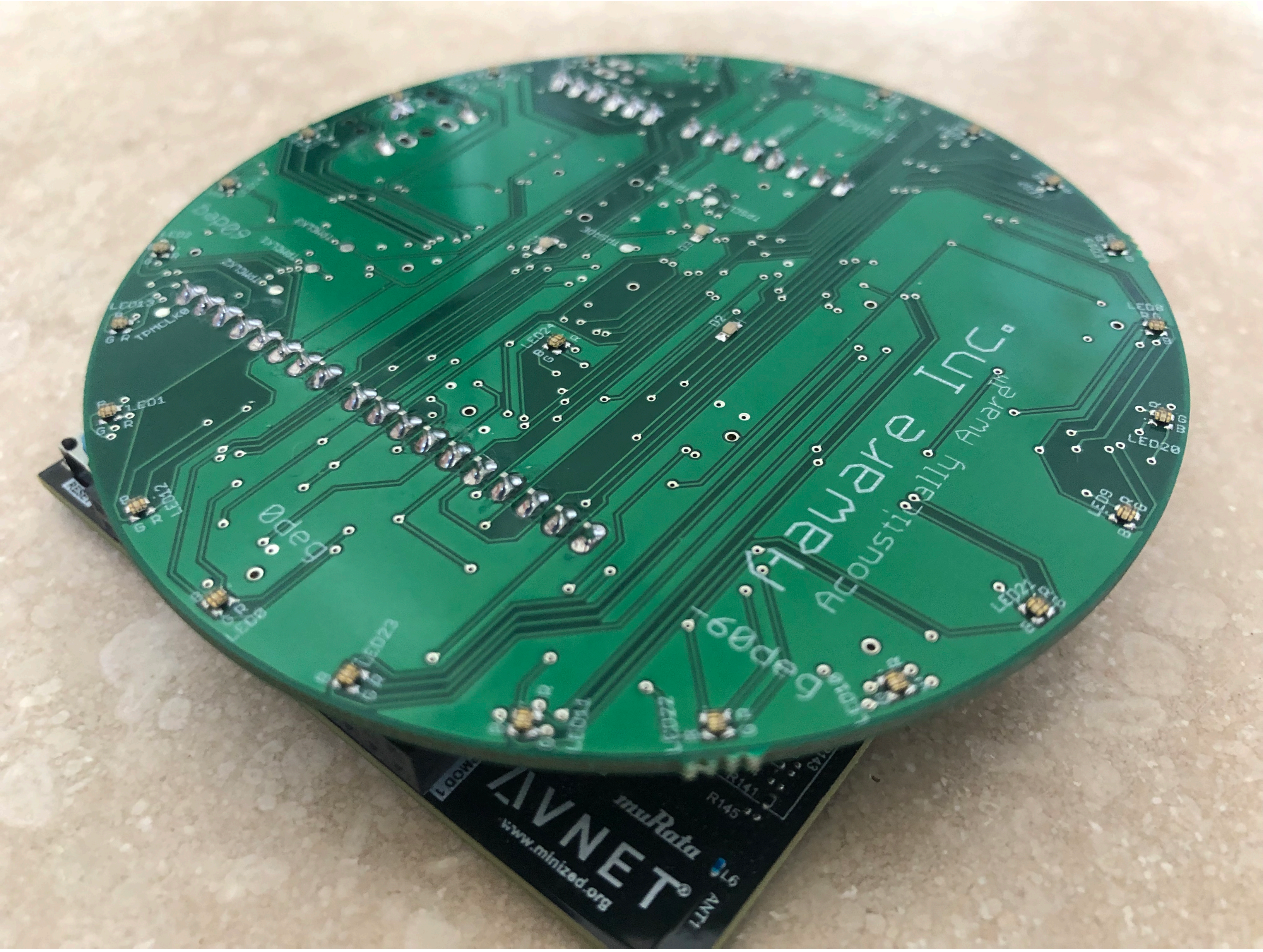
User Guide

AEV13-MZ

Version 2.0 • October 25, 2019



Aaware, Inc. - 20 S. Santa Cruz Ave. #300, Los Gatos, CA 95030, US
support@aaware.com - www.aaware.com

Trademarks

Acoustically Aware is a trademark of Aaware, Inc. All other trademarks or registered trademarks are the property of their respective owners.

