings are unchanged. If the AudioMoth is not already flashing both LED to indicate that a temporary configuration is being applied, it will show a constant LED alongside the single flashing LED whenever the gain settings differ from those set in the persistent configuration. For example, if the switch is in the CUSTOM position, the red LED will flash and a constant green LED will show whenever the gain settings vary from those specified in the persistent configuration.

**LED** enables and disables the AudioMoth LED without changing any other settings. The command is followed by a single argument with allowable options being, *true*, *false*, *on*, *off*, *1* or *0*.

**READ** returns a list of the connected AudioMoth USB Microphones along with their current configuration

---

[2]In normal operation, the red LED flashes to show that the switch is in the CUSTOM position and all configuration settings are being applied, and the green LED flashes to show that the switch is in the DEFAULT position and just the gain and sample rate settings are being applied.
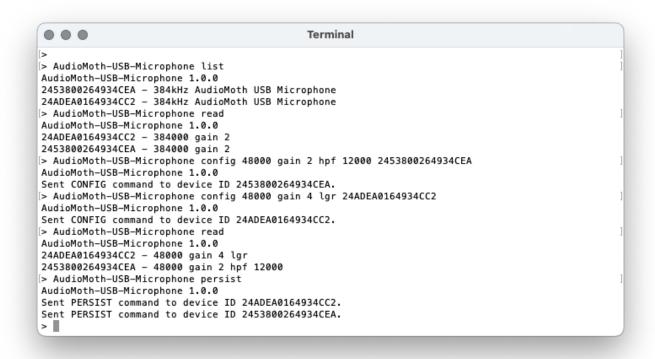
**Figure 1:** *Example use of the AudioMoth-USB-Microphone command line tool to configure two microphones.*

settings. The settings are presented in the format described in Section 2.1.

**RESTORE** returns the AudioMoth to normal operation using the configuration that has been persistently configured on the device.

**PERSIST** writes the current configuration settings to persistent storage within the AudioMoth.

**FIRMWARE** returns a list of the connected AudioMoth USB Microphones along with their current installed firmware version and description.

**BOOTLOADER** causes the AudioMoth to enter serial USB bootloader mode. The device firmware may then be updated using either the AudioMoth Flash App[3] or the EFM32-Flash command line tool[4]. Note that entering the serial USB bootloader causes the AudioMoth USB Microphone to re-enumerate as a USB serial device, and it will no longer be seen by this command line tool. The USB serial bootloader is factory installed on the AudioMoth's Wonder Gecko processor and is maintained by Silicon Labs[5].

---

[3]https://www.openacousticdevices.info/applications
[4]https://github.com/OpenAcousticDevices/EFM32-Flash
[5]https://www.silabs.com/documents/public/application-notes/an0042-efm32-usb-uart-bootloader.pdf

## 2.1 Settings

The **CONFIG** command can be followed by several arguments that describe the settings to be used as the temporary configuration. The settings can be presented in any order, in lower or upper case, and when a setting is not specified, the default value is used.

The **sample rate** is specified with a numerical value with valid rates being *8000, 16000, 32000, 48000, 96000, 192000, 250000* and *384000*. The default value is *384000*.

**GAIN** or **G** specifies the gain setting. It must be followed by a numerical value from *0* to *4*. Table 1 shows the gain that is applied by AudioMoth's software-controlled amplifier for each numerical setting[6]. The default setting is *2*.

**LOWPASSFILTER** or **LPF** specifies the use of a low-pass filter. It must be followed by the filter frequency, specified as a numerical value between *0* and half of the sample rate, divisible by 100Hz. The default is no filter.

**HIGHPASSFILTER** or **HPF** specifies the use of a high-pass filter. It must be followed by the filter frequency, specified as a numerical value between *0* and half of the sample rate, divisible by 100Hz. The default is no filter.

