

A2B ALSA USER GUIDE

Document Status	Approved
Approved by	ASH

ANALOG DEVICES, INC.

www.analog.com

Revision List

Table 1: Revision List

Revision	Date	Description
0.1	12-Feb-2019	Draft Version
0.2	13-Feb-2019	Including review comments
0.3	14-Feb-2019	Addressed additional review comments
1.0	14-Feb-2019	Approved and Baselined
1.1	10-Aug-2021	Modified for 1.0.0_Alpha release- 1) Support for Yocto Linux added 2) ALSA driver patches for following platforms – a. ADSP-SC584 b. ADSP-SC573 c. ADSP-SC589-mini d. ADSP-SC594
2.0	11-Aug-2021	Approved and Baselined
2.1	15-Feb-2023	Modified for yocto-2.1.0 for Linux 19.4.4 release 1) Support for Yocto Linux added 2) ALSA driver patches for following platforms – a. ADSP-SC589-mini b. ADSP-SC598
2.2	09-Mar-2023	Addressed review comments
3.0	16-Mar-2023	Approved and Baselined for Linux 19.4.4 release

Copyright, Disclaimer Statements

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Table of Contents

Revision List.....	2
Copyright, Disclaimer Statements	3
Table of Contents.....	4
List of Figures	5
List of Tables.....	5
1 Introduction.....	6
1.1 Scope & Assumptions	6
1.2 Organization of the Guide	6
2 Package Details	7
3 A2B Sound card on ADSP-SC5xx EZ kit	8
3.1 Software Pre-requisites.....	8
3.2 Building and Installing SDK	8
3.3 Linux Kernel with A2B Sound Card	8
3.3.1 Apply the Patch	8
3.3.2 Configure and Build Linux Kernel	9
3.3.2.1 Enable sound card driver	9
3.3.2.2 Creating recipe and adding into Yocto build.....	10
3.3.2.3 Enable alsa-lib support and Compile alsa-utils into Linux image	10
3.4 Example Application.....	11
4 Modifying A2B configuration	13
5 Roadmap	14
Terminology	15
References.....	15

List of Figures

Figure 1: A2B ALSA Model.....	7
Figure 2: Applying A2B patch file (e.g., sc589-mini platform).....	9
Figure 3: Sound card settings for SC589-Mini platform.....	10
Figure 4: Sound card settings for SC598 platform	10
Figure 5: Sample Linux kernel boot log for SC598.....	12
Figure 6: Enabling audio using A2B sound card	12

List of Tables

Table 1: Revision List	2
Table 2: a2b-alsa-driver directory details	7
Table 3: Terminology.....	15
Table 4: References	15

1 Introduction

A2B Advanced Linux Sound Architecture (ALSA) driver package enables the user to integrate 'A2B initialization sequence' into embedded Linux-kernel and represent A2B network as ALSA device. The package contains a platform independent A2B ALSA Codec driver. It also contains machine & platform driver updates required for ADSP-SC58x, and ADSP-SC59x Linux kernel to realize A2B ALSA application.

1.1 Scope & Assumptions

The scope of document is to illustrate A2B ALSA Software package & demonstrate A2B sound card on ADI platforms- ADSP-SC58x and ADSP-SC59x EZ kits. It is assumed that the recipient of this package has basic knowledge of ALSA.

Support for A2B ALSA driver based approach added for *release/yocto-2.1.0* branch. And currently verified with SC589mini and SC598 platforms. Downstream audio is not tested in this release.

1.2 Organization of the Guide

Section 1: Introduction and Scope

Section 2: Covers the details of the various folders in the A2B Stack Software for Linux.

Section 3: Covers the steps to enable A2B sound card on SC5xx EZ kit. This section may be skipped if user intends to deploy the driver on a custom platform directly.

Section 4: Covers the ways to modify the A2B configuration for a generic system

Section 5: Captures the roadmap for A2B ALSA driver

2 Package Details

After the installation of this package, the various files related to ALSA are in the `'/opt/analog/a2b-alsa-driver/19.X.Y/Yocto/<a2b-adsp-sc5xx>'` folder.

Here `<a2b-adsp-sc5xx>` can be following folder based on platform

a2b-adsp-sc589_mini
a2b-adsp-sc59x

The various folders in the `'a2b-alsa-driver'` are explained in Table 2.