Disclaimer

The information provided in this document is provided "as is" without warranty of any kind. Aaware disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Aaware be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Aaware or its suppliers have been advised of the possibility of such damages.

Document Lifetime

Aaware may occasionally update online documentation between releases of the related software. Consequently, if this document was not downloaded recently, it may not contain the most up-to-date information. Please send inquiries to support@aaware.com for the most current information.

Where to get help

Aaware support, product, and licensing information can be obtained as follows.

Product information — For documentation, release notes, software updates, or for information about Aaware products, licensing, and service, send inquiries to support@aaware.com.

Technical support — For technical support, go to http://zedboard.org/forums/aaware-voice-kit and post your support question or issue. Aaware will respond promptly in this forum to inquiries and issues.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send any feedback regarding this document to: info@aaware.com.

If you have issues, comments, or questions about specific information or procedures, please include the title and, if available, the part number, the revision, the page numbers, and any other details that will help us locate the subject that you are addressing.

Preface

Intended Audience

The Aaware embedded voice development platform is primarily intended for software engineers and product engineers who are familiar with Linux based embedded environments and wish to evaluate and integrate the Aaware embedded voice technology and to also use this platform to create a prototype product that has an embedded voice interface.

Style Conventions

The following style conventions are used in this document:

Bold

- Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system call, man pages
- Names of interface elements (such windows, dialog boxes, buttons, fields, and menus)
- Interface elements the user selects, clicks, presses, or types

`Courier`

- System output, such as an error message or script
- URLs, complete paths, filenames, prompts, and syntax when shown outside of running text

< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means “or”
{ }	Braces indicate content that you must specify (that is, x or y or z)
...	Ellipses indicate nonessential information omitted from the example

Table of Contents

Document History

Paper copies are valid only on the day they are printed. Contact the author if you are in any doubt about the accuracy of this document.

Revision History

This document has been revised by:

Revision Number	Revision Date	Summary of Changes	Author(s)
v .1	11/23/2017	Initial Template Created	JG
v .2	11/27/2017	Linux Management, AVS and Google Assistant interface	CE, JC
v 1.0	01/18/2018	First Complete User Guide	CE, JC, JG
v 1.1	01/19/2018	Consolidate Content	JC
v 2.0	11/08/2019	Embedded Voice update	JG, CE, JC

Reference Documents

Please see the following documents for more information:

Document Name	Version	Author
AEV13MZ Quick Start Guide	2.0	JG

Introduction

Purpose

The purpose of this User Guide is to explain how to use the AEV13-MZ to:

1. Demonstrate/evaluate the voice and sound AI technology from Aaware, Inc.
2. Configure its various features
3. Manage/debug its operation
4. Develop custom voice and sound applications based on Aaware's APIs

Scope

The primary scope of this User Guide will be to communicate the use of this Aaware embedded voice platform to software engineers who will access the Aaware platform and the audio application programming interface (API).

Using AEV13-MZ

Operation and Demo Overview

This section describes how to power up the platform and use the demo voice interface application. There are several demo configurations available and by default the system will boot and run one of these. You can reconfigure the system to run different demos - see Demo Configuration in the Administration section for details.

Powering Up

The AEV13-MZ platform has two micro-USB ports, J2 and J6, and both will power on the platform when connected to a power supply. However, they have different functions

- J2: has USB-UART terminal access to the AEV13-MZ Linux environment and also JTAG access.
- J6: provides auxiliary power and is required for powering J1, a type-A USB connector, for using peripherals such as a USB-Ethernet dongle or USB-flash storage.

