Getting Started with the SAMA7G54-EK Audio System under Linux™

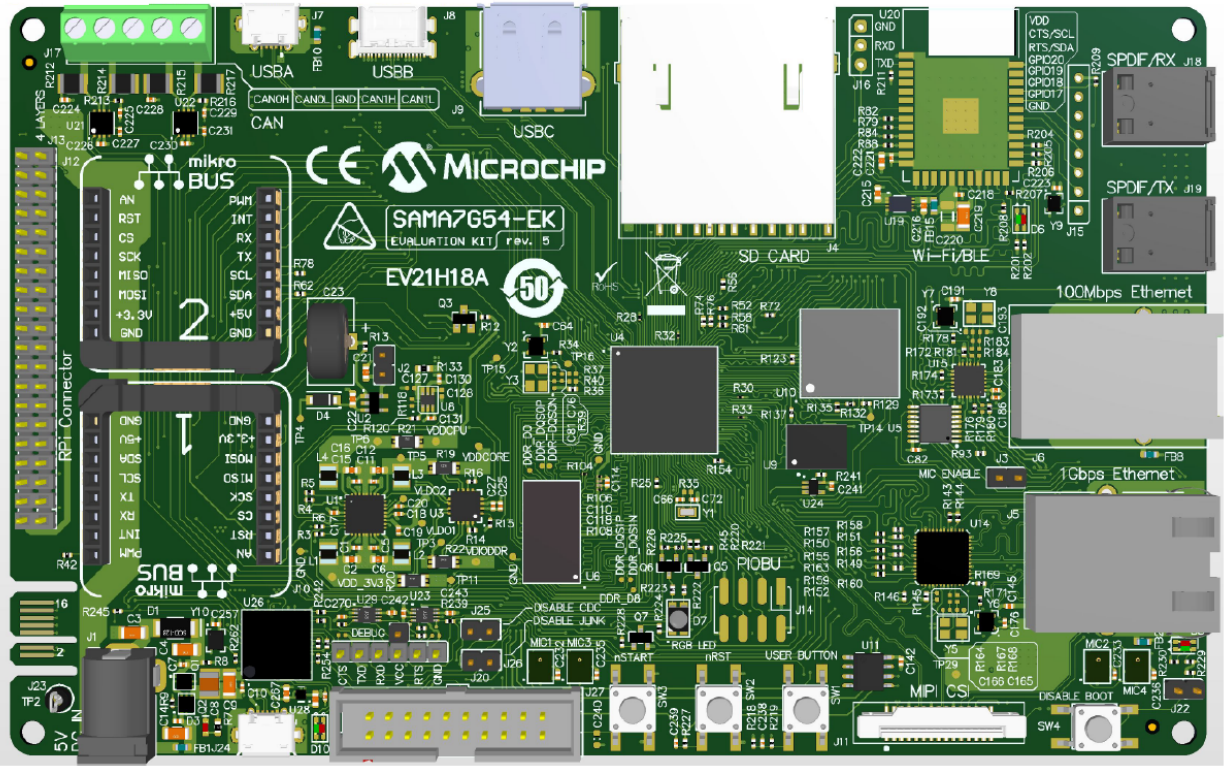
TB3372

www.microchip.com
Product Page Links

Introduction

This document explains how to use the Quad PDM microphones and the SPDIF transmitter and receiver in the SAMA7G54-EK evaluation kit with Linux.

Figure 1. SAMA7G54-EK Board

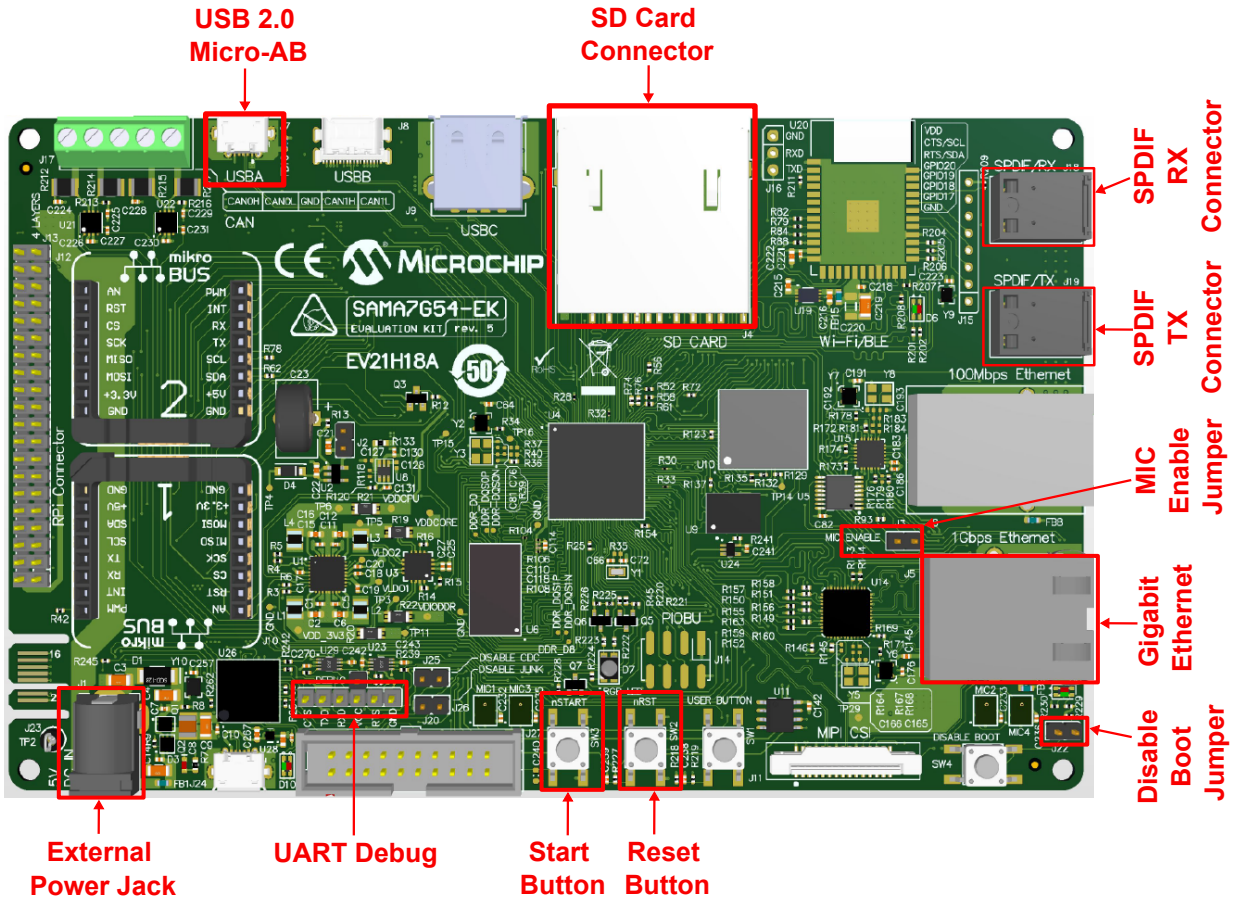


This guide helps the user to configure, build, and test audio interfaces like the Quad PDM microphones and SPDIF transmitter/receiver available on the SAMA7G54-EK evaluation kit in a Linux buildroot environment. It includes instructions for configuring buildroot to support audio functionalities, capturing sound through the built-in microphone, and using the SPDIF interface for both transmitting and receiving audio data.

1. Prerequisites

- Hardware:
  - SAMA7G54-EK evaluation kit
  - USB audio adapter external sound card with SPDIF digital audio
  - TOSLINK plug fiber optic cable
  - Headphones with microphone
  - 3.5 mm headphone splitter for computer, 2 male to 1 female 3.5 mm headphone mic audio Y splitter cable
  - FTDI cable
  - SD card
- Software:
  - Buildroot for audio functionality
    - Buildroot version: linux4microchip-2023.10
    - Buildroot external Microchip (buildroot-external-microchip-) version: linux4microchip-2023.10
  - Xear audio center
  - WinSCP

2. Hardware Setup

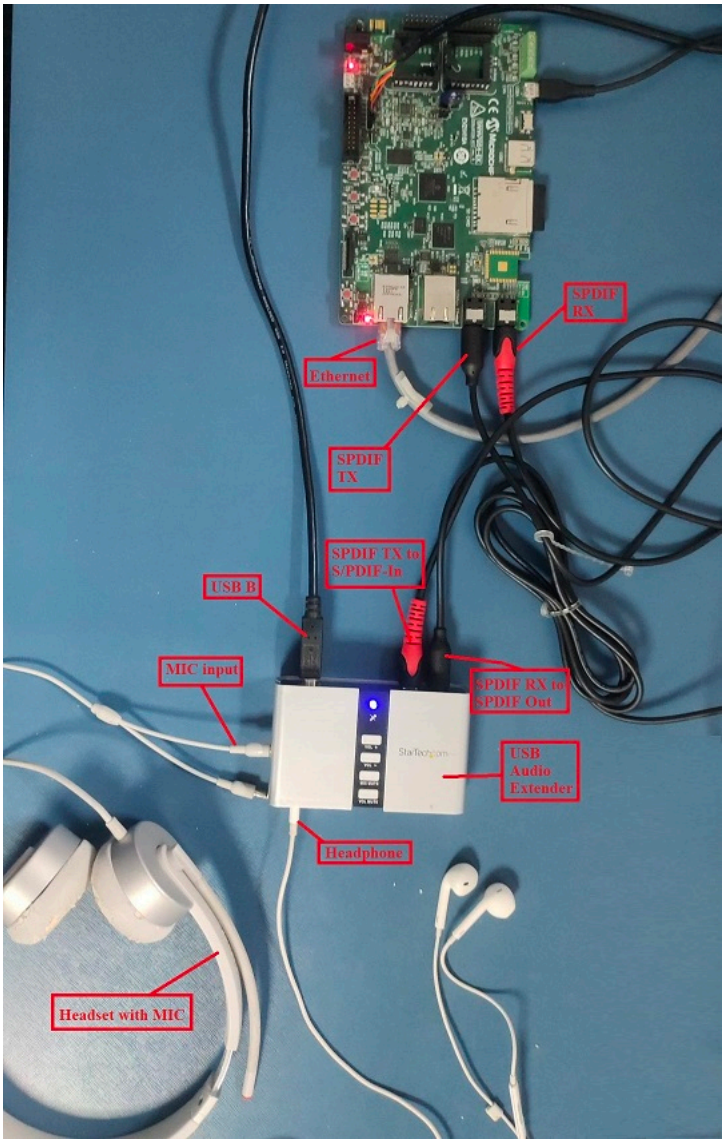


Follow these instructions:

1. Close the J3 jumper to enable the 4-microphone PDMIC interface.
2. Connect the Toslink cable between the SPDIF TX connector (J19) of the SAMA7G54-EK and the *S/PDIF IN* of the USB audio extender.