Table 2: a2b-alsa-driver directory details

Folder Name	Purpose
a2bpatch	This folder contains a patch file to be applied on 'Yocto Linux' kernel to realize A2B sound card. This folder is provided for each platform.
build	This folder contains the source files for the A2B ALSA. This includes A2B codec driver, PCM driver, platform drivers & machine driver. Folder structure is mapped to SC5xx Yocto kernel. The mapping of various files into ALSA model is captured in Figure 1. This folder is provided for each platform.
Docs	Contains this document & the license agreement (GPL 2.0). This folder is common for all platforms(<code>'/opt/analog/a2b-alsa-driver/19.X.Y/Yocto/Docs'</code>)

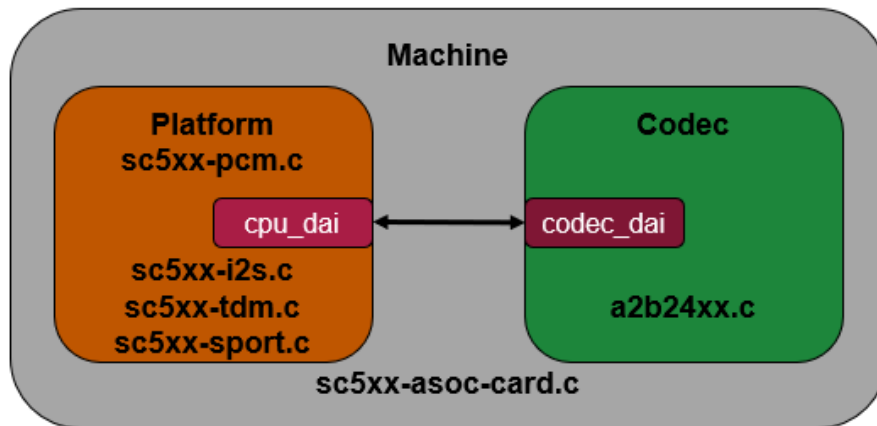


Figure 1: A2B ALSA Model

The following section explains how A2B sound card can be enabled on a reference ADI platform - ADSP-SC58x and ADSP-SC59x EZ kit. For a custom platform, the A2B codec driver (available at `build/tmp/work-shared/adsp-sc5xx-ezkit/kernel-source/sound/soc/codecs/`) can be used as-is. However, the platform drivers need to be updated specifically for custom Linux port.

3 A2B Sound card on ADSP-SC5xx EZ kit

This section may be skipped if user intends to deploy the driver on a custom platform directly. Also, this section assumes working knowledge with 'Yocto Linux for SC5xx'.

3.1 Software Pre-requisites

1. Linux Host PC running Ubuntu 20.04, 64-bit.
2. Yocto Linux for SC5xx – Refer [4] for ADSP-SC589 Mini and [5] for ADSP-SC598 platform.
3. Cross Core Embedded Studio (CCES) 2.11.0 or later
4. adi-a2b_alsa_driver-linux-amd64-19.X.Y.deb – For example ALSA application on A2B

3.2 Building and Installing SDK

To build and Install SDK on ADSP-SC5xx, follow steps in Section 6.1 for SC589 Mini and SC598 in Linux User Guide [3].

3.3 Linux Kernel with A2B Sound Card

To enable the A2B sound card driver in the Linux kernel follow the steps below.

1. Apply the patch
2. Configure and Build Linux Kernel

3.3.1 Apply the Patch

The patch file (*/opt/analog/a2b-alsa-driver/19.X.Y/Yocto/a2b-adsp-sc5xx/a2bpatch/a2bpatchfile*) enables the A2B sound card support in the supplied **Yocto Linux for ADSP-SC5xx**. On Linux machine, navigate to the *Yocto* working directory and apply the a2b patch file to the build folder

```
$ cd ~/<Yocto_Directory>
$ source setup-environment -m <MACHINE>
```

Append below line into "*~/<Yocto_Directory>/build/conf/local.conf*" file

```
LINUX_VERSION = "5.4.0"
```

Then run below commands

```
$ devtool modify linux-adi
$ cd workspace/sources
```