---

[6]The software-controlled amplifier described here is preceded by a pre-amplifier with a fixed gain of 11.

| | | Low Gain Range | Normal Range |
|---|---|---|---|
| 0 | Low | 0.33 | 4.33 |
| 1 | Low-Medium | 0.55 | 7.00 |
| 2 | Medium | 1.00 | 15.00 |
| 3 | Medium-High | 1.67 | 25.05 |
| 4 | High | 2.00 | 30.00 |

**Table 1:** *Gain applied at each setting when low gain range and normal range are selected.*

**BANDPASSFILTER** or **BPF** specifies the use of a band-pass filter. It must be followed by the lower and upper filter frequencies. Both are specified as numerical values between *0* and half of the sample rate, divisible by 100Hz. The default is no filter.

**LOWGAINRANGE** or **LGR** specifies the use of the low gain range. Table 1 shows the gain that is applied by AudioMoth's software-controlled amplifier in each range. The default is to use the normal gain range.

**ENERGYSAVERMODE** or **ESM** specifies emulation of the AudioMoth energy saver mode. When running the standard AudioMoth firmware, energy saver mode uses less over-sampling to generate recordings of slightly lower quality while saving energy when the sample rate is 48kHz and less. The AudioMoth USB Microphone can emulate this reduction in recording quality. However, there is no corresponding reduction in energy consumption. The default is not to use energy saver mode.

**DISABLE48HZ** or **D48** disables the 48Hz DC blocking filter. This improved the low-frequency response of the microphone by reducing the corner frequency of the DC blocking filter from 48Hz to 8Hz.

When the **UPDATE** command is used, just the **GAIN** and **LOWGAINRANGE** settings are expected and any other settings will result in a parsing error.

## 2.2   Device ID

The arguments that follow the **CONFIG**, **UPDATE**, **LED**, **READ**, **RESTORE**, **PERSIST**, **FIRMWARE** and **BOOTLOADER** commands can also include a list of device IDs such that the command will only be sent to those specified devices. The device ID is given by a 64-bit, 16-character, hexadecimal number. The same device ID appears in the standard AudioMoth USB Microphone App and the ID of connected devices is shown by the **LIST** command. If no device IDs are specified, the commands will be sent to all connected AudioMoth USB Microphones.

## 2.3   Example

Figure 1 shows an example using the AudioMoth-USB-Live command line tool. The **LIST** command is first used to show the connected microphones and their configurations are read with the bfLIST command. Each microphone is then configured individually via its device ID, setting both to use a sample rate of 48kHz, with one in low gain range and one with a high-pass filter. The configurations are then confirmed with the **READ** command, before they are set persistently with the **PERSIST** command

# 3   Compiling from Source Code

The AudioMoth-USB-Microphone command line tool can also be built from source on macOS, Windows, Linux and Raspberry Pi machines by following the instructions in the GitHub repository[7].

On macOS, Linux and Raspberry Pi you can copy the resulting executable to `/usr/local/bin/` so it is immediately accessible from the terminal. On Windows copy the executable to a permanent location and add this location to the `PATH` variable.

When running on Linux and Raspberry Pi, it is necessary to add an additional `udev` rule to allow access to the USB HID interface. See the GitHub repository for full details. Note that this is in addition to the normal update of the `udev` rules to use the standard AudioMoth desktop tools as the AudioMoth USB Microphone uses a different PID and VID combination when enumerated as a USB microphone.

# 4   Installation

Pre-built installers are also available for all platforms[8]. On Windows, macOS and Linux, these will install pre-compiled executables that can be immediately called from the command line. The Raspberry Pi version automatically builds the binary from source files and then installs it. The Linux and Raspberry Pi installers are run from the command line with:

```
> sh AudioMothUSBMicrophoneSetup1.0.0.sh
```

and:

```
> sh AudioMothUSBMicrophoneBuild1.0.0.sh
```

As above, when running on Linux and Raspberry, it is necessary to add the additional `udev` rule to allow access to the USB HID interface. The Windows and macOS installer can simply be double-clicked to run them.

---

[7]https://github.com/OpenAcousticDevices/AudioMoth-USB-Microphone-Cmd
[8]https://github.com/OpenAcousticDevices/AudioMoth-USB-Microphone-Cmd/releases/tag/1.0.0