3. Connect the Toslink cable between the SPDIF RX connector (J18) of the SAMA7G54-EK and the *S/PDIF OUT* of the USB audio extender.
4. Connect the USB-TTL cable to J20 for console serial communication. The baud rate is 115200.

5. Power up the USB audio extender by connecting the USB B cable to the USB B port of the USB audio extender.
6. Power up the board by connecting the USB 2.0 micro-AB cable to the USB port J7 on the SAMA7G54-EK development board.
7. Connect the headphones to the headphone jack located on the front of the USB audio extender.
8. Use the audio splitter cable to split the audio and mic. Connect the audio splitter mic cable to the mono *MIC 1 In* port of the USB audio extender.



3. Custom Linux Build Procedure For Audio System

If the user intends to use the pre-build img, skip these steps.

3.1 Get Buildroot for MCHP

To get the source code, the buildroot-mchp and buildroot-external-microchip repositories should be cloned. The buildroot-mchp repository is a fork of Buildroot with a few specific patches. The external tree provides additional defconfigs and packages dedicated to our demos.

To clone buildroot-mchp from the repository:

```
$ git clone https://github.com/linux4microchip/buildroot-mchp.git
```

To clone buildroot-external-microchip from the repository:

```
$ git clone https://github.com/linux4microchip/buildroot-external-microchip.git
```

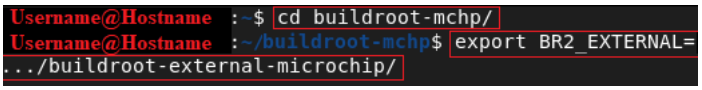
The source code should be pointing to the latest version of the buildroot-mchp and buildroot-external-microchip repositories.

Tip: We recommend using the same linux4microchip tag for both repositories.

Note: This document is verified with the tag “linux4microchip-2024.04”.

3.2 Configure buildroot-mchp

1. Navigate to the buildroot-mchp directory:
   ```
   $ cd buildroot-mchp
   ```
2. Enter the following command to export the additional defconfigs and packages from the external tree (buildroot-external-microchip):
   ```
   $ export BR2_EXTERNAL=../buildroot-external-microchip/
   ```



3. To find the SAMA7G54-EK defconfig files, go to the *configs* directory located inside the *buildroot-external-microchip* directory:

```
user@ hostname :~/buildroot-external-microchip/configs$ ls
at91sam9x5ek_headless_defconfig            sama5d29_curiosity_headless_defconfig
icicle_amp_defconfig                       sama5d2_icp_headless_defconfig
icicle_defconfig                           sama5d2_ptc_ek_graphics_defconfig
icicle_nand_defconfig                      sama5d2_ptc_ek_headless_defconfig
icicle_nor_defconfig                       sama5d2_ptc_ek_nodered_defconfig
icicle_rootfs_defconfig                    sama5d2_xplained_graphics_defconfig
sam9x60_curiosity_graphics_defconfig       sama5d2_xplained_headless_defconfig
sam9x60_curiosity_headless_defconfig       sama5d2_xplained_nodered_defconfig
sam9x60ek_graphics_defconfig               sama5d2_xplained_optee_graphics_defconfig
sam9x60ek_headless_defconfig               sama5d2_xplained_optee_headless_defconfig
sam9x75_curiosity_graphics_defconfig       sama5d3_eds_headless_defconfig
sam9x75_curiosity_headless_defconfig       sama5d3_eds_nf_defconfig
sama5d27_som1_ek_graphics_defconfig        sama5d3_xplained_graphics_defconfig
sama5d27_som1_ek_headless_defconfig        sama5d3_xplained_headless_defconfig
sama5d27_som1_ek_headless_wilc_defconfig   sama5d3_xplained_nodered_defconfig
sama5d27_som1_ek_nodered_defconfig         sama5d4_xplained_graphics_defconfig
sama5d27_som1_ek_optee_graphics_defconfig  sama5d4_xplained_headless_defconfig
sama5d27_som1_ek_optee_headless_defconfig  sama5d4_xplained_headless_wilc_defconfig
sama5d27_wlsom1_ek_graphics_defconfig      sama5d4_xplained_nodered_defconfig
sama5d27_wlsom1_ek_headless_defconfig      sama7g5ek_headless_defconfig
sama5d29_curiosity_graphics_defconfig
```

4. Make the configuration files using the following command from the *buildroot-mchp* directory:
   `$ make sama7g5ek_headless_defconfig`

```
Username@Hostname :~/buildroot-external-microchip$ cd ../buildroot-mchp/
Username@Hostname :~/buildroot-mchp$ make sama7g5ek_headless_defconfig
```

3.3 Configure Buildroot for the Audio System

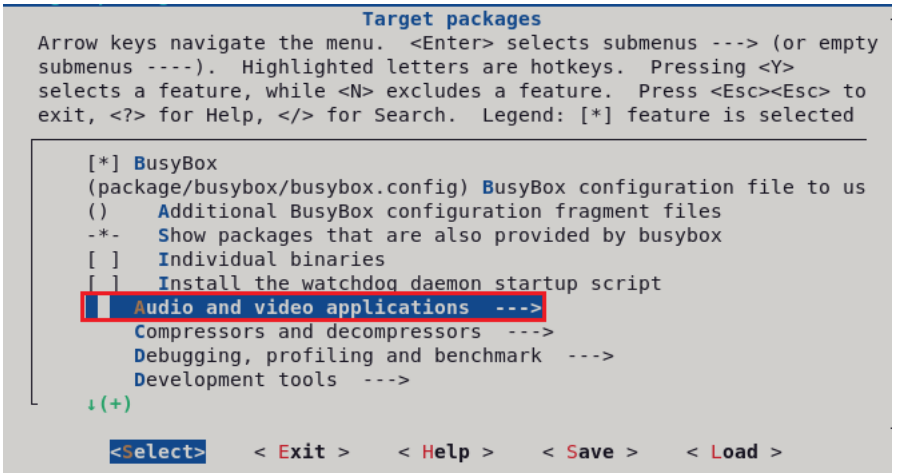
1. Enter the following command to modify the configurations:
   `$ make menuconfig`
   The *Buildroot Configuration* window opens.

```
Usrname@Hostname :~/buildroot-mchp$ make menuconfig
```

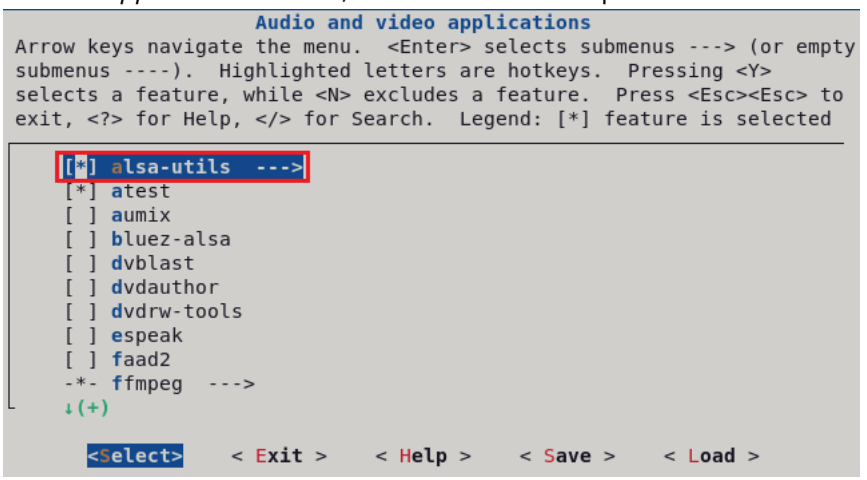
2. In the menu, click *Target packages.*

```
        Buildroot linux4microchip-2024.04 Configuration
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
selects a feature, while <N> excludes a feature.  Press <Esc><Esc> to
exit, <?> for Help, </> for Search.  Legend: [*] feature is selected
┌──────────────────────────────────────────────────────────────────────┐
│        Target options  --->                                          │
│        Toolchain  --->                                               │
│        Build options  --->                                           │
│        System configuration  --->                                    │
│        Kernel  --->                                                  │
│        Target packages  --->                                         │
│        Filesystem images  --->                                       │
│        Bootloaders  --->                                             │
│        Host utilities  --->                                          │
│        Legacy config options  --->                                   │
└    ↓(+)                                                              ┘
      <Select>    < Exit >    < Help >    < Save >    < Load >
```