Embedded Voice Interface Demos

Upon power-up, the platform enters a pre-configured embedded voice demo based on your initial request. The platform begins listening for the wake word “Hey Aware” and LEDs indicate the direction of voice arrival (see figure 1 below) After recognizing the wake word, the platform is listening for voice commands from that direction.

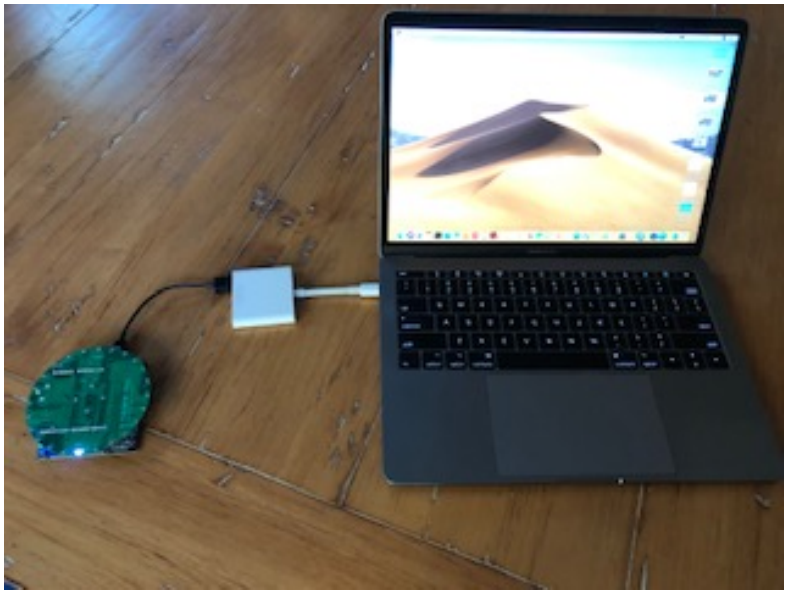


Figure 1: Laptop Power/Terminal and LED Direction of Voice Arrival

Wake Word Directionality / Azimuth

The default demos are setup to continuously listen for the wake word in 360 degree azimuth range. On detection, an LED sequence will occur resulting in an LED remaining on in the direction of the wake word. The PCB has labels indicating 0deg, 180deg, etc. which determines the orientation of the detection location metadata published to upstream applications.

Noise Robust Source Detection and Separation Overview

The AEV13MZ implements a complete acoustic source detection and separation solution that enables robust voice and sound detection in noisy environments. When an audio source is detected coming from a particular direction, separation and noise reduction algorithms are applied to remove interfering sources and background noise as much as possible. When a wake-word, e.g. “Hey Aware” is detected in one of these sources, audio and event messages are published so applications using Speech Recognition and Natural Language Processing can start listening and responding more accurately to the voice/acoustic source.

The default demo, for example, has an application that implements a smart light voice interface where you can control the LEDs on the board by commands like “Hey Aware, turn on the lights”, or “Hey Aware, set the bedroom light to green”.

User applications can be developed using standard interfaces that Aaware provides. Standard ALSA (Advanced Linux Sound Architecture) interfaces are provided to deliver “clean” audio in the direction of detection. This provides a familiar streaming data interface that supports common audio and voice applications. Detection events and metadata like location published using MQTT event messaging. User applications can subscribe to these messages either locally or over the network. More details on these interfaces are provided in the Administration section.

Audio Recording

The Aaware interface provides streaming of “clean” audio listening in the direction of the last wake word. To capture/record this audio to a .wav file, use the following standard ALSA Linux command:

```
arecord -r16000 -fS32_LE -c1 -d5 cap.wav
```

The **`-d5`** parameter specifies a capture duration of 5 seconds. See the Linux man page for **`arecord`** for more details.

Sending Audio Over Network

To send the streaming output over the network use **`sox`** and **`nc`**.

