

Table of Contents

Table of Contents	1
Introduction	1
Setting up AVB4Linux	1
Running AVB4Linux	2
Troubleshooting AVB4Linux	4
Not Sure How to Get MAC Addresses	4
PTP isn't Listed as a Module after running Modprobe	8
No Talker Information in the Stream	8
Computer can't Connect to Wifi	9
The Kernel Updates Itself	9
General Debugging Advice	9
Sending Audio via AVB to another Room	9
AVB on Beaglebone	10

Introduction

This is a documentation of everything we did during our internship at SonicArts so that any poor souls trying to recreate our work have something to go off of.

Setting up AVB4Linux

The instructions below assume you're using the same setup as we did. We had an Intel i210 (obviously), as well as a MOTU and a Netgear. Our computer was running Ubuntu 20.04 and the kernel version was 5.15.0-67-generic. The Ubuntu and kernel versions probably aren't that important so long as you don't switch to a different version post-installation (trust me, doing this will cause everything to break).

To begin with, ensure you have run *sudo apt-get update* and *sudo apt-get upgrade*. Also make sure you have the following installed (given that we weren't very good about documenting our process, this list may not be exhaustive):

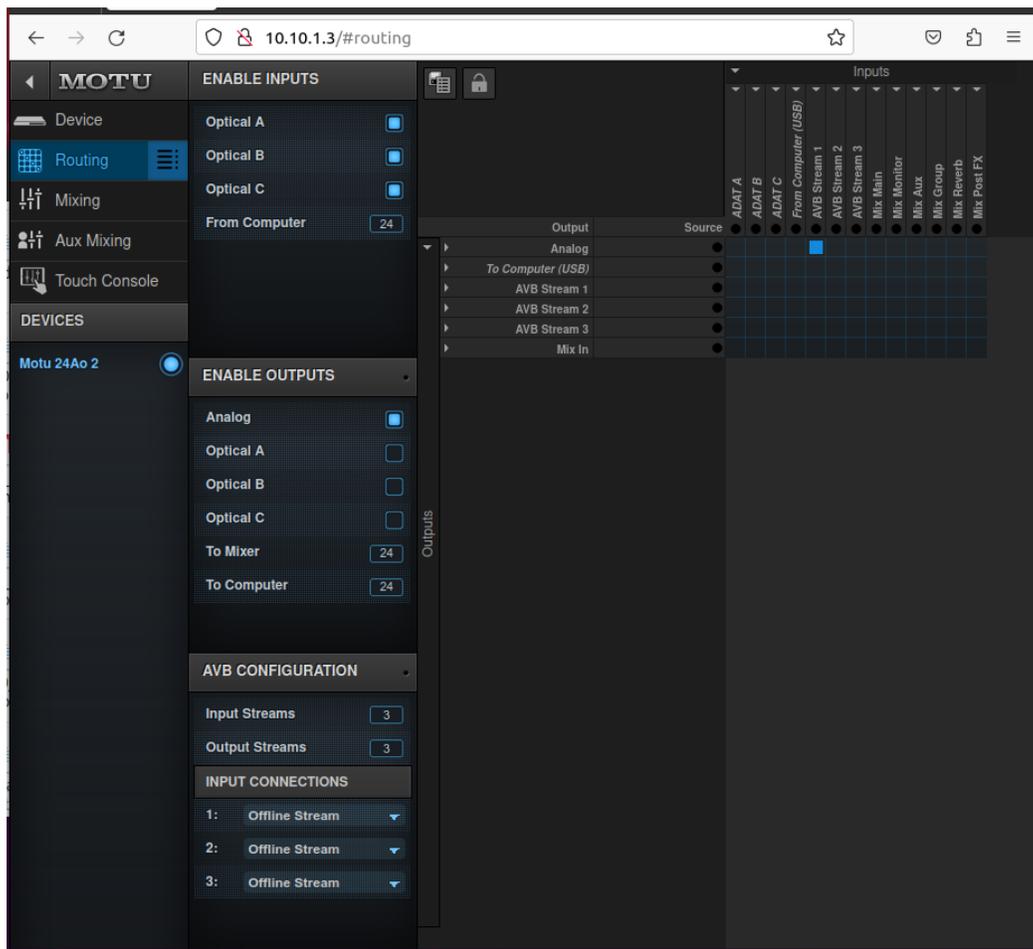
- net-tools
- build-essential
- git
- gcc
- make
- cmake

You may also need to install Linux PTP. We're not sure if it's actually necessary, but we never tried running AVB4Linux without it. The installation instructions can be found [here](#) (you can ignore everything that comes after the section on installation), but you're welcome to try leaving Linux PTP out.