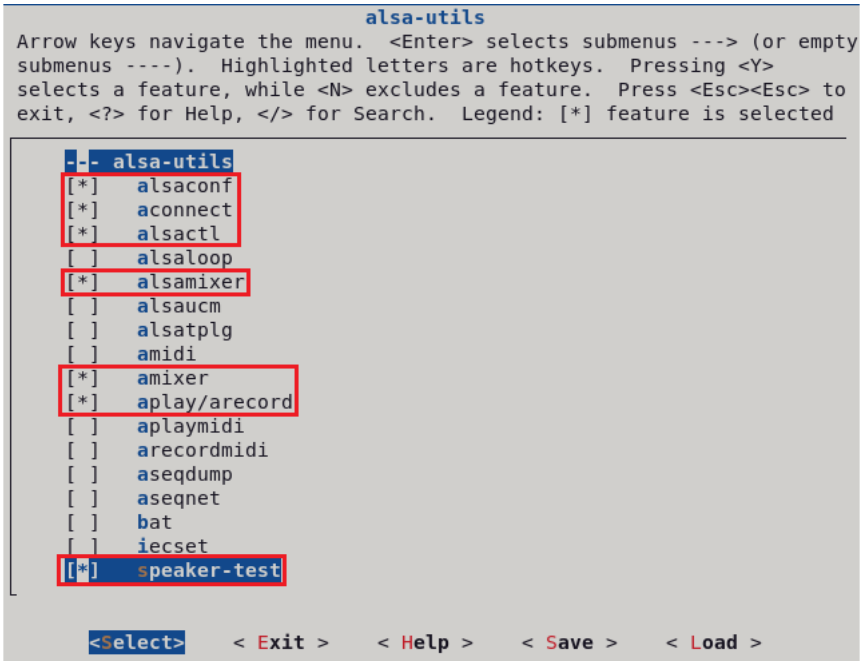
3. In the *Target packages* window, select *Audio and video applications* and press *Enter*.



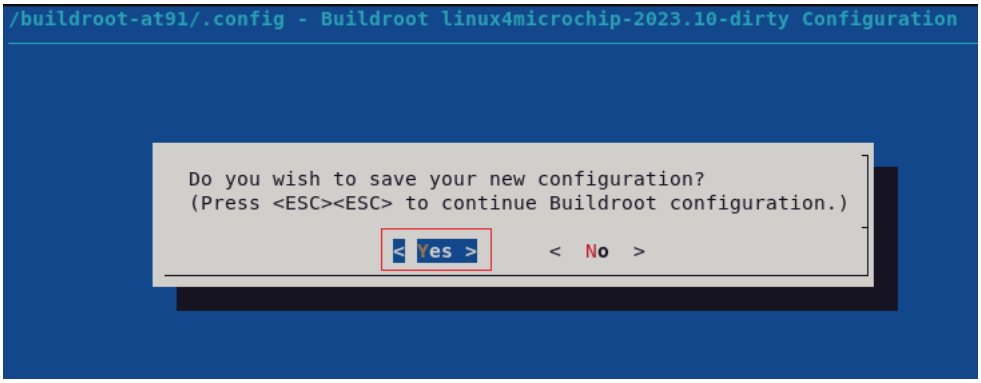
4. In the *Audio and video application* window, select *alsa-utils* and press *Enter*.



5. In the *alsa-utils* window, ensure the *alsaconf, aconnect, alsactl, alsamixer, amixer, aplay/arecord,* and *speaker-test* check boxes are selected. If not, select them and press *Enter*.



6. Move to `<Exit>` with the arrows and press the *Enter* key to exit from this screen, and click *Yes* to save the configuration.



3.4 Build the Linux Image

After all the required changes are made, the building can be performed.

1. Navigate to the *buildroot-at91* directory:
   ```
   $ cd buildroot-at91
   ```
2. Build the Linux image with the following command:
   Note: Before building the buildroot-at91, setup the cross-compile toolchain. Refer to the cross compiler section for additional instructions.

   ```
   $ make
   ```

   The building process will take a few hours to complete.
3. Once the build is complete, navigate to *buildroot-at91/output/Images*.
4. Verify that the *sdcard.img* file is generated.

4. Flash the Linux Image in the SD Card

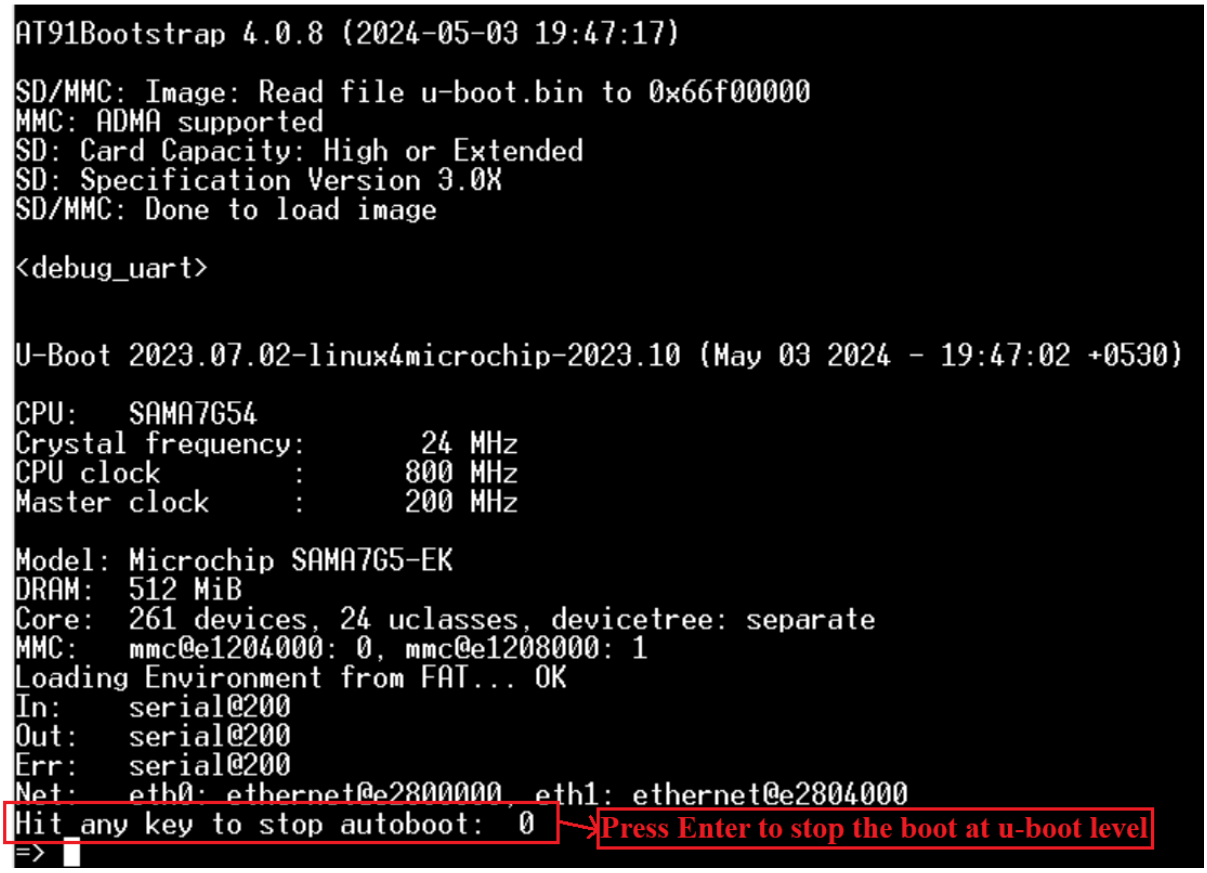
1. Follow the steps listed here to flash the generated sdcard.img in the SD card.
2. Insert the SD card into the board and press the *reset* button.
3. Open any console serial communication and observe Linux booting from SD card.

5. Apply the Device Tree Overlay to Enable PDM Microphones

Pulse Digital Modulation (PDM) for microphones is a method used to convert an analog signal into a digital signal. The SAMA7G54-EK features four PDM microphones. For more information, refer to the *Quad MEMS Microphones* section in the *SAMA7G54-EK User´s Guide (DS50003273)*, available on www.microchip.com.

Follow the steps below to use the device tree overlay to enable the PDM microphones:

1. During booting, press *ENTER* to stop at U-Boot level and apply the device tree overlay to enable the Quad PDM microphones.



2. a. Edit the *bootcmd* environment variable using the *edit bootcmd* command and apply the *PDMC0* overlay by updating the string bootcmd from

```
bootcmd=fatload mmc 1:1 0x63000000 sama7g5ek.itb; bootm
0x63000000#kernel_dtb
```

to

```
bootcmd=fatload mmc 1:1 0x63000000 sama7g5ek.itb; bootm
0x63000000#kernel_dtb#pdmc0
```

b. Use the *saveenv* command to save the bootcmd variables previously set in the environment permanent storage space.