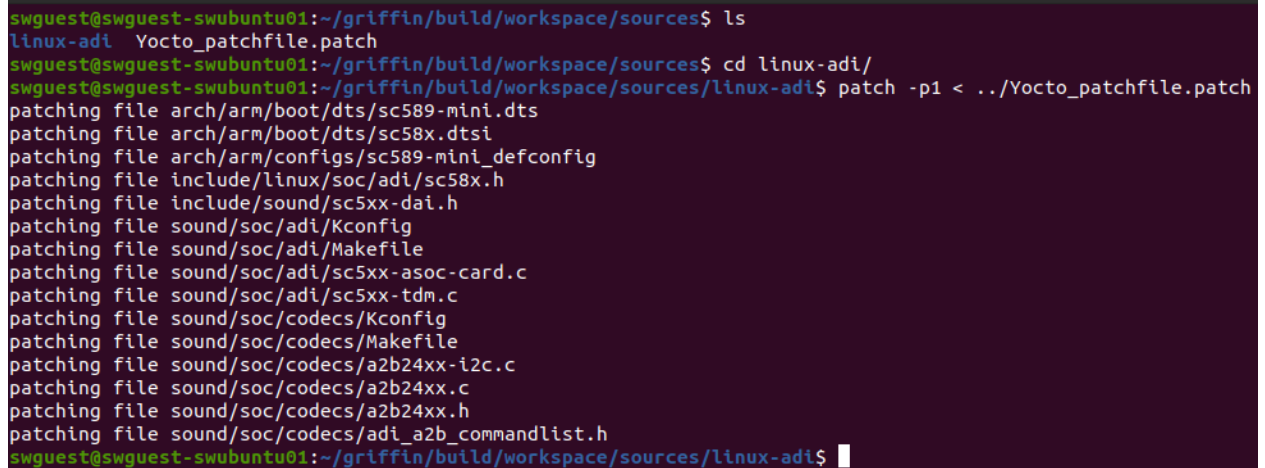
Copy the patch file into *~/<Yocto_Directory>/build/workspace/sources/* directory and run below commands

```
$ cd linux-adi
$ patch -p1 < ../Yocto_patchfile.patch
$ cd ~/<Yocto_Directory>/build
$ devtool modify linux-adi
$ devtool build linux-adi
```

<Yocto_Directory> is the path where Yocto for SC5xx is installed.

Here set the <MACHINE> one of the following based on your platform.

```
adsp-sc589-mini
adsp-sc598-som-ezkit
```



```
swgquest@swgquest-swubuntu01:~/griffin/build/workspace/sources$ ls
linux-adi  Yocto_patchfile.patch
swgquest@swgquest-swubuntu01:~/griffin/build/workspace/sources$ cd linux-adi/
swgquest@swgquest-swubuntu01:~/griffin/build/workspace/sources/linux-adi$ patch -p1 < ../Yocto_patchfile.patch
patching file arch/arm/boot/dts/sc589-mini.dts
patching file arch/arm/boot/dts/sc58x.dtsi
patching file arch/arm/configs/sc589-mini_defconfig
patching file include/linux/soc/adi/sc58x.h
patching file include/sound/sc5xx-dai.h
patching file sound/soc/adi/Kconfig
patching file sound/soc/adi/Makefile
patching file sound/soc/adi/sc5xx-asoc-card.c
patching file sound/soc/adi/sc5xx-tdm.c
patching file sound/soc/codecs/Kconfig
patching file sound/soc/codecs/Makefile
patching file sound/soc/codecs/a2b24xx-i2c.c
patching file sound/soc/codecs/a2b24xx.c
patching file sound/soc/codecs/a2b24xx.h
patching file sound/soc/codecs/adi_a2b_commandlist.h
swgquest@swgquest-swubuntu01:~/griffin/build/workspace/sources/linux-adi$
```

Figure 2: Applying A2B patch file (e.g., sc589-mini platform)

3.3.2 Configure and Build Linux Kernel

Follow steps below to configure Linux Kernel

3.3.2.1 Enable sound card driver

1. Enable soundcard devices based on platforms.
2. Open a terminal and navigate to Yocto directory. Give “source setup-environment -m <MACHINE>” command to setup source environment.

Here set the <MACHINE> one of the following based on your platform.

```
adsp-sc589-mini
adsp-sc598-som-ezkit
```

3. Give “*bitbake linux-adi -c menuconfig*” command and navigate to “Advanced Linux Sound Architecture”
4. Then enable A2B sound card driver as follows,

```
Device Drivers --->
  <*> Sound card support --->
    <*> Advanced Linux Sound Architecture --->
      <*> ALSA for SoC audio support --->
        <*> SoC Audio for the ADI SC5XX chip
          <*> Support for the A2B AD24XX board on SC5XX ezkit board
```

5. Ensure that ADAU1962, ADAU1979 sound card driver is enabled for ADSP-SC598 platforms and ADAU1761 sound card driver is enabled for ADSP-SC589-Mini.

```
-- ALSA for SoC audio support
<*> SoC Audio for the ADI SC5XX chip
      SPORT data feed (Regular SPORT operation) --->
<*> Support for the ADAU1761 Machine driver on SHARC Audio board
<*> Support for the A2B AD24XX board on SC5XX ezkit board
```