On the receiving end:

```
while true; do nc -l 2389 | play --buffer 32 -r16k -b16 \
-esigned-integer -L -c1 - traw -; sleep 1; done
```

Opens a connection on port 2389 and plays the received PCM stream to the default system audio output. You can also pipe the **`nc`** output to **`sox`** or any other PCM data consumer.

On the Aaware platform:

```
sox --buffer 32 -r16k -b16 -esigned-integer -L -c1 \
-talsa hw:aawaloop,0,0 -traw - | nc 10.0.1.21 2389
```

Reads the Aaware sound capture daemon streaming output using sox and pipes it to **`nc`**. In this case, the receiving host is at IP address 10.0.1.21 port 2389. See the Linux man pages for **`sox`**, **`play`**, and **`nc`** for more details.

Aaware Sound Output Interface

The Aaware development platform includes a stereo 24-bit audio DAC line output interface. On most versions of hardware, a 3.5mm stereo audio jack is provided as the output connector. The interface is an ALSA sound card which is set as the default (called **`aawsnksvol`**). Applications can use this standard interface to play audio to the output jack.

Playing Audio Files

Use the standard command line audio players **`aplay`** (from ALSA) and **`play`** (from SoX) to play .wav format audio files. For example:

```
aawadm@aawzsb2:~$ play /opt/slib/wgn_2ch24_32k.wav
/opt/slib/wgn_2ch24_32k.wav:
File Size: 2.88M
Bit Rate: 1.54M
Encoding: Signed PCM
Channels: 2 @ 24-bit
Samplerate: 32000Hz
Replaygain: off
Duration: 00:00:15.00

In:29.0% 00:00:04.35 [00:00:10.65] Out:139k [-=====|======] Hd:2.1 Clip:0
```

Setting the Volume

The output interface is an ALSA sound card (called **`aawsnksvol`**) that is set as the default output. It has a master output volume control accessible through ALSA as the parameter 'aawsnk'. Use **`amixer`** to inspect and set the volume, for example:

```
root@aawsd1:~# amixer set aawsnk 10%
Simple mixer control 'aawsnk',0
  Capabilities: volume
  Playback channels: Front Left - Front Right
  Capture channels: Front Left - Front Right
  Limits: 0 - 255
  Front Left: 26 [10%]
  Front Right: 26 [10%]
```

Linux and Aaware Platform Administration

This section describes basic administration of the Linux operating system and Aaware applications installed on your Aaware platform.

Overview

Aaware provides a standard Ubuntu Linux environment with the following features (as of the date of this document):

- Linux Kernel 4.14.0 from Xilinx with pre-installed:
    a. Xilinx peripheral drivers (flash,gpio,etc.)
    b. WiFi networking drivers
    c. Aaware drivers for audio capture and playback
    d. Aaware drivers for algorithm hardware acceleration
    e. Aaware drivers for LED control
- Ubuntu 18.04 LTS root file system:
    a. Basic audio tools like ALSA ,etc. (from standard Ubuntu repositories)
    b. Aaware sound capture software applications
    c. Development packages for C++ and Python speech/audio/ai-neural-net applications
- Uboot boot loader configured for Xilinx Zynq ARMv7 devices

Linux Network Management

Terminal Connection

A PC or Mac can connect (and optionally powered) by one of the micro-USB connectors on the platfrom. There are two micro-USB connectors, use J2, the one closest to the standard USB connector.