3. Enter the *boot* command to boot the kernel and dtb.

```
=> boot
4943532 bytes read in 304 ms (15.5 MiB/s)
## Loading kernel from FIT Image at 63000000 ...
   Using 'kernel_dtb' configuration
   Trying 'kernel' kernel subimage
     Description:  Linux4SAM Linux kernel
     Type:         Kernel Image
     Compression:  uncompressed
     Data Start:   0x630000d4
     Data Size:    4881144 Bytes = 4.7 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x62000000
     Entry Point:  0x62000000
     Hash algo:    crc32
     Hash value:   4bbc49d3
     Hash algo:    sha1
     Hash value:   5a9aa6c02feaa35d17f5a17124eff6640ace8d05
   Verifying Hash Integrity ... crc32+ sha1+ OK
## Loading fdt from FIT Image at 63000000 ...
   Using 'kernel_dtb' configuration
   Trying 'base_fdt' fdt subimage
     Description:  SAMA7G5-EK Flattened Device Tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x634a7d04
     Data Size:    44378 Bytes = 43.3 KiB
     Architecture: ARM
     Load Address: 0x61000000
     Hash algo:    crc32
     Hash value:   f4dfd9ed
     Hash algo:    sha1
     Hash value:   ccd88978c716bcee62a659cb2a644be486b850b2
   Verifying Hash Integrity ... crc32+ sha1+ OK
```

4. To log in as the root user, enter *root*.

```
Welcome to the Microchip SAMA7G5 Demo
sama7 login: root
```

This completes enabling PDM microphones with the pdmc0 device tree overlay.

6. Testing Methods

6.1 Test 1: Testing the PDM Microphones using ALSA

This test is intended to record the user's audio input using the quad PDM microphones and to save it as a *.wav* file to the SD card using the ALSA (Advanced Linux Sound Architecture) utility.

Follow these instructions:

1. To check the PDMC audio device card number and device number, use the command below:
   ```
   $ arecord -l | grep pdmc
   ```
   In the following figure, the PDMC card number is 0 and the device number is 4.

   ```
   # arecord -l | grep pdmc
   card 0: EK [mchp-asrc-card @ sama7g5 EK], device 4: mchp-pdmc - dmic-hifi dmic-hifi-4 [mchp-pdmc - dmic-hifi dmic-hifi-4]
   #
   ```

   This command verifies that the pdmc0 device tree overlay has been applied successfully.
2. To capture the sound from the PDM microphones, use the command below:
   ```
   $ arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE pdmc.wav
   ```

   ```
   # arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE pdmc.wav
   Recording WAVE 'pdmc.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
   Aborted by signal Interrupt...
   ```

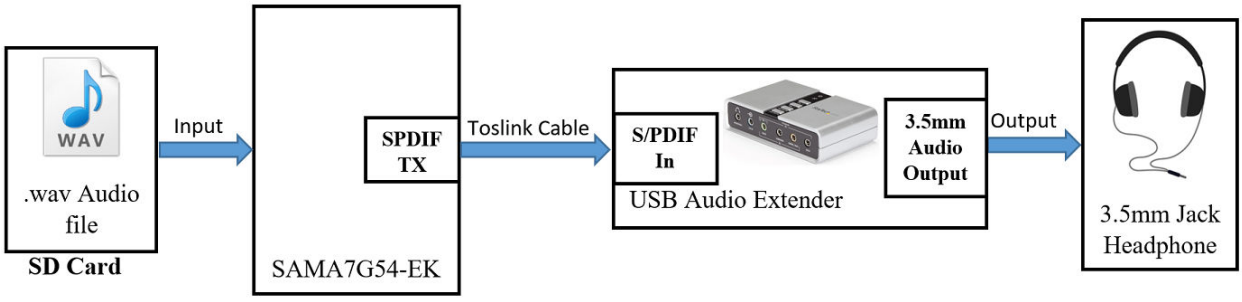
   To stop recording, press *Ctrl* + *C*.

   The .wav file is saved to the SD card in the */root* directory.
3. To play the recorded audio, either use the SPDIF Transmitter on the SAMA7G54-EK or transfer the audio file from the SAMA7G54-EK to the host PC using the Secure Copy Protocol (SCP) and listen to it on the host PC.

6.2 Test 2: Testing the SPDIF Transmitter using ALSA and a USB Audio Extender

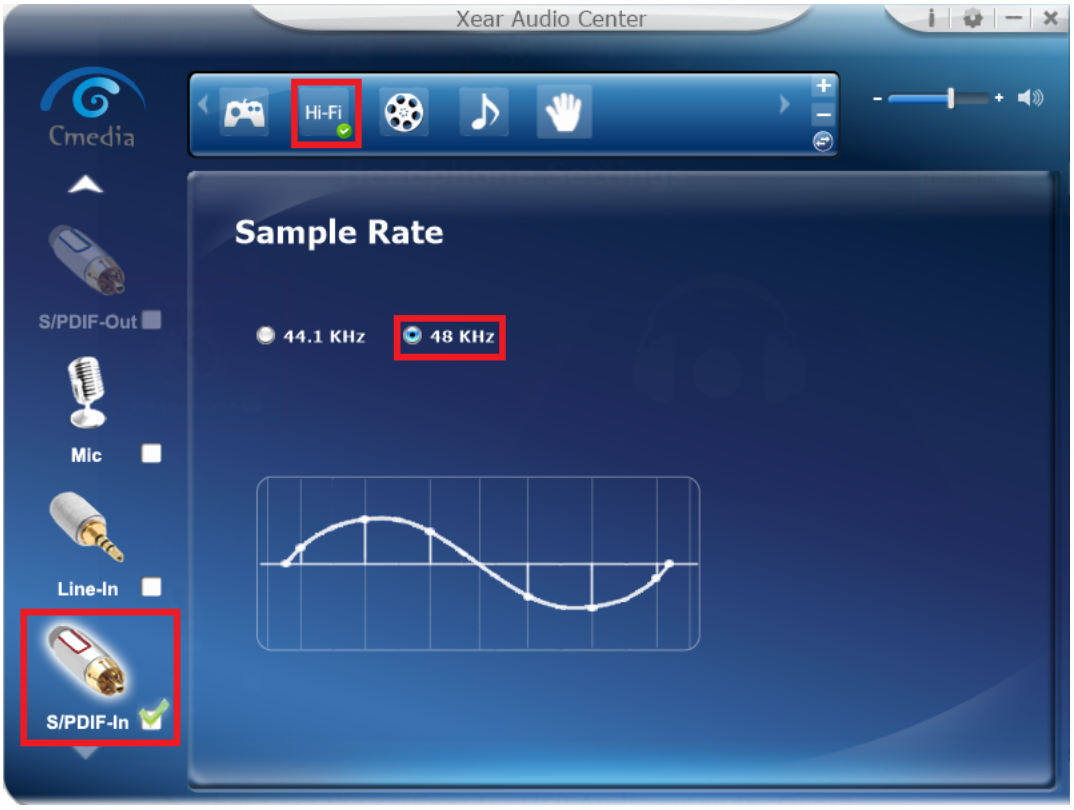
The SPDIF transmitter converts the audio signal from the source (.wav file) internal format (which can be PCM or compressed audio) into a format suitable for SPDIF transmission.

In this test, the audio file (.wav file) is transmitted through the SPDIF TX peripheral on the board. It is received as an SPDIF input on the USB audio extender and can be heard through headphones connected to the USB audio extender.



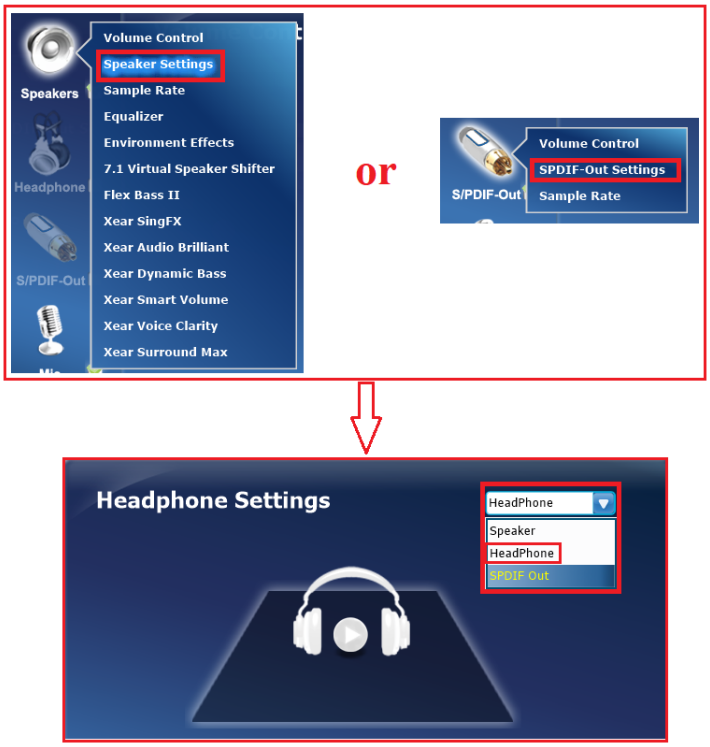
6.2.1 Host PC Setup

1. Open the *Xear audio center* application and select the input and output options to configure the USB audio extender.
2. Select the S/PDIF-In as the input for the USB audio extender in the application, set the sample rate to 48 kHz and select the HiFi mode as shown in the following figure.

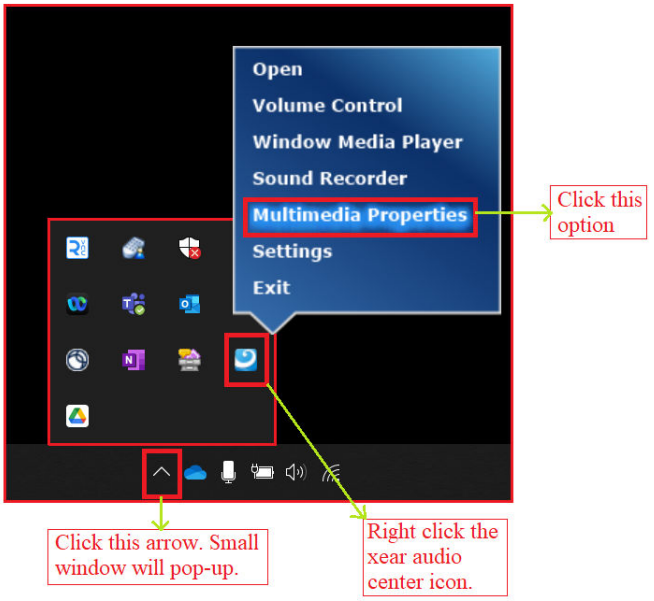


3. Right click on *Headphone* and select *Headphone Setting* to configure the *headphone* as the output for the USB audio extender.