Figure 3: Sound card settings for SC589-Mini platform

```
-- ALSA for SoC audio support
<*> SHARC-ALSA SoC Audio card for the ADI SC5XX chip
<*> SoC Audio for the ADI SC5XX chip
      SPORT data feed (Regular SPORT operation) --->
< > Support for the ADAU1761 Machine driver on SHARC Audio board
<*> Support for the ADAU1979 Machine driver on SC5XX ezkit board
<*> Support for the ADAU1962 Machine driver on SC5XX ezkit board
<*> Support for the A2B AD24XX board on SC5XX ezkit board
```

Figure 4: Sound card settings for SC598 platform

6. Then run “*bitbake linux-adi -C compile*” command build changes.

3.3.2.2 Creating recipe and adding into Yocto build

Follow steps from Section 6.2.1 in Linux User Guide [3] for recipe creation and A2B application code addition into Yocto build.

3.3.2.3 Enable alsa-lib support and Compile alsa-utils into Linux image

To enable ALSA library support, add recipes (*alsa-utils*, *alsa-lib*, *mplayer-common*, *play*) to ramdisk image and build ramdisk image

1. Append to recipes to “IMAGE_INSTALL” in the file Yocto_Directory/sources/meta-adi/meta-adi-adsp-sc5xx/recipes-adi/images/adsp-sc5xx-ramdisk.bb with

```
IMAGE_INSTALL = " \
    packagegroup-core-boot \
    packagegroup-base \
    busybox-watchdog-init \
    alsa-utils \
    alsa-lib \
    mplayer-common \
    play \
"
```

2. Remove/Comment below lines from “Yocto_Directory/sources/meta-adi/meta-adi-adsp-sc5xx/recipes-adi/images/adsp-sc5xx-ramdisk.bb” file.

```
rm -rf ${IMAGE_ROOTFS}/usr/share/alsa
rm -rf ${IMAGE_ROOTFS}/usr/lib/libasound.so.2.0.0
rm -rf ${IMAGE_ROOTFS}/usr/lib/libasound.so.2
rm -rf ${IMAGE_ROOTFS}/usr/sbin/alsactl
```

3. Then run “bitbake adsp-sc5xx-ramdisk -C compile” to compile alsa-utils into Linux image.
4. In order to load Linux, a copy of the FIT IMAGE FILE and RAMDISK FILE should be copied into the /tftpboot directory

```
$ cp tmp/deploy/images/<MACHINE>/fitImage /tftpboot
$ cp tmp/deploy/images/<MACHINE>/<RAMDISK_FILE> /tftpboot/ramdisk.cpio.xz.u-boot
```

MACHINE	DTB_FILE	RAMDISK_FILE
adsp-sc589-mini	sc589-mini.dtb	adsp-sc5xx-ramdisk-adsp-sc589-mini.cpio.xz.u-boot
adsp-sc598-som-ezkit	sc598-som-ezkit.dtb	adsp-sc5xx-ramdisk-adsp-sc598-som-ezkit.cpio.xz.u-boot

5. To Load Linux on ADSP-SC5xx follow steps in Section 7.4 for SC589 Mini and Section 8.4 for SC598 in Linux User Guide [3].

3.4 Example Application

Once kernel gets loaded, A2B network appears as an ALSA device capable of both playback & record. Now, a user mode application can be used to route the audio to A2B network. One such example application is provided with software – ‘adi-a2b_alsa_driver-linux-amd64-19.X.Y’. This package contains an ALSA application to route audio between A2B network & onboard codec.

Refer document [3] to understand the example demo & hardware set up details. By default, the example application configures A2B network from user mode (using A2B Stack). And audio path is handled by SHARC core1. However, the application can be configured to run ALSA application (where the A2B set up is part of kernel boot). Steps to run ALSA example application is explained below.

1. Build Linux kernel supporting A2B sound card as described in Section 3.3.2
2. Set up the hardware as described in the document [3]
3. Boot the Linux kernel. Refer to Section 7 in Linux User Guide [3] for SC589 Mini and Section 8 for SC598.