If you have connected the USB port to a laptop/USB, then you can use terminal software to log into the Aaware embedded Linux environment. The terminal serial settings must be `115200,8N1` (115200 baud rate, 8-bit, no parity bits, 1 stop bit). Once connected, you will be presented with a Linux login prompt:

```
port is         : /dev/ttyUSB1
  ...
  Terminal ready
  ...
Ubuntu 18.04.2 LTS aawmz0 ttyPS1
aawmz1 login:
```

The Administrative User: aawadm

As a general policy recommended by Ubuntu, one should operate in most cases as a non-root user. For the Aaware platform, the sound capture administrator user aawadm is the recommended account for using the platform. The default passwd is admaaw17. However, the root user is also setup with the same password but is disabled for remote login (accessible from aawadm with the sudo -i command).

```
aawmz0 login: aawadm
Password:
Last login: Thu Nov  7 16:25:21 PST 2019 on ttyPS1
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.14.0-aawos+ armv7l)
aawadm@aawmz0:~$ uname -a
Linux aawmz0 4.14.0-aawos+ #1 SMP PREEMPT Thu Oct 17 23:42:11 UTC 2019
armv7l armv7l armv7l GNU/Linux
aawadm@aawmz0:~#
```

WiFi Overview

The Aaware Linux platform is setup with a WiFi interface named **`wlan0`** using **`wpa_supplicant`** and the **`systemd-networkd`** method of managing networks in Debian/Ubuntu. The interface will automatically come up as shown by the ifconfig command, but likely will not be connected to an AP (access point) yet until you add your configuration:

```
awmz1:~$ ifconfig
lo             Link encap:Local Loopback
         inet addr:127.0.0.1 Mask:255.0.0.0
         UP LOOPBACK RUNNING MTU:65536 Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0
txqueuelen:0
         RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
wlan0    Link encap:Ethernet HWaddr a0:cc:2b:ff:54:62
         UP BROADCAST MULTICAST MTU:1500 Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:1 errors:0 dropped:0 overruns:0 carrier:0 collisions:0
txqueuelen:1000
         RX bytes:0 (0.0 B) TX bytes:330 (330.0 B)
```

Instructions to configure WiFi to connect to your AP (access point) are provided below.

Note that current Ubuntu versions manage both wpa_supplicant and the network using systemd, and the systemd-networkd method is different from past methods (i.e., ifupdown and /etc/network/interfaces are NOT used). This is already setup on your Aaware platform, but below are some references for more details using systemd-networkd and wpa_supplicant together:

- https://wiki.archlinux.org/index.php/systemd-networkd
- https://beaveris.me/systemd-networkd-with-roaming/72
- http://manpages.ubuntu.com/manpages/xenial/man5/wpa_supplicant.conf.5.html
- http://manpages.ubuntu.com/manpages/xenial/en/man8/systemd-networkd.8.html

WiFi Setup

The Aaware Linux platform is set up with a WiFi interface named wlan0 using standard wpa_supplicant and the systemd-networkd methods of managing networks in Debian/Ubuntu. The interface will automatically come up if your AP (access point) is added to the configuration. Instructions to configure WiFi to connect to your AP (access point) are provided here.

You can use the provided WiFi setup script, **`aawwifisetup.sh`**, and follow the prompts to configure the wpa_supplicant configuration on the platform.

Or you can manually edit the **`/etc/wpa_supplicant/wpa_supplicant-wlan0.conf`** file using a text editor (vi/vim is available on the platform). An example entry is shown below for a standard password protected network; just change the placeholder parameters like myssid to your real network values.

```
aawadm@aawmz1:~# sudo vim /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
...
network={
auth_alg=OPEN
key_mgmt=WPA-PSK
psk="mypassword"
ssid="myssid"
proto=RSN
mode=0
}
...
```

Optionally you can protect your access point password using **`wpa_passphrase`** and use the resulting psk in **`wpa_supplicant.conf`** instead of using the clear text password:

```
aawadm@aawmz1:~$ sudo wpa_passphrase myssid mypasswd
network={
ssid="myssid"
#psk="mypasswd"
psk=e39d75850c60012e6164b936d1fcfc3b5c96119026210727ba79081a9f4f8e85
}
```