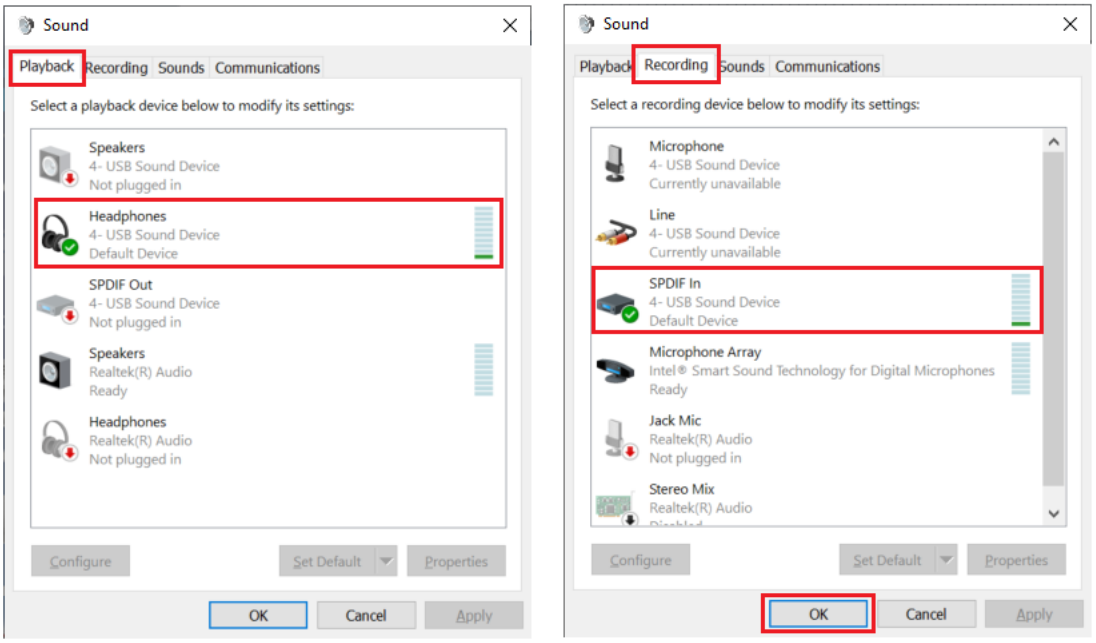
4. If the headphone is enabled by default, right-click on the *S/PDIF-Out* or *Speakers* icon, then navigate to *Settings* and select *Headphone*.



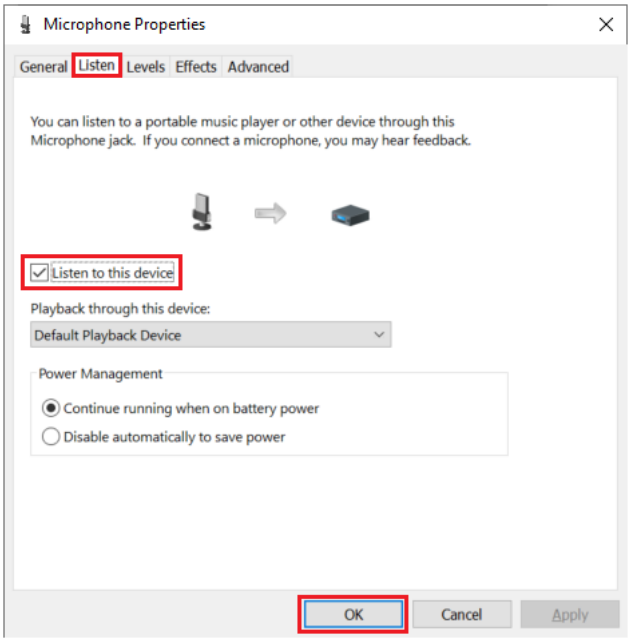
5. Move to the lower-right corner of the Windows desktop and click on the upward-pointing arrow to reveal the hidden software icons. Then, right-click on the *Xear audio center* icon and select *Multimedia Properties*.
   Note: If the *Xear audio center* icon is not present, refer to the *CM6206 Software_User_Manual_Xear_Audio_Center*, available in the download folder.



6. In the Media Properties page on the Playback tab, ensure that *Headphones* is selected. In the Recording tab, verify that *SPDIF In* is chosen. Then, double click *SPDIF In* to open the *SPDIF In Properties* dialog box.



7. In the *SPDIF In Properties* window, go to the *Listen* tab. Then, enable the *Listen to this device* feature by checking the corresponding checkbox and click *OK*.



6.2.2 ALSA Testing Procedure

1. Ensure the *TOSLINK optical cable* is connected at one end to *SAMA7G54-EK SPDIF TX* and at the other end to *SPDIF-In* on the USB audio extender, as explained in Hardware Setup.
2. To check the *SPDIF TX* audio device card number and device number, use the following command:

   ```
   $ aplay -l | grep spdiftx
   ```

   In the capture below, the card number is 0 and the device is 6.

```
# aplay -l | grep spdiftx
card 0: EK [mchp-asrc-card @ sama7g5 EK], device 6: mchp-spdiftx - dit-hifi dit-hifi-6 [mchp-spdiftx - dit-hifi dit-hifi-6]
```

3. To test the *SPDIF TX* using the *speaker-test* command:
   ```
   $ speaker-test -Dplughw:0,6 -c 2 -t wave
   ```
   Listen to the audio data, making sure it can be heard on the *Front left* and *Front Right* audio channels.

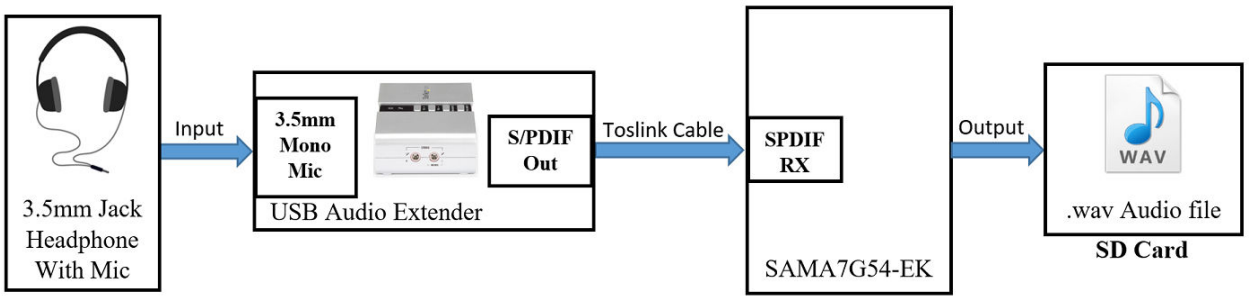
```
# speaker-test -Dplughw:0,6 -c 2 -t wave

speaker-test 1.2.8

Playback device is plughw:0,6
Stream parameters are 48000Hz, S16_LE, 2 channels
WAV file(s)
Rate set to 48000Hz (requested 48000Hz)
Buffer size range from 4 to 131072
Period size range from 2 to 16384
Using max buffer size 131072
Periods = 4
was set period_size = 16384
was set buffer_size = 131072
 0 - Front Left
 1 - Front Right
Time per period = 0.594185
 0 - Front Left
 1 - Front Right
Time per period = 3.077637
 0 - Front Left
 1 - Front Right
Time per period = 3.071827
 0 - Front Left
 1 - Front Right
Time per period = 3.062805
 0 - Front Left
 1 - Front Right
Time per period = 2.756612
 0 - Front Left
 1 - Front Right
```