Note: Ensure that the SHARC core audio path is disabled.

```

00:00:00.00
sc5xx-l2s-dal 31002400.l2s: SPORT create success
sc5xx-tdm 31002000.tdm: SPORT create success
snd-sc5xx scb:sound: adau1962-hifi <-> 31002400.l2s mapping ok
snd-sc5xx scb:sound: adau1977-hifi <-> 31002400.l2s mapping ok
snd-sc5xx scb:sound: a2b24xx-hifi <-> 31002000.tdm mapping ok
ICP: cubic registered
NET: Registered protocol family 17
can: controller area network core (rev 20120528 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20120528)
can: broadcast manager protocol (rev 20120528 t)
can: netlink gateway (rev 20130117) max_hops=1
do_initcall_level level 7
ThumbEE CPU extension supported.
console [netcon0] enabled
netconsole: network logging started
/home/aceguest/sc5xx_dev/bulldroot/linux/linux-kernel/drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
IP-Config: Complete:
  device=eth0, hwaddr=02:80:ad:20:31:e8, ipaddr=192.168.0.15, mask=255.255.255.0, gw=192.168.0.1
  host=sc5xx, domain=, nis-domain=(none)
  bootserver=192.168.0.2, rootserver=192.168.0.2, rootpath=
ALSA device list:
  #0: sc5xx-asoc-card
Freeing unused kernel memory: 16860K (c0503000 - c157a000)
Starting logging: OK
Starting mdev...
Starting watchdog...
Initializing random number generator... random: dd urandom read with 29 bits of entropy available
done.
Starting system message bus: done

```

Figure 5: Sample Linux kernel boot log for SC598

4. Refer Section 6.1 from Linux User Guide [3] to build and install SDK
5. Refer Section 6.3 from Linux User Guide [3] to build the 'a2bapp-linux'. Define the macro 'A2B_ALSA_AUDIO' in /opt/analog/a2b-software/X.Y.Z/Target/examples/demo/a2b-linux/a2b-adsp-sc5xx-linux/a2b-app-linux_Core0/app/a2bapp_defs.h. A reference implementation of audio handling routines using ALSA API's is implemented in the /opt/analog/a2b-software/X.Y.Z/Target/examples/demo/a2b-linux/a2b-adsp-sc5xx-linux/a2b-app-linux_Core0/a2bstack-pal/pal_audio.c
6. Load the application & run with "-k" as a command line option as shown in Figure 6.

```
# ./a2bapp-linux -k
```

Figure 6: Enabling audio using A2B sound card

4 Modifying A2B configuration

A2B ALSA driver takes '*adi_a2b_i2c_commandlist.h*' file (build/tmp/work-shared/adsp-sc5xx-ezkit/kernel-source/sound/soc/codecs/) as input to configure the A2B network. This file contains series of I2C commands to discover & configure A2B nodes (including the remote devices). The default command file corresponds 1 master and 2 slave system. For custom configuration, command list file needs to be updated.

User may modify the command list example/template file manually. Alternative option is to generate the I2C command list from SigmaStudio. It allows the user to graphically represent A2B network & generate the corresponding *adi_a2b_commandlist.h* seamlessly. The procedure to generate the command list is explained in the document [2].

5 Roadmap

The following feature enhancements are planned for future A2B ALSA driver releases.

- Integrate A2B diagnostics into kernel driver
- Remove build time dependency between A2B command list & driver
- Configuration tool integrated with kernel build to generate configuration set for ALSA driver
- Automotive production qualified driver
- Merge A2B drivers into upstream kernel tree

Terminology

Table 3: Terminology

Term	Description
A2B	Automotive Audio Bus
ALSA	Advanced Linux Sound Architecture
A2B node	Refers to AD241x/AD242x.
ADI	Analog Devices, Inc.
GPL	General Public License
GNU	GNU's not UNIX
Master Node	A2B transceiver that is connected to the host processor is considered as the master A2B node.
Slave Node	A2B Slave Transceiver with local peripherals such as speakers and microphones.
I2C	Is a multi-master single-ended serial bus used for attaching low-speed peripherals to a processor. In I2C protocol the serial data transmission is done in asynchronous mode. This protocol uses only two wires named <i>SDA</i> (serial data) and <i>SCL</i> (serial clock) for communicating between two or more ICs.

References

Table 4: References

Reference No.	Description
[1]	https://wiki.analog.com/resources/tools-software/linuxdsp
[2]	/opt/analog/a2b-software/X.Y.Z/Docs/AE_09_A2B_SigmaStudio_UserGuide.pdf
[3]	/opt/analog/a2b-software/X.Y.Z/Docs/AE_09_A2B_Stack_Linux_UserGuide.pdf
[4]	https://wiki.analog.com/resources/tools-software/linuxdsp/docs/quickstartguide/quickstart_sc589#installing_the_sources
[5]	https://wiki.analog.com/resources/tools-software/linuxdsp/docs/quickstartguide/quickstart_sc598