Moving on to the installation of [AVB4Linux](#) (finally, right?), the README for the directory pretty much covers everything you need to know. However, when configuring the submodules, before following the instructions in the README, you'll want to first replace the url in `.gitmodules` with <https://github.com/jdkofinoff/jdksavdecc-c.git>, then run `git submodule sync`. You'll also want to edit the MAC addresses in `kernel_module/igb/avb-config.h` before building the directory (if you edit the MAC addresses later on, always make sure to rebuild the directory after doing so). Furthermore, make sure to keep the "ul" at the end of each address. If you're having trouble finding the MAC addresses, you can check the troubleshooting section below.

Running AVB4Linux

Before running AVB4Linux, make sure your MOTU's routing is correct. This wasn't something I dealt with much (and therefore know very little about), so I've included a picture of our routing. If you need more help, you'll have to ask someone else at SonicArts.



You'll also need to set up a subnet. Don't ask me how or why, because I don't know. Again, there are plenty of lovely people at SonicArts who you can ask instead.

Alright, moving on to actually running AVB4Linux. Surprise surprise, you can just follow the instructions given in the README. If you're having trouble finding the ethernet interface name, check out the bit on finding MAC addresses under the Troubleshooting section, where I discuss how to find the name of your i210.

If everything is set up correctly, when you run `avb_up.sh`, this is what you should see in the terminal:

```
rmmod: ERROR: Module igb_avb is not currently loaded
driver: igb_avb
version: 5.3.2_AVB
firmware-version: 3.20, 0x80000573
expansion-rom-version:
bus-info: 0000:04:00.0
supports-statistics: yes
supports-test: yes
supports-EEPROM-access: yes
supports-register-dump: yes