4. Play the *.wav* file transmitted through the *SPDIF TX* interface using the following command:
   ```
   $ aplay -Dplughw:0,6 -r 48000 -c 2 -f S16_LE pdmc.wav
   ```
   In this capture, *pdmc.wav* is an input audio file.

```
# aplay -Dplughw:0,6 -r 48000 -c 2 -f S16_LE pdmc.wav
Playing WAVE 'pdmc.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

6.3 Test 3: Testing the SPDIF Receiver using ALSA and a USB Audio Extender

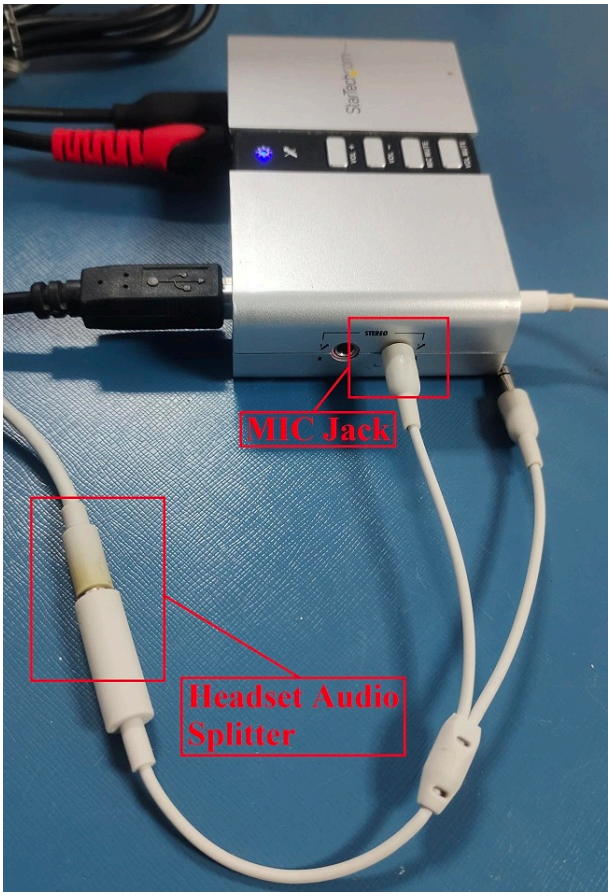
In this test, record the audio input from the mono microphone connected to a USB audio extender, transmit it through the S/PDIF-Out of the USB audio extender, and record an audio file received via the SPDIF RX peripheral using the ALSA utility on SAMA7G54-EK.



The SPDIF receiver is the component that receives the digital audio signal, and it decodes the SPDIF signal back into an audio signal that the device can process.

1. Plug the audio splitter cable into the headphone jack.
2. Connect the microphone plug from the audio splitter to the MIC 1 In port (right channel).

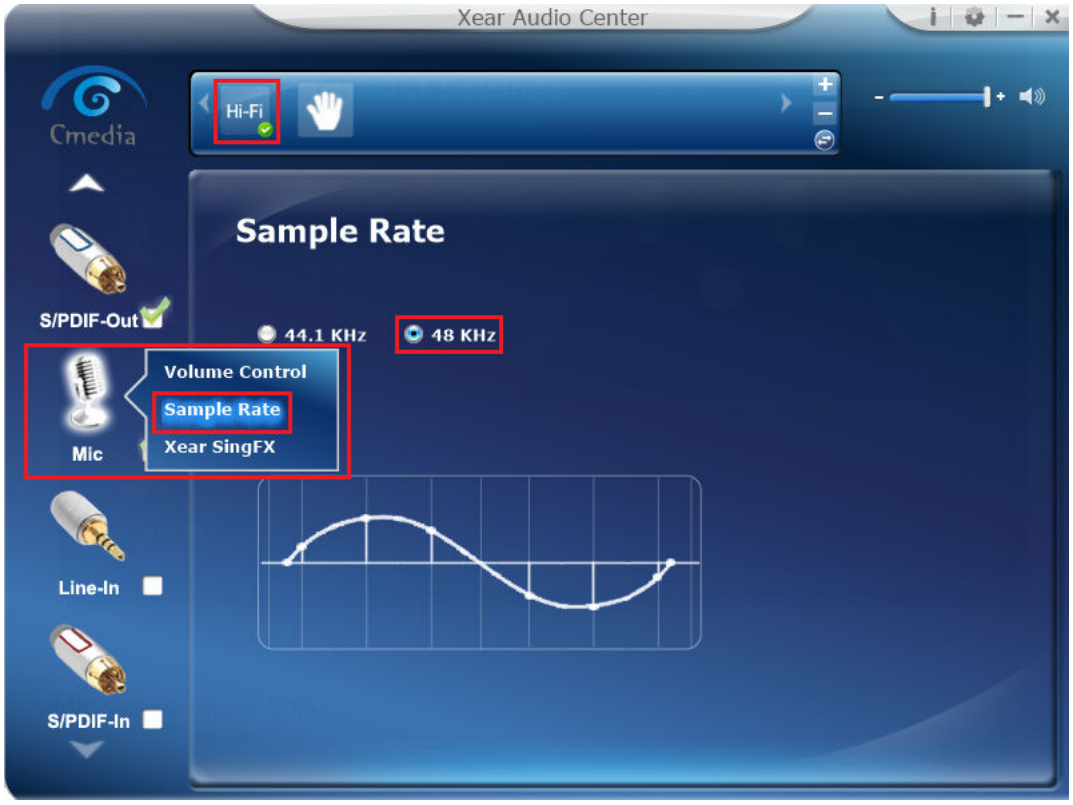
Important: To split the audio and microphone signals, use a headphone audio splitter. Connect the microphone cable from the splitter to the 3.5 mm mono jack input on the USB audio extender.



6.3.1 Host PC Setup

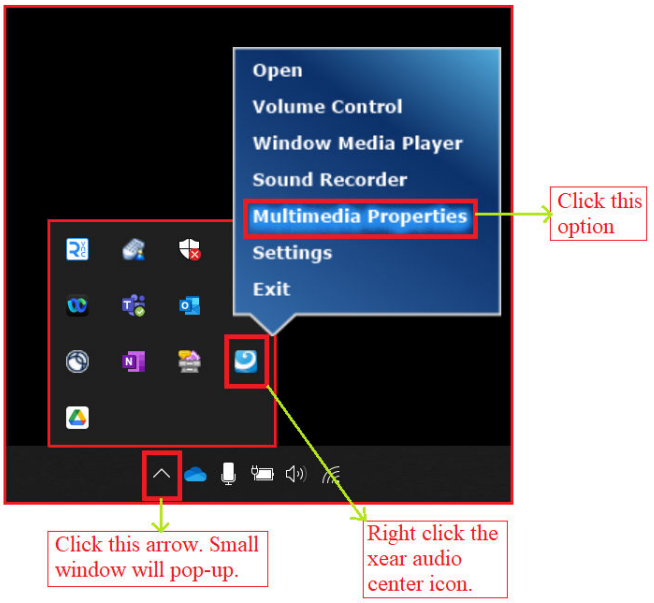
1. Open the *Xear audio center* application and select the input and output options to configure the USB audio extender.

2. Select the MIC as the input for the USB audio extender in the application, set the sample rate to 48 kHz and select the HiFi mode as shown in the following figure.

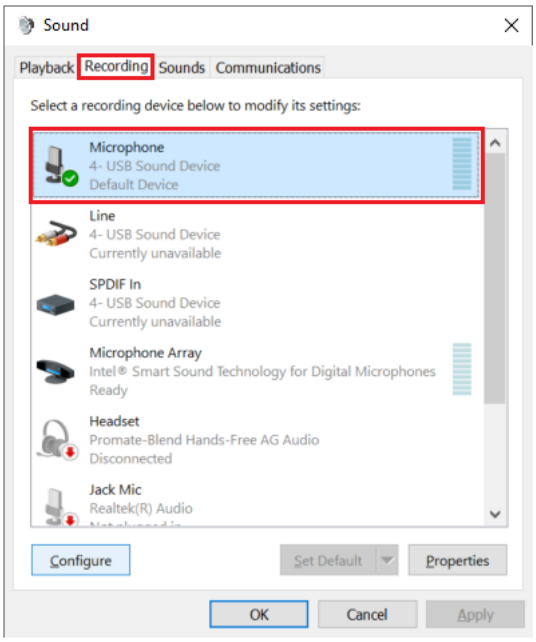


3. Right click on *S/PDIF-Out* and select *SPDIF-Out Setting* to configure *SPDIF Out* as the output for the USB audio extender.

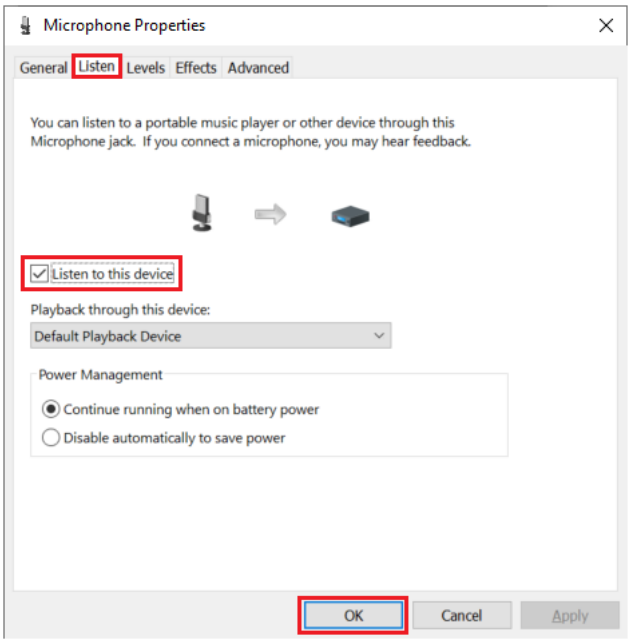
4. Move to the lower-right corner of the window and click on the upward-pointing arrow to reveal the hidden software icons. Then, right-click on the *Xear audio center* icon and select *Multimedia Properties*.
   Note: If the *Xear audio center* icon is not present, refer to the *CM6206 Software_User_Manual_Xear_Audio_Center*, available in the download folder.



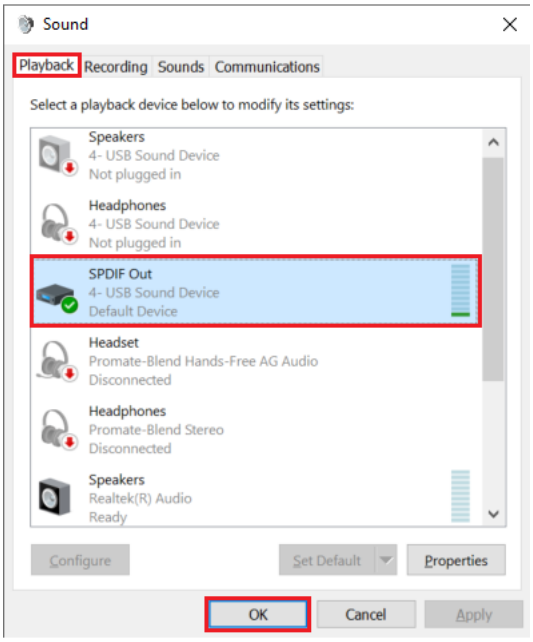
5. In the Media Properties page on the Recording tab, ensure that *Microphone* is selected. Then, double click *Microphone* to open the *Microphone Properties* dialog box.



6. In the *Microphone Properties* window, go to the *Listen* tab. Then, enable the *Listen to this device* feature by checking the corresponding checkbox and click *OK*.



7. In the Media Properties page on the Playback tab, ensure that *SPDIF Out* is selected.



6.3.2 ALSA Testing Procedure

1. Connect one end of a *TOSLINK optical cable* to the *SAMA7G54-EK SPDIF RX* and the other end to the *SPDIF-Out* on the USB audio extender.
2. To check the *SPDIF RX* audio device card number and device number, use the following command:

   ```
   $ arecord -l | grep spdifrx
   ```

   In the capture below, the card number is 0 and the device is 8.

   ```
   # arecord -l | grep spdifrx
   card 0: EK [mchp-asrc-card @ sama7g5 EK], device 8: mchp-spdifrx - dir-hifi dir-hifi-8 [mchp-spdifrx - dir-hifi dir-hifi-8]
   ```

3. Receive audio through the *SPDIF RX* interface using the following command:

   ```
   $ arecord -Dplughw:0,8 -r 48000 -c 2 -f S16_LE spdifrx.wav
   ```

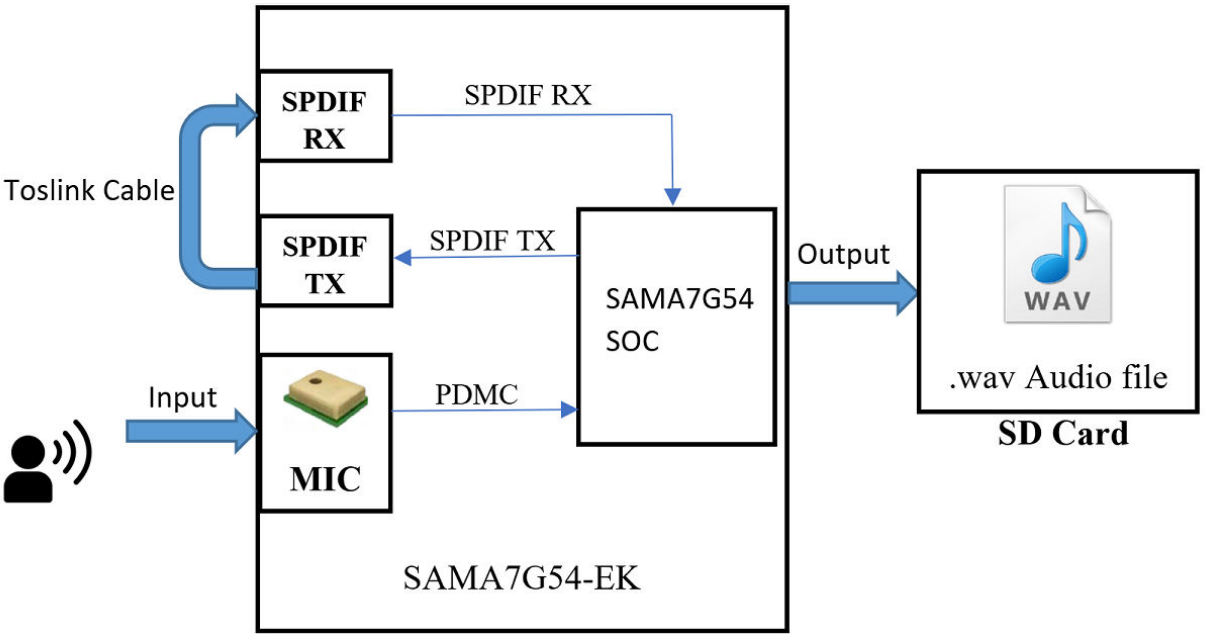
```
# arecord -Dplughw:0,8 -r 48000 -c 2 -f S16_LE spdifrx.wav
Recording WAVE 'spdifrx.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Aborted by signal Interrupt...
```

To stop recording, press *Ctrl* + *C*.

The *.wav* file is saved to the SD card in the */root* directory.

4. To play the recorded audio, either use the SPDIF Transmitter on the SAMA7G54-EK or transfer the audio file from the SAMA7G54-EK to the host PC using the Secure Copy Protocol (SCP) and listen to it on the host PC.

6.4 Test 4: Loopback Testing of Microphone, SPDIF Transmitter and Receiver



1. To check the PDM microphones, SPDIF output (TX) and input (RX) without a USB audio extender, perform a loopback test.
2. To find the card number and device ID of the PMD microphones, SPDIF transmitter and SPDIF receiver, refer to the QUAD PMD Microphones, SPDIF Transmitter and SPDIF Receiver sections.
3. To perform the loopback test, use the following command:

