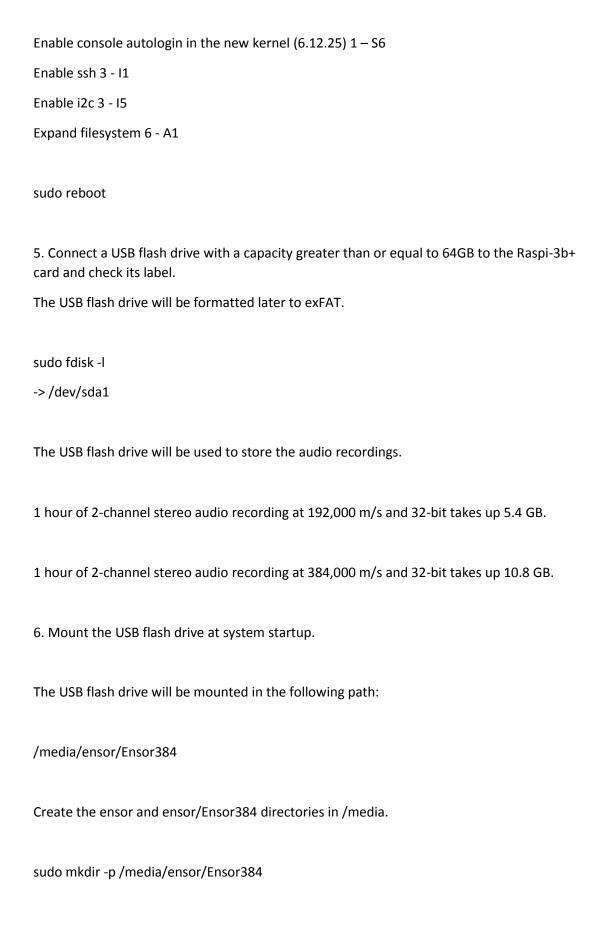
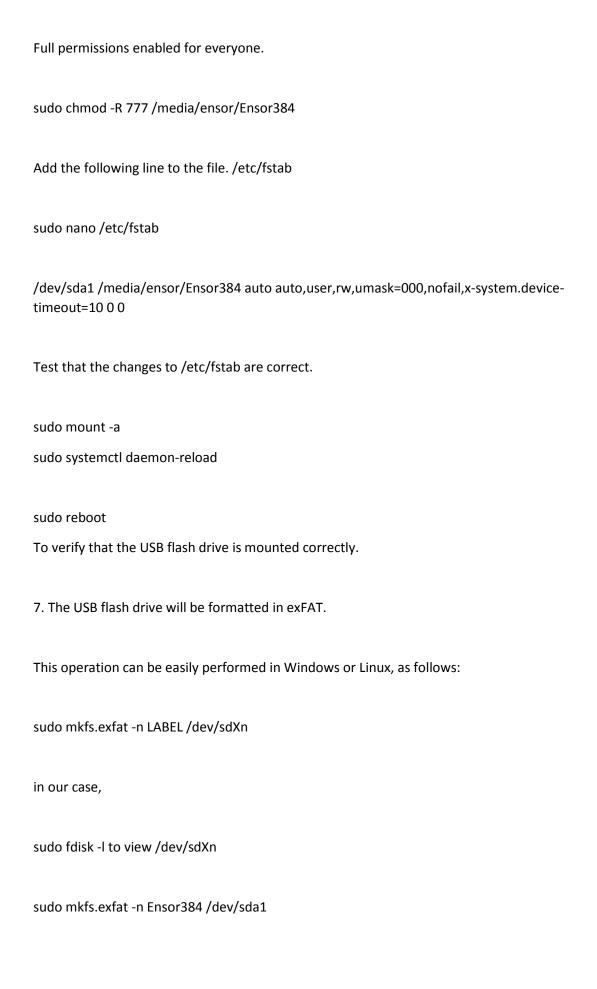
## Ensor-384 + Raspberry Pi 5 (Kernel 6.12.25)