supports-priv-flags: no
Starting daemons on enp4s0
INFO : GTP [13:12:10:620] gTP starting
ERROR : GTP [13:12:10:621] Using clock device: /dev/ptp0
STATUS : GTP [13:12:10:621] Starting PDelay
process_events()
ERROR : GTP [13:12:10:717] Error (TX) timestamping PDelay request, error=-72
ERROR : GTP [13:12:11:781] Error (TX) timestamping PDelay request, error=-72
STATUS : GTP [13:12:12:470] LINKUP
STATUS : GTP [13:12:12:470] Link Speed: 1000000 kb/sec
STATUS : GTP [13:12:13:514] AsCapable: Enabled
STATUS : GTP [13:12:13:621] *** Announce Timeout Expired - Becoming Master
STATUS : GTP [13:12:13:621] New Grandmaster "1C:FD:08:FF:79:72:CB" (previous "00:00:00:00:00:00:00")
index=5
rc = 0, mrp_register_domain
rc = 0, mrp_join_vlan
Setting samplerate to 48000.
PARSE: C=6,P=3,V=0002
PARSE: I=0002

VNE 0002 R=000000000000 VN/MT

VNE 0002 R=000000000000 VN/MT
unhandled from mrpd
SJO T:S=0001f20157610000,A=91e0f0009c22,V=0002,Z=224,I=1,P=112,L=137942 R=6ccdd6c3ffe4 VO/IN

talker:
mode JOIN
stream: 0001f20157610000
dst_mac: 91e0f0009c22
vid: 2
SJO T:S=0001f20157610000,A=91e0f0009c22,V=0002,Z=224,I=1,P=112,L=137942 R=6ccdd6c3ffe4 VO/IN

VJO 0002 R=6ccdd6c3ffe4 QA/IN

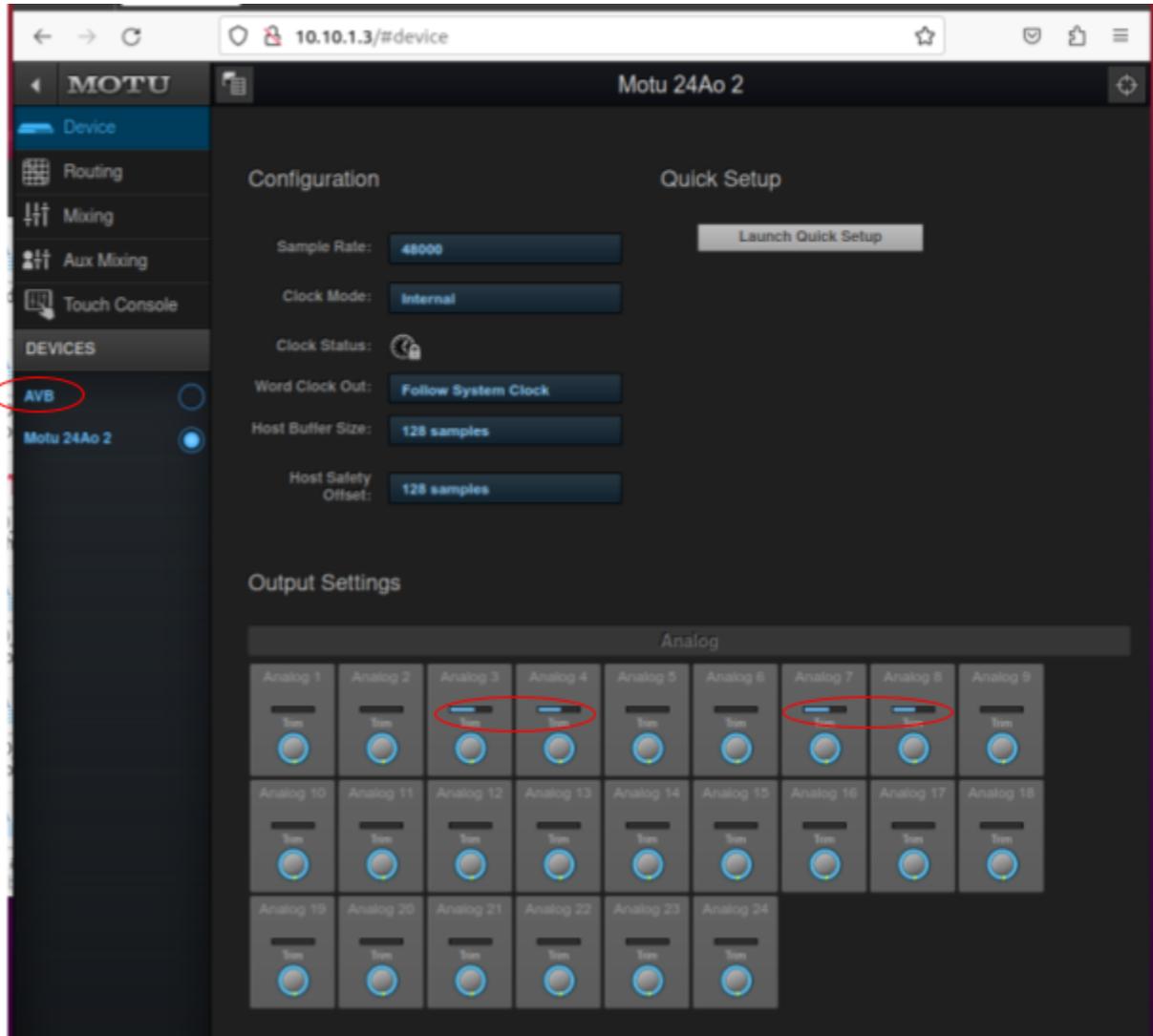
VJO 0002 R=6ccdd6c3ffe4 QA/IN
unhandled from mrpd
VJO 0002 R=6ccdd6c3ffe4 QA/IN

VJO 0002 R=6ccdd6c3ffe4 QA/IN
unhandled from mrpd
Done setting samplerate.

Mac: 1c-fd-08-79-72-cb
S+L:L=0001f20157610000, D=2
rc = 0, done sending ready
rc = 0, advertising stream
PARSE: L=0001f20157610000, D=2
PARSE: S=1cfd087972cb0000,A=91e0f000fe00,V=0002,Z=224,I=1,P=112,L=95
SNE L:D=2,S=0001f20157610000 R=000000000000 VN/MT

EVENT -- GTP -- TO -- 0001f20157610000 -- with state ready
```

Additionally, from the MOTU's web interface, you should be able to see two things: a device listed as AVB and flickering blue bars under some of your analog outputs (see picture below).