Rescan for networks using the **`iw wlan0 scan`** command. This should automatically connect to your interface if it is in range.

```
root@aawmz1:~# sudo iw wlan0 scan | grep SSID
SSID: NETGEAR05
SSID: familyabchome
SSID: TurnKey Vacation Rentals
...
```

WiFi Troubleshooting

The standard ifconfig commands will work for checking status; however, the commands for controlling the interface are systemd services. The following wpa_supplicant@wlan0 example can be used to stop/start or restart the underlying WiFi interface in case of problems:

```
sudo systemctl restart wpa_supplicant@wlan0
```

The wpa_cli application can also be useful for rescanning and further debug of your WiFi wpa_supplicant configuration. Listing access points for example:

```
aawadm@aawmz1:/$ wpa_cli -i wlan0
wpa_cli v2.4
...
Interactive mode
> scan
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-RESULTS
> scan_results
bssid / frequency / signal level / flags / ssid
10:ff:ff:fe:57:bb 2437 -53 [WPA2-PSK-CCMP][WPS][ESS] NETGEAR94
```

Please see the wpa_supplicant documenation for more details.

Remote Login With SSH

Once WiFi is up, use secure shell to login as aawadm:

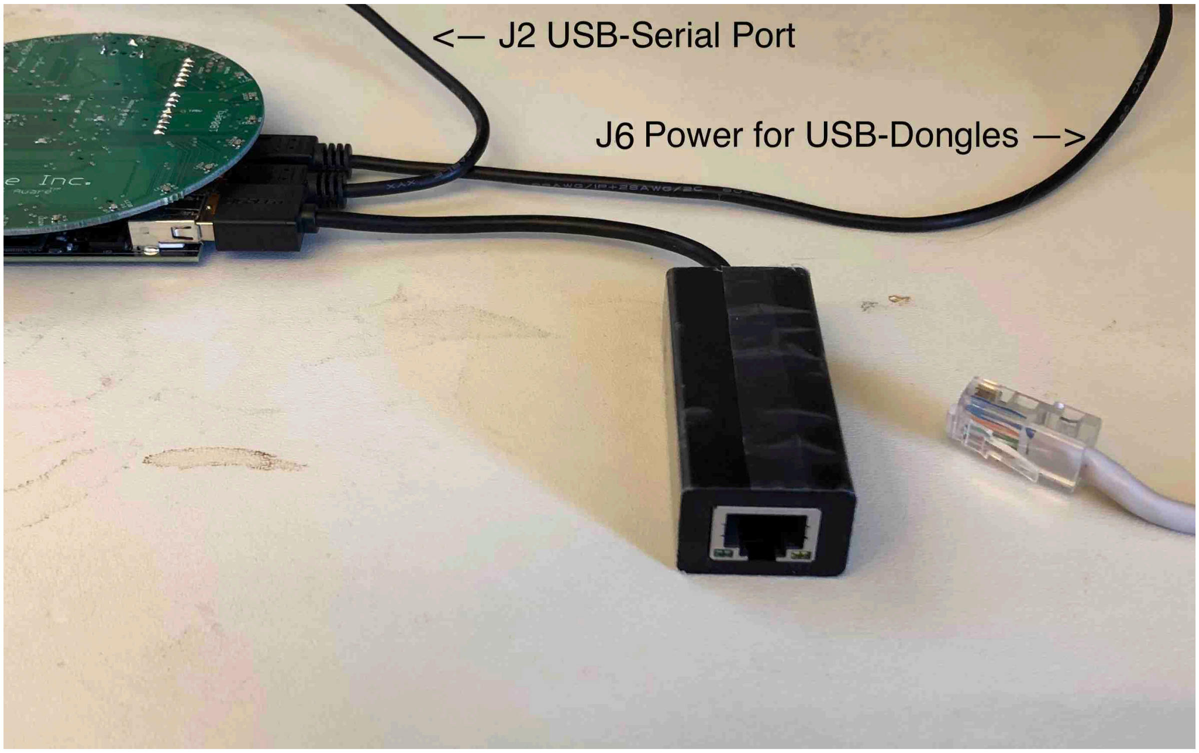
```
[laptop@vmel6 ~]$ ssh aawadm@192.168.0.102
aawadm@192.168.0.102's password:
Welcome to Ubuntu 18.04 (GNU/Linux 4.4.0-aawos+ armv7l)
...
Last login: Thu Aug 10 00:53:02 2017 from 192.168.0.106
aawadm@aawmz1:~$
```