```
$ arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE | aplay -Dplughw:0,6 |
arecord -Dplughw:0,8 -c 2 -r 48000 -f S16_LE test_loop_back.wav
```

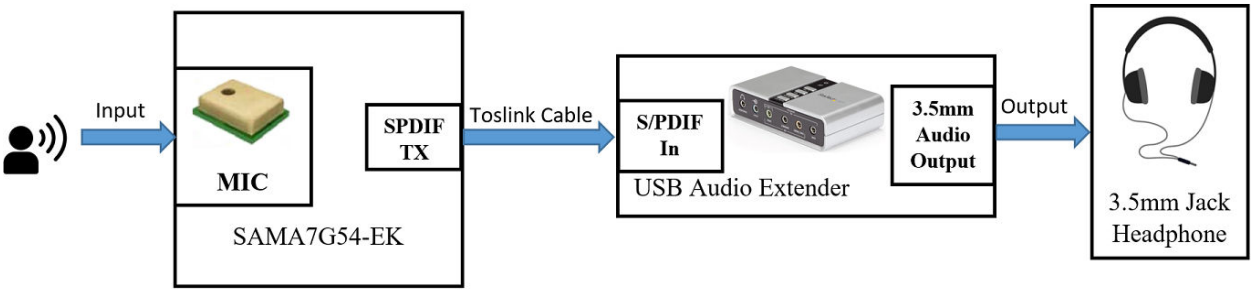
```
# arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE |  aplay -Dplughw:0,6 | arecord -D
Dplughw:0,8 -c 2 -r 48000 -f S16_LE test_loop_back.wav
Recording WAVE 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Recording WAVE 'test_loop_back.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Playing WAVE 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

To stop recording, press *Ctrl* + *C*.

The *.wav* file is saved to the SD card in the */root* directory.

4. To play the recorded audio, either use the SPDIF Transmitter on the SAMA7G54-EK or transfer the audio file from the SAMA7G54-EK to the host PC using the Secure Copy Protocol (SCP) and listen to it on the host PC.

6.5 Test 5: Microphone Audio Live Streaming via SPDIF Transmitter



1. To apply the pdmc0 device-tree overlay at u-boot level, refer to the Apply the Device Tree Overlay to Enable PDM Microphones section.
2. To find the card number and device ID of the PMD microphones, SPDIF transmitter and SPDIF receiver, refer to the QUAD PMD Microphones, SPDIF Transmitter, and SPDIF Receiver sections.
3. To configure the *Xear audio center* application, refer to SPDIF Transmitter.
4. To stream capture sound from the PDM microphones through SPDIF TX to the headphones of the USB audio extender, use the following command:

```
$ arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE | aplay -Dplughw:0,6
```