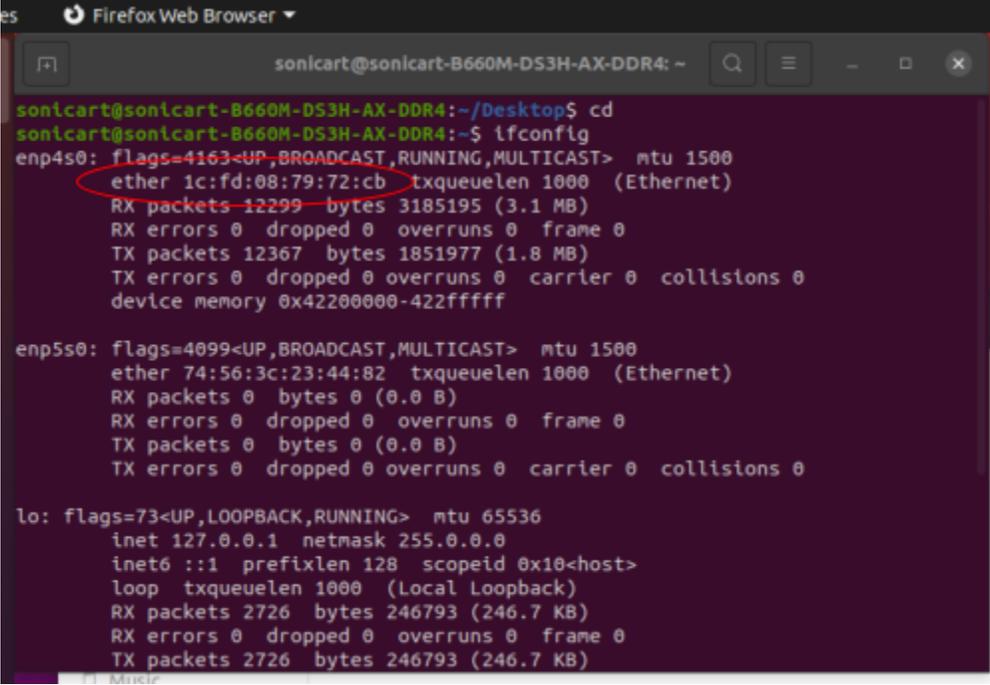
If that's the case, congratulations, it works! If not, first make sure you're actually playing some sort of audio on your computer (I may or may not have forgotten to do this and wasted an hour trying to troubleshoot a nonexistent problem), then check the troubleshooting section. If AVB4Linux still isn't working after all that, then you're on your own. Good luck!

Troubleshooting AVB4Linux

Not Sure How to Get MAC Addresses

I scream, you scream, we all scream because no one likes dealing with MAC addresses. Anyways, as the AVB4Linux README states, you'll need to edit the values of `OWN_MAC`, `AVB_DEVICE_SOURCE_MAC` and `AVB_DEVICE_TALKER_MAC_BASE`. `OWN_MAC` should be the MAC address of your i210, and you can find this by running *ifconfig*. Your i210 will be listed as `enp4s0` or something similar, and the MAC address is the funky-looking string of characters following

“ether.” You can reference the picture below if I’m not making any sense. Just a heads up, you might see multiple `enp*s0` listed when you run `ifconfig` (note both `enp4s0` and `enp5s0` in the picture below). The easiest way to figure out which is the i210 is to connect the i210 to your Netgear (after you’ve set up a subnet) and rerun `ifconfig`. Whichever `enp*s0` now has an ip address is your i210.