USB Ethernet Interface to Host

To add an Ethernet interface, the AEV13MZ kernel has been built and tested with included drivers for the Amazon Basics USB-Ethernet dongle:

https://www.amazon.com/UGREEN-Ethernet-Adapter-Nintendo-Chromebook/dp/B00MYT481C

The J6 USB port must be powered via a standard micro-USB power cable for USB dongles to get power.



Once J6 power and the Ethernet dongle are plugged in, the Ethernet interface will become visible to the systemd networkctl command with an enx* name. The example results below show the status when no ethernet cable is plugged in:

```
aawadm@aawmz0:~$ networkctl
IDX            Link              TYPE        OPERATIONAL SETUP
  1            LO                Loopback    Carrier     Unmanaged
  2            Sit0              Sit         Off         Unmanaged
  3            wlan0             wlan        No Carrier  Configuring
  4            enx000ec6ac7501   ether       No Carrier  Configuring
4 links listed.
```

When an ethernet cable is plugged in with an active network and DHCP is provided, the interface should become active with an IP address, see below.

```
aawadm@aawmz0:~$ networkctl
IDX            Link              TYPE        OPERATIONAL SETUP
  1            LO                Loopback    Carrier     Unmanaged
  2            Sit0              Sit         Off         Unmanaged
  3            wlan0             wlan        No Carrier  Configuring
  4            enx000ec6ac7501   ether       Routable    Configuring
4 links listed.
```

The network status will also be visible using the ifconfig command, for example:

```
aawadm@aawmz0:~$ ifconfig
enx000ec6ac7501: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.20.32.180 netmask 255.255.254.0 broadcast 172.20.33.255
   inet6 fe80::20e:c6ff:feac:7501 prefixlen 64 scopeid 0x20<link>
   ether 00:0e:c6:ac:75:01 txqueuelen 1000 (Ethernet)
   RX packets 191 bytes 11976 (11.9 KB)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 34 bytes 4153 (4.1 KB)
   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
   inet 127.0.0.1 netmask 255.0.0.0
   inet6 ::1 prefixlen 128 scopeid 0x10<host>
   loop txqueuelen 1000 (Local Loopback)
   RX packets 40 bytes 2960 (2.9 KB)
   RX errors 0 dropped 0 overruns 0
   TX packets 40 bytes 2960 (2.9 KB)
   TX errors 0 dropped 0 overruns 0
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
   ether a0:cc:2b:ff:d4:14 txqueuelen 1000 (Ethernet)
   RX packets 0 bytes 0 (0.0 B)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 0 bytes 0 (0.0 B)
   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Once the ethernet interface is up, use secure shell to login as aawadm, e.g., for the above interface:

```
ssh aawadm@172.20.32.180
```

Installing General Linux Applications

The AEV13MZ provides a standard Ubuntu 18.04 environment with applications available via the Ubuntu repositories. Simply use `apt` to install your applications or development tools. Many of the most popular development environments like Python2 and 3 are already installed.