Sequence for preparing a Raspberry Pi 5 card to work with the "Ensor-384" audio recorder card
1. Items needed to get started
- Raspberry Pi 5
- Original Raspberry Pi power supply
- Original 8GB micro SD card + USB micro SD recorder
- 64GB or larger USB flash drive
- A computer with an internet connection running Windows or Linux
- The ensor384_RPI5_ KERNEL_ 6.12.25.zip file downloaded from GitHub
2. Install the Raspberry Pi OS Lite (64) operating system without a desktop for the Raspberry Pi 5
Use Raspberry Pi Imager by configuring the username and password,
enabling the ssh server
user: ensor
password: ****** any
password: ****** any
password: ****** any  It is recommended to use the Putty program to send commands via SSH to the Raspberry Pi
It is recommended to use the Putty program to send commands via SSH to the Raspberry Pi  3. Connect the card Ensor-384 to the Raspberry Pi, and then connect it to the internet via an
It is recommended to use the Putty program to send commands via SSH to the Raspberry Pi  3. Connect the card Ensor-384 to the Raspberry Pi, and then connect it to the internet via an Ethernet cable. Turn on the system.
It is recommended to use the Putty program to send commands via SSH to the Raspberry Pi  3. Connect the card Ensor-384 to the Raspberry Pi, and then connect it to the internet via an Ethernet cable. Turn on the system.  3. Update the operating system.
It is recommended to use the Putty program to send commands via SSH to the Raspberry Pi  3. Connect the card Ensor-384 to the Raspberry Pi, and then connect it to the internet via an Ethernet cable. Turn on the system.  3. Update the operating system.  sudo apt update





Download the compressed file ensor384\_RPI5\_KERNEL\_6.12.25.zip from the internet, which includes the Ensor directory and the audio configuration file ensor.conf. Unzip ensor384\_RPI5\_KERNEL\_6.12.25.zip and copy the ensor.conf file and the Ensor directory to the USB flash drive. Connect the USB flash drive to the Raspberry Pi and turn on the system. The USB flash drive contains the recording configuration file: ensor.conf and the Ensor directory, which will be copied to: /home/ensor/ cp -dr /media/ensor/Ensor384/Ensor /home/ensor/ Once the Ensor directory has been copied, you can delete it from the USB flash drive if desired. rm -dr/media/ensor/Ensor384/Ensor To record files larger than 4GB. Install the exfat-fuse program on the Raspberry Pi sudo apt install exfat-fuse 8. Copy the compiled dts file, (tlv320adcx140-overlay.dtbo) for the "Ensor-384" audio recording card to /boot/firmware/overlays The tlv320adcx140-overlay.dts file is located in the /home/ensor/Ensor/dts/tlv320adcx140

From a Linux computer or Windows:

directory.

/home/ensor/Ensor/dts/tlv320adcx140/tlv320adcx140-overlay.dts We will need to compile it with the following command: sudo dtc -@ -H epapr -O dtb -o tlv320adcx140-overlay.dtbo -Wno-unit\_address\_vs\_reg tlv320adcx140-overlay.dts and copy the compiled file tlv320adcx140-overlay.dtbo to /boot/firmware/overlays/ sudo cp /home/ensor/Ensor/dts/tlv320adcx140/tlv320adcx140-overlay.dtbo /boot/firmware/overlays/ 9.- Modify the /boot/firmware/config.txt file sudo nano /boot/firmware/config.txt dtparam=i2c=on dtparam=i2s=on gpio=4=op,dh dtoverlay=tlv320adcx140-overlay #Disable the default audio cards #dtparam=audio=on #The following line cannot be disabled because the display is not working on reboot dtoverlay=vc4-kms-v3d #disable Wi-Fi and Bluetooth to avoid noise and power consumption dtoverlay=disable-wifi