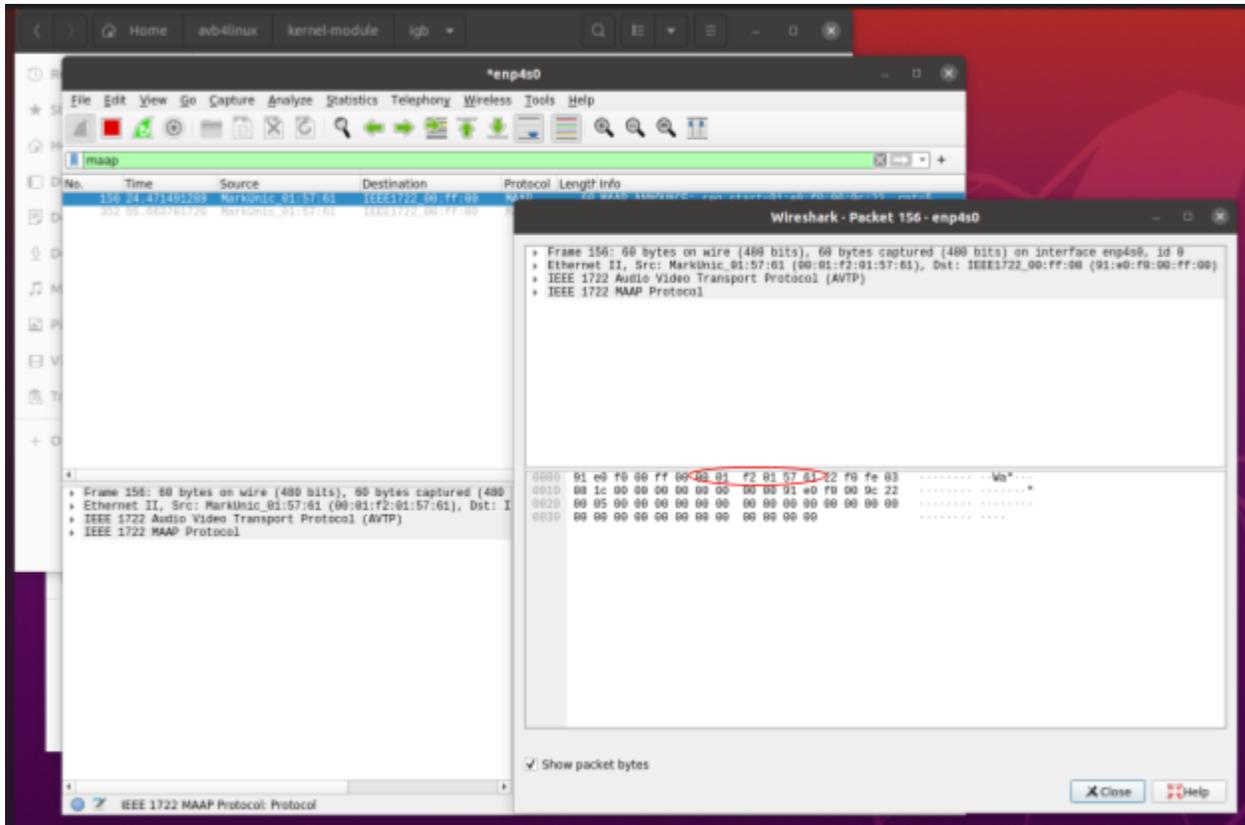


```
sonicart@sonicart-B660M-DS3H-AX-DDR4: ~/Desktop$ cd
sonicart@sonicart-B660M-DS3H-AX-DDR4:~$ ifconfig
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  ether 1c:fd:08:79:72:cb txqueuelen 1000 (Ethernet)
  RX packets 12299 bytes 3185195 (3.1 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12367 bytes 1851977 (1.8 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  device memory 0x42200000-422fffff