```
aawadm@aawmz1:~$ sudo apt install python
[sudo] password for aawadm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (2.7.15~rc-1).
python set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Managing Linux Services

Services are managed by systemd which used the systemctl command line tool for inspecting and changing service configurations. Below is an example listing of the active services:

```
root@aawmz1:~# systemctl status
● aawmz0
   State: running
     Jobs: 0 queued
   Failed: 0 units
    Since: Wed 1969-12-31 16:00:02 PST; 49 years 10 months ago
   CGroup: /
           ├─user.slice
           │ └─user-1000.slice
           │   ├─user@1000.service
           │   │ └─init.scope
           │   │   ├─14734 /lib/systemd/systemd --user
  ...
           ├─init.scope
           │ └─1 /sbin/init earlyprintk
           └─system.slice
             ├─alsa-state.service
             │ └─1419 /usr/sbin/alsactl -E HOME=/run/alsa -s -n 19
  ...
             ├─system-wpa_supplicant.slice
           │ └─wpa_supplicant@wlan0.service
             │   └─1412 /sbin/wpa_supplicant -c/etc/wpa_supplicant/
wpa_supplicant-wlan0.conf -Dnl80211,wext -iwlan0
             └─
               └─1316 /lib/systemd/systemd-networkd
```

Managing the Aaware Sound Capture Daemon

The Aaware sound capture daemon (aawscd) automatically starts at boot and begins listening for the wake word. It can be managed by systemctl status/start/stop, for example:

```
root@aawmz1:~# systemctl status aawscd
● aawscd.service - Aaware Sound Capture Daemon

  Loaded: loaded (/etc/systemd/system/aawscd.service; enabled; vendor
preset: enabled)

   Active: active (running) since Thu 2019-11-07 16:25:14 PST; 33min ago

 Main PID: 1436 (aawscd)

   CGroup: /system.slice/aawscd.service

           └─1436 /usr/local/bin/aawscd -c /etc/aawscd/aawscd.conf
```

Installing New Aaware Software Releases

Releases for the Aaware software include various system, application, and configuration packages and are released to customers on a case-by-case basis. They are provided as Debian packages. The preferred location for releases is in the /opt/aawrel directory. The process for installing a new release is as follows:

1. Copy the new .deb file you have been provided to the platform's /opt/aawrel directory.
2. Install with the dpkg -i command, e.g.,:

```
aawadm@aawmz0:~$ sudo dpkg -i /opt/aawrel/aawlight1804_1.2.5_armhf.deb

(Reading database ... 29832 files and directories currently installed.)

Preparing to unpack aawlight1804_1.2.5_armhf.deb ...

Unpacking aawlight (1.2.5) over (1.2.3) ...

Setting up aawlight (1.2.5) ...
```

Demo Configuration for Microphones and Wake Words

Different demo and array configurations are available and you can change between each by installing a configuration .deb package. They are located in /opt/aawrel/aawscdconf*.deb and installed with dpkg, i.e.,:

```
aawadm@aawmz0:~$ sudo dpkg -i /opt/aawrel/aawscdconf_20191107-13m4z-
heyaaware-snsr.deb
...
aawadm@aawmz0:~$ sudo systemctl restart aawscd
```

Note that new configurations are provided for:

1. Changing the array and number of microphones used, e.g., 13-mic circular endfire, 7-mic circular endfire, 13-mic broadside, etc.
2. Changing the wake words, e.g., “Hey Aware”, or “OK Google”, or both, or different vendors
3. Other configurations including listening directions, noise directions, AEC, etc.

Please contact Aaware for more information on various configuration options.

Troubleshooting and Diagnostics

There is a diagnostics utility aawdiag_mic.sh that checks the health of all 13 microphones on the array and will detect various failures that can be caused by water, air pressure, or shock damage. Running the diagnostic will return a PASS/FAIL message, for example:

```
aawadm@aawmz1:~$ sudo aawdiag_mic.sh
PASS
```

MQTT Messaging Interface

The sound capture daemon provides status and control via MQTT messaging. **`aawsc_wwclient.py`** is a Python script (located in /usr/local/bin) that opens a connection to the Aaware sound capture daemon and listens for wake word detection events. This script can be used as the basis for user applications that need notification of wake word events. Sample output:

```
Detected alexa in direction 5 at Thu Jul 20 06:40:08 2017
```