dtoverlay=disable-bt 10.- Reboot the system and verify that the ADCX140 recording card has been installed. arecord -I Verify that card is 0 and device is 0. card 0: ADCX140 [ADCX140], device 0: These are the necessary parameters for the recording program. /home/ensor/Ensor/eea.py 11.- Check that the card records audio at a sampling rate of 192,000 m/s and 32 bits. The recording program is located in /home/ensor/Ensor/eea.py and the recording configuration file, ensor.conf, is located on the USB flash drive. If the USB flash drive does not have the ensor.conf configuration file saved, the eea.py program creates a default one with the data shown below and the program will run with that data. By default, the system does not record until "record=yes" is specified. Recording parameters can be modified with any text editor. nano/media/ensor/Ensor384/ensor.conf [DEFAULT]

#date of the recording

#file name

record = no #enable recording or not <yes, no>

#by default, does not record

file\_name = test\_192kHz\_32bits

date = 30/05/2025

```
[sound.format]
```

sampling = 192000 #sample rate in Hz

bits = 32 #number of bits per sample, fixed at 32 bits

[record.time]

total\_time = 10 #total recording time in seconds

file\_time = 10 #since the files are very large, it is advisable to split them into multiple

#files

delay = 0 #wait time, in seconds, from the start, before starting recording

index = 0 # index of possible recordings

# that are made on the same day, each recording increments

# the index by one

[channel.selection] # selection of channels for recording

channel\_1 = yes # only 2 channels can be selected

channel\_2 = no # the microphones on the board correspond to channels 1 and 4

channel\_3 = no # channels 2 and 3 can be configured as line-in or microphone

channel\_4 = yes

[gain.channels] # channel gain in dB

# the range of allowed values is from 0 dB to 42 dB

range = from 0dB to 42dB

 $channel_1 = 42$ 

 $channel_2 = 0$ 

 $channel_3 = 0$ 

 $channel_4 = 42$ 

[aux\_channels.type]

possible\_options = microphone, line\_in

aux\_ch2\_type = microphone

aux\_ch3\_type = microphone

```
[aux_channels.input_class]

possible_options = differential, single-ended

aux_ch2_input_class = differential

aux_ch3_input_class = differential

[aux_channels.input_impedance]

possible_options = 2K5, 10K, 20K

aux_ch2_zin = 2K5
```

A directory structure will be created based on the current date, and file names will depend on the current hour and minute, ending with the file number.

Modify the configuration file to record

nano/media/ensor/Ensor384/ensor.conf

[DEFAULT]

aux\_ch3\_zin = 2K5

record = yes #enable recording or not <yes, no>

... #by default, it doesn't save

And try saving

sudo /home/ensor/Ensor/eea.py

You need to use sudo to change the system date.

12.- Automatically record at system startup

At system startup, the eea.py program has a 20-second delay to allow time for the USB flash drive to mount correctly and be used.

In the new kernel, the rc-local service service must be added.

We will use the rc-local.service service.

sudo nano /etc/systemd/system/rc-local.service

[Unit]

Description=rc-local

After=network.target

[Service]

User=root

ExecStart=/home/ensor/Ensor/recorder.sh

[Install]

WantedBy=multi-user.target

Enable the service to start at system startup.

sudo systemctl enable rc-local.service

13.- Script /home/ensor/Ensor/recorder.sh

nano /home/ensor/Ensor/recorder.sh

#!/bin/sh

sudo /home/ensor/Ensor/eea.py

sudo shutdown -h now

This script starts recording according to the ensor.conf settings, and when it's finished, it shuts down the system to minimize power consumption.

## 14. Back up the card

widows: Win32DiskImager

Linux: dd if=/dev/sda of=/path/to/image

From Raspi: sudo dd if=/dev/mmcblk0 of=/media/ensor/Ensor384/raspbian.img bs=1M

status=progress

15. Prepare Raspberry Pi to record at 384,000 samples/s and 32 bits

Kernel 6.12.25 now supports two stereo channels at 384,000 samples per second and 32 bits.

Therefore, edit the ensor.conf configuration file and set it to the following:

[sound.format]

sampling = 384000

bits = 32