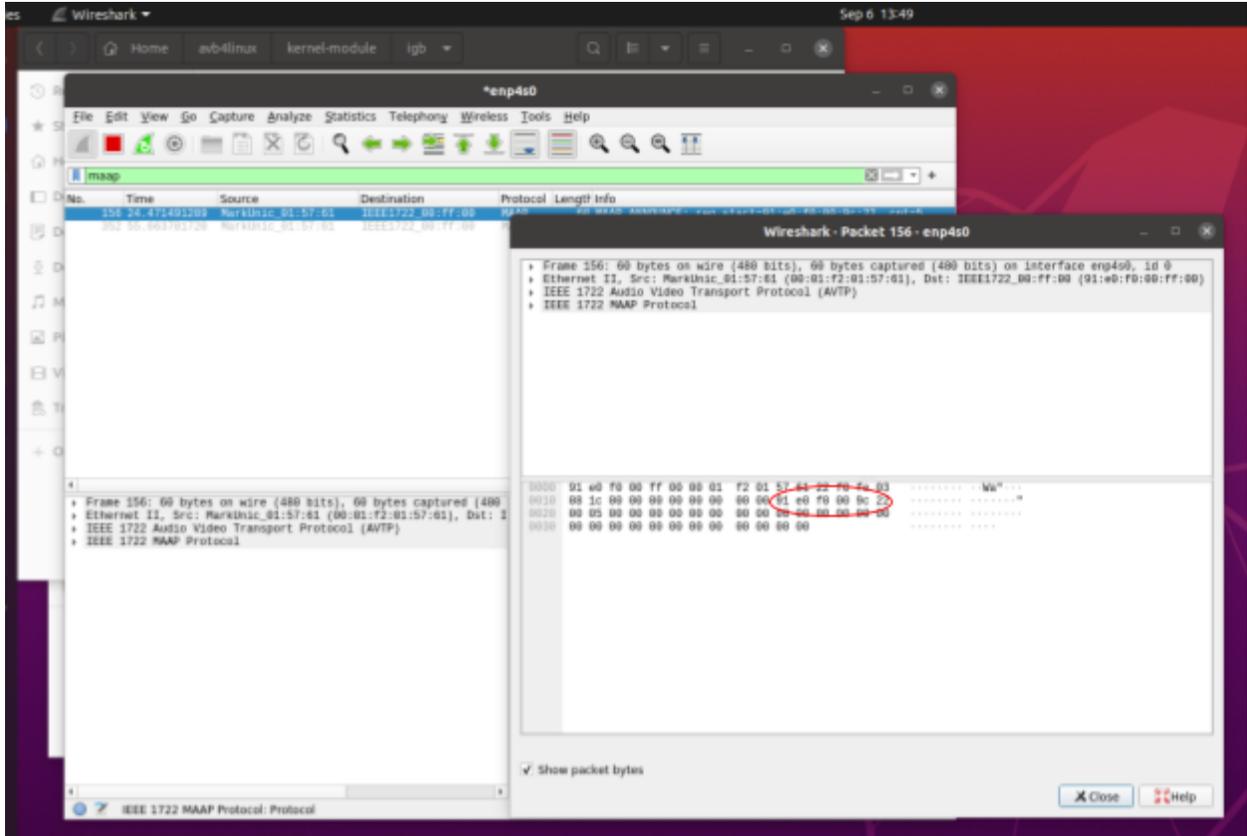
enp5s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
  ether 74:56:3c:23:44:82 txqueuelen 1000 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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 2726 bytes 246793 (246.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 2726 bytes 246793 (246.7 KB)
```