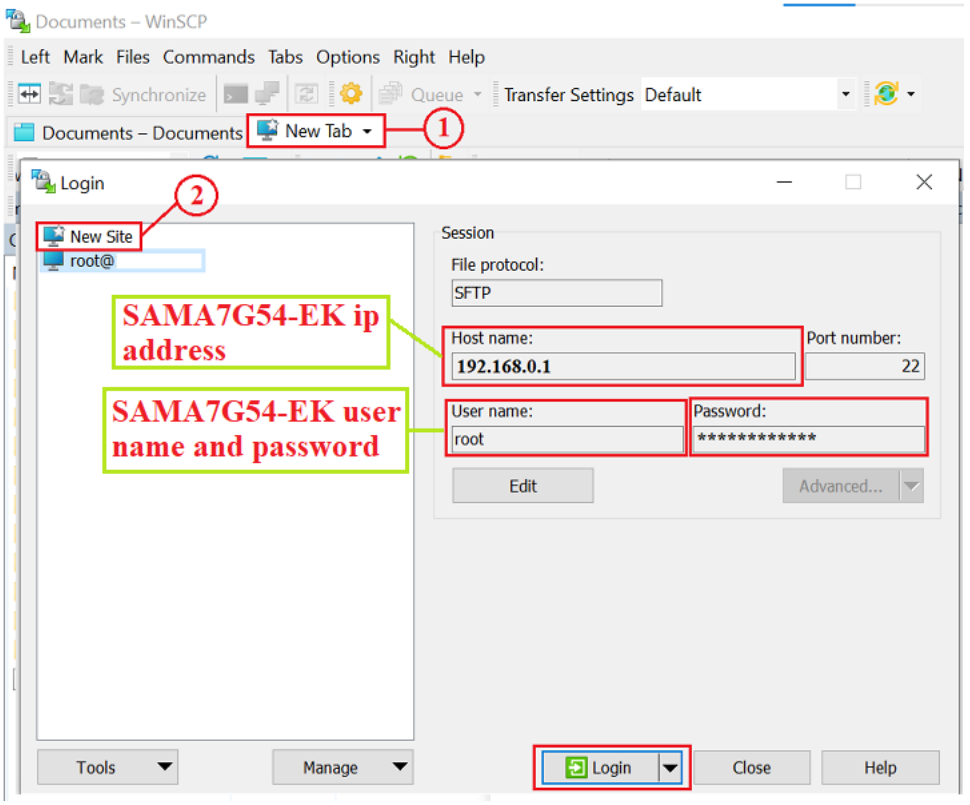
```
# arecord -Dplughw:0,4 -r 48000 -c 2 -f S16_LE | aplay -Dplughw:0,6
Recording WAVE 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Playing WAVE 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

To stop streaming, press *Ctrl* + *C*.

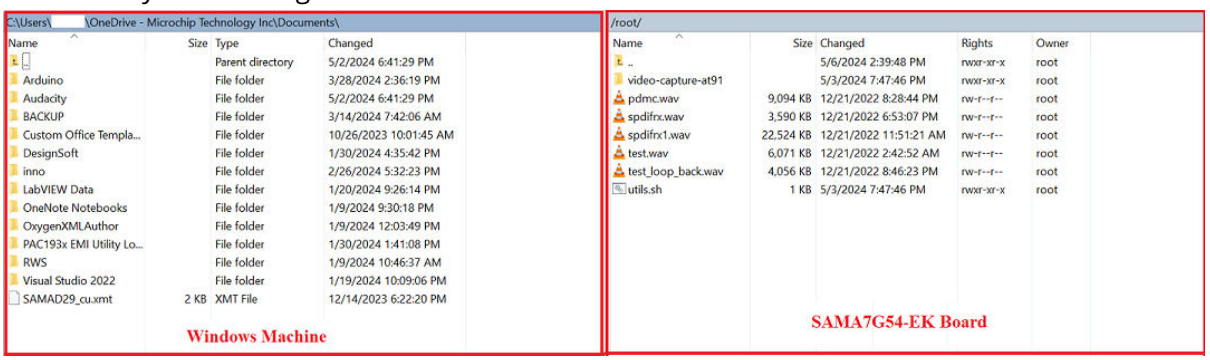
7. FAQ

7.1 Secure Copy Protocol (SCP) for File Transfer to a Windows PC

1. Open the WinSCP software.
2. Connect the Ethernet cable to the 1 Gbps Ethernet (J5) port.
3. The PC and the Ethernet cable connected to SAMA7G54-EK should be on the same network.
4. Find the IP address of the SAMA5G54-EK board using the *ifconfig* command.
5. To initiate a new session and establish a connection to the SAMA7G54-EK, enter *the board's host name* (i.e., the IP address of the SAMA7G54-EK board), *the username* for the SAMA7G54-EK board, and the corresponding password. Once all the information is entered, click *Login*.



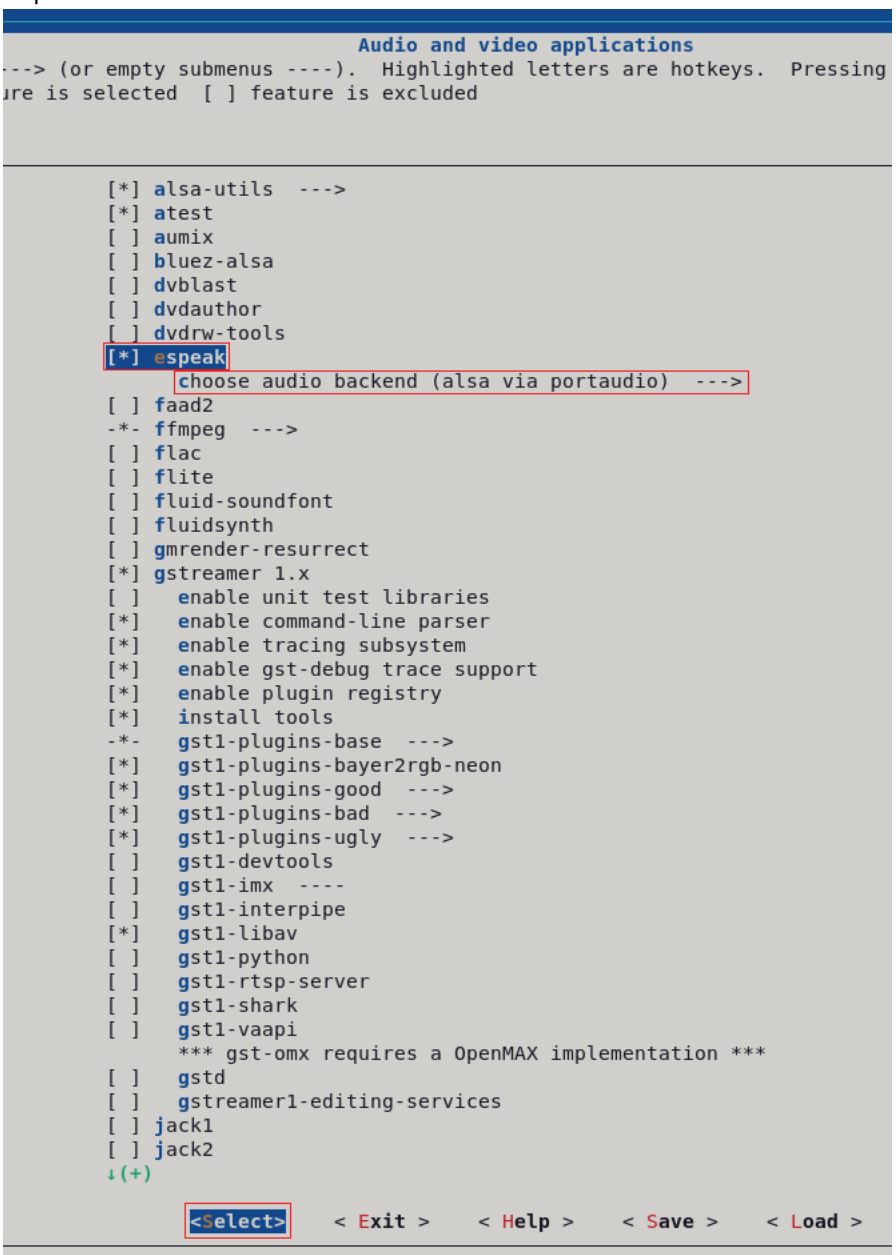
6. Transfer the file by dragging it from the left side window to the right side window, or alternatively from the right side window to the left side window.

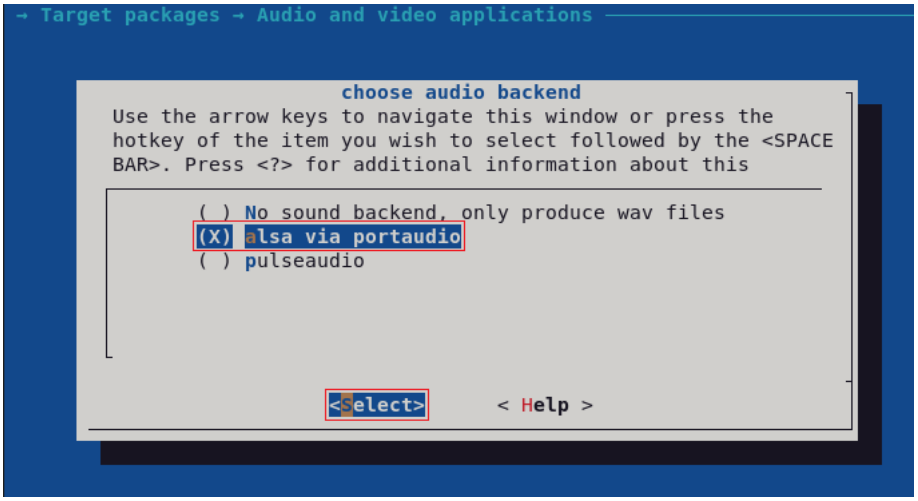


7.2 Adding an espeak Tool

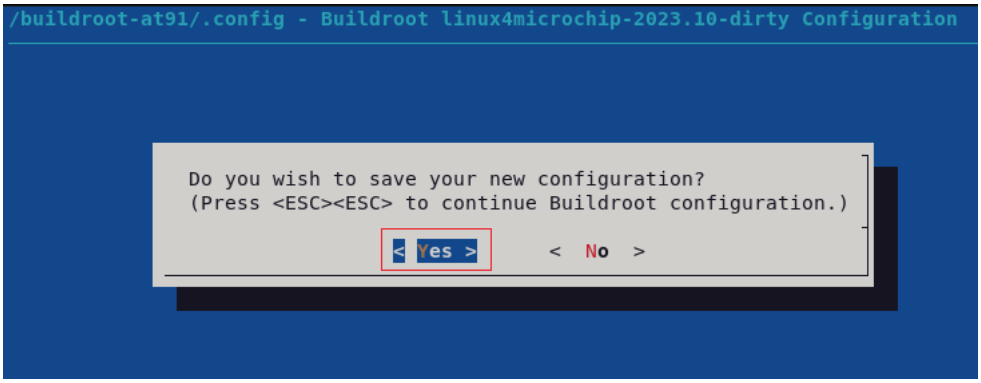
1. Enter the following command to modify the configurations:
   ```
   $ make menuconfig
   ```
   The *Buildroot Configuration* window opens.

2. In the *Target packages* window, select *Audio and video application* and press *Enter*.
3. In the *Audio and video application* window, select *espeak* and *choose audio backend (alsa via portaudio)* and press *Enter*.





4. Move to `<Exit>` with the arrows and press *Enter* to save the configuration and exit from this screen.



5. To test the audio peripheral using espeak, use the following command:
   `$ espeak "Hello world" --stdout | aplay -Dplughw:0,6`

8. References

- Pre-built Buildroot image
- Alsa aplay/arecord commands

9. Revision History

9.1 Rev. A - 02/2025

First issue.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-0557-4

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Product Page Links

SAMA7G54