Next, both `AVB_DEVICE_SOURCE_MAC` and `AVB_DEVICE_TALKER_MAC_BASE` can be found through Wireshark. Run a packet capture on the i210 and filter for MAAP announce messages. Select one from the MOTU, and you should get a page showing more information about the packet. On this page, the MAC address that’s located where I’ve circled below is the MOTU’s; use your equivalent address for `AVB_DEVICE_SOURCE_MAC`.



On the other hand, to get AVB_DEVICE_TALKER_MAC_BASE, use the MAC address that's located where I've circled below.



On the other hand, to get AVB_DEVICE_TALKER_MAC_BASE, use the MAC address that's located where I've circled below.

That said, you can also check the talker information shown in the stream when you're running AVB4Linux. The MAC address you need will be listed as dst mac (see picture below).

```
Terminal
sonicart@sonicart-E

node JOIN
stream: 1cfd087972cb0000
listener: 6ccdd6c3ffe4
S30 L:D=2,S=1cfd087972cb0000 R=6ccdd6c3ffe4 VO/IN

V30 0002 R=6ccdd6c3ffe4 QA/IN

V30 0002 R=6ccdd6c3ffe4 QA/IN
unhandled from nrpd
S30 D:C=5,P=2,V=0002,N=2 R=6ccdd6c3ffe4 VO/IN

S30 D:C=5,P=2,V=0002,N=2 R=6ccdd6c3ffe4 VO/IN

S30 D:C=6,P=3,V=0002,N=3 R=6ccdd6c3ffe4 VP/IN

S30 D:C=6,P=3,V=0002,N=3 R=6ccdd6c3ffe4 VP/IN

S30 T:S=0001f20157610000,A=91e0f0009c22,V=0002,Z=224,I=1,P=112,L=137942 R=6ccdd6c3ffe4 VO/IN

talker:
node JOIN
stream: 0001f20157610000
st_mac: 91e0f0009c22
vtd: 2
S30 T:S=0001f20157610000,A=91e0f0009c22,V=0002,Z=224,I=1,P=112,L=137942 R=6ccdd6c3ffe4 VO/IN

S30 L:D=2,S=1cfd087972cb0000 R=6ccdd6c3ffe4 VO/IN

EVENT on STREAM ID=1cfd087972cb0000 with state ready
got a new/join indication
listener:
node JOIN
stream: 1cfd087972cb0000
listener: 6ccdd6c3ffe4
S30 L:D=2,S=1cfd087972cb0000 R=6ccdd6c3ffe4 VO/IN

V30 0002 R=6ccdd6c3ffe4 QA/IN

V30 0002 R=6ccdd6c3ffe4 QA/IN
unhandled from nrpd
S30 D:C=5,P=2,V=0002,N=2 R=6ccdd6c3ffe4 VO/IN

S30 D:C=5,P=2,V=0002,N=2 R=6ccdd6c3ffe4 VO/IN

S30 D:C=6,P=3,V=0002,N=3 R=6ccdd6c3ffe4 VP/IN

S30 D:C=6,P=3,V=0002,N=3 R=6ccdd6c3ffe4 VP/IN

S30 T:S=0001f20157610000,A=91e0f0009c22,V=0002,Z=224,I=1,P=112,L=137942 R=6ccdd6c3ffe4 VO/IN
```

PTP isn't Listed as a Module after running Modprobe

As unhelpful as this is going to sound, you'll probably be fine. We had the same issue, and while we never figured out why PTP wasn't listed when we ran *lsmod* after running *modprobe ptp*, AVB4Linux still worked. You can try installing Linux PTP (instructions given above) if that'll make you feel better, but as we already mentioned, we're not sure that actually did anything.

No Talker Information in the Stream

If you aren't seeing any information about a talker in the stream, make sure your MOTU routing is correct; in particle, make sure you configure the MOTU to have output streams. It's unlikely to be an issue with your MAC addresses, as we found that even when our MAC address for the talker was wrong, we still saw information about the talker being printed.

Computer can't Connect to Wifi

I know this sounds odd, but a day or two after installing the Intel i210, our computer could no longer connect to any wifi network. Turns out, this is a Windows problem. Simply turn off Fast Startup in the Windows settings.

The Kernel Updates Itself

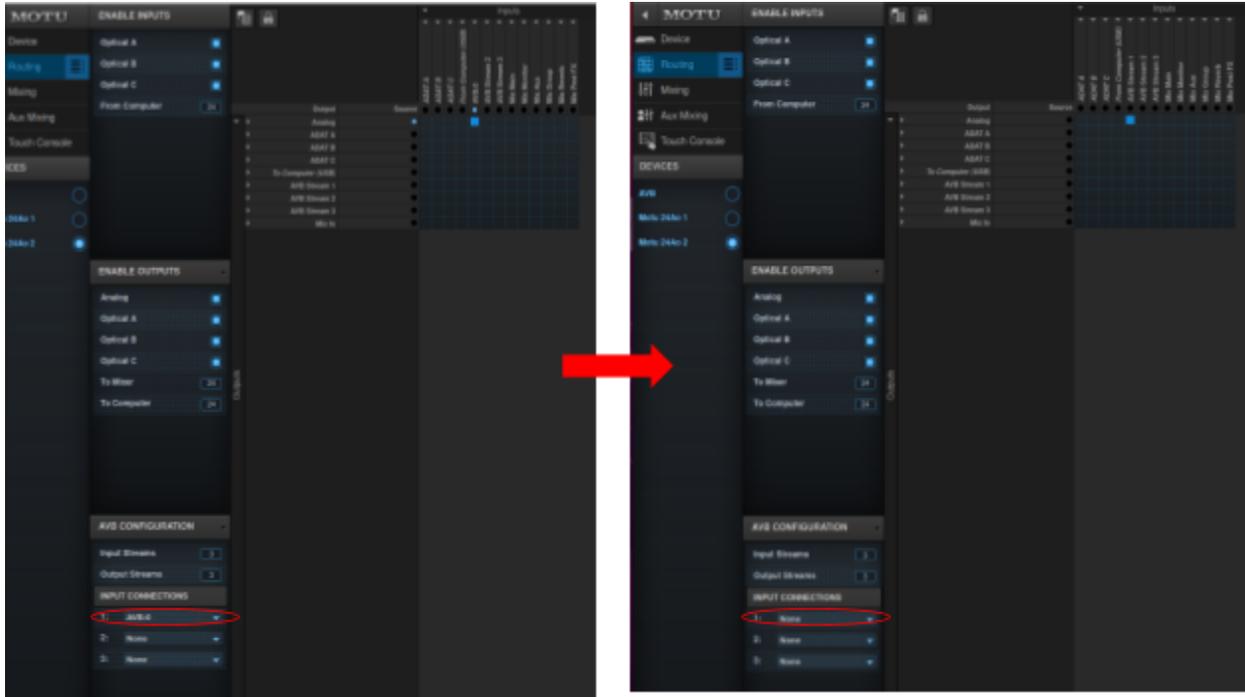
For reasons unbeknownst to us, the kernel sometimes updates itself after we power off the computer. When this has happened, we've found that AVB4Linux no longer runs correctly and we get all sorts of errors. To resolve the issue, you can try running *make clean* within the AVB4Linux directory, then *make* and *sudo make install*. That said, we simply configured our computer to boot into the GRUB menu, where we could choose which kernel version we wanted to run. To do the same, you can follow the instructions given by the top answer on [this page](#).

General Debugging Advice

Sending Audio via AVB to another Room

Once we had AVB working in Performative Computing, we began trying to send audio from Performative Computing to Spatlab via AVB. The setup in Performative Computing remained unchanged. In Spatlab, we had another MOTU. Naturally, you'll need the MAC address of this MOTU and the corresponding `AVB_DEVICE_TALKER_MAC_BASE`; you can get these the same way you got the first ones (make sure the new MOTU is connected to the rest of the network beforehand). To send audio to either MOTU, simply change `AVB_DEVICE_SOURCE_MAC` and `AVB_DEVICE_TALKER_MAC_BASE` to be the addresses that correspond to your chosen MOTU. Simple, right?

If you have everything set up correctly, when you run `avb_up.sh`, you should see a device listed as "AVB" and flickering blue bars under some of your analog outputs. Interestingly enough, for us, regardless of which MOTU's `AVB_DEVICE_SOURCE_MAC` and `AVB_DEVICE_TALKER_MAC_BASE` MAC addresses we used, both MOTUs would receive audio. As such, if you only want one MOTU to receive audio, in the MOTU web interface, simply set the input connections of the MOTU you don't want receiving any audio to "None." If I'm not making any sense, just look at the picture below.



Our next step was to send separate audio streams to both MOTUs. However, based on the issues opened for AVB4Linux, it seems that AVB4Linux cannot do this.

AVB on Beaglebone

Let me start by saying that if you thought getting AVB to work on Linux was a pain in the butt, AVB on Beaglebone is going to make that look like a cakewalk. Unsurprisingly, we didn't get very far on this. It seems that only one person has ever made any headway in actually getting AVB working on Beaglebone (her work can be found [here](#)). While we did download her code onto the Beaglebone, we never got around to testing it, in part because it seems to require two Beaglebones to work. We leave that to you! An AVB alternative we were considering (and that you can also try out) can be found [here](#).

While working on AVB for Beaglebone, you'll likely encounter a slew of other issues. In the interest of saving you some time, below are the major issues we ran into and how we solved them:

- No wifi on Beaglebone: we followed this [guide](#)
- Computer doesn't see Beaglebone even though it's connected: reflash your micro USB
- Problems with downloading Duraipandian's code: we used an image from this [page](#)