```
BlueALSA (formerly know as BluezALSA) :  https://github.com/Arkq/bluez-alsa

https://forum.armbian.com/topic/6480-bluealsa-bluetooth-audio-using-alsa-not-pu
lseaudio/

Se supone que tenemos BlueALSA as background task:
bluealsa --disable-hfp &

Se entiende que el 'usuario' pertenece al grupo 'audio'

Final Note: Bluealsa and PulseAudio do not play well together. You will have to
completely uninstall PulseAudio and all its baggage in order to use bluealsa.
O por lo menos Bluetooth support has to be disabled in the PulseAudio, any
Bluetooth related module has to be unloaded.

Note: Given your bluetooth device is a virtual device, it will not be listed
when you run aplay -l or similar listing commands.
```

Comandos Bluetooth:

$ echo help | bluetoothctl

```
[bluetooth]# help
Available commands:
  list                       List available controllers
  show [ctrl]                Controller information
```

```
  select <ctrl>              Select default controller
  devices                    List available devices
  paired-devices             List paired devices
  power <on/off>             Set controller power
  pairable <on/off>          Set controller pairable mode
  discoverable <on/off>      Set controller discoverable mode
  agent <on/off/capability>  Enable/disable agent with given capability
  default-agent              Set agent as the default one
  advertise <on/off/type>    Enable/disable advertising with given type
  set-advertise-uuids [uuid1 uuid2 ...] Set advertise uuids
  set-advertise-service [uuid][data=[xx xx ...] Set advertise service data
  set-advertise-manufacturer [id][data=[xx xx ...] Set advertise manufacturer
data
  set-advertise-tx-power <on/off> Enable/disable TX power to be advertised
  set-scan-filter-uuids [uuid1 uuid2 ...] Set scan filter uuids
  set-scan-filter-rssi [rssi] Set scan filter rssi, and clears pathloss
  set-scan-filter-pathloss [pathloss] Set scan filter pathloss, and clears rssi
  set-scan-filter-transport [transport] Set scan filter transport
  set-scan-filter-clear      Clears discovery filter.
  scan <on/off>              Scan for devices
  info [dev]                 Device information
  pair [dev]                 Pair with device
  trust [dev]                Trust device
  untrust [dev]              Untrust device
  block [dev]                Block device
  unblock [dev]              Unblock device
  remove <dev>               Remove device
  connect <dev>              Connect device
  disconnect [dev]           Disconnect device
  list-attributes [dev]      List attributes
  set-alias <alias>          Set device alias
  select-attribute <attribute> Select attribute
  attribute-info [attribute] Select attribute
  read                       Read attribute value
  write <data=[xx xx ...]>   Write attribute value
  notify <on/off>            Notify attribute value
  register-profile <UUID ...> Register profile to connect
  unregister-profile         Unregister profile
  version                    Display version
  quit                       Quit program
```

$ hcitool

```
hcitool - HCI Tool ver 5.43
Usage:
      hcitool [options] <command> [command parameters]
Options:
      --help      Display help
      -i dev      HCI device
Commands:
      dev   Display local devices
      inq   Inquire remote devices
      scan  Scan for remote devices
```

```
      name  Get name from remote device
      info  Get information from remote device
      spinq Start periodic inquiry
      epinq Exit periodic inquiry
      cmd   Submit arbitrary HCI commands
      con   Display active connections
      cc    Create connection to remote device
      dc    Disconnect from remote device
      sr    Switch master/slave role
      cpt   Change connection packet type
      rssi  Display connection RSSI
      lq    Display link quality
      tpl   Display transmit power level
      afh   Display AFH channel map
      lp    Set/display link policy settings
      lst   Set/display link supervision timeout
      auth  Request authentication
      enc   Set connection encryption
      key   Change connection link key
      clkoff      Read clock offset
      clock Read local or remote clock
      lescan      Start LE scan
      leinfo      Get LE remote information
      lewladd     Add device to LE White List
      lewlrm      Remove device from LE White List
      lewlsz      Read size of LE White List
      lewlclr     Clear LE White List
      lerladd     Add device to LE Resolving List
      lerlrm      Remove device from LE Resolving List
      lerlclr     Clear LE Resolving List
      lerlsz      Read size of LE Resolving List
      lerlon      Enable LE Address Resolution
      lerloff     Disable LE Address Resolution
      lecc  Create a LE Connection
      ledc  Disconnect a LE Connection
      lecup LE Connection Update

For more information on the usage of each command use:
      hcitool <command> --help
```

$ hcitool con

```
Connections:
      > ACL 54:E4:3A:1E:FC:92 handle 70 state 1 lm SLAVE AUTH ENCRYPT
```

--- PAIRING a BT Device:

```
bluetoothctl >
    scan on
    [NEW] Device 30:23:23:F4:48:2C TH-S10U
    scan off
```

```
    pair 30:23:23:F4:48:2C
    trust 30:23:23:F4:48:2C
    exit
```

After that power off&on the bt-device and the device will connect automatically as a trusted device.

Before we try our first audio-command we had to export one thing to make BlueALSA mor system-friendly:

```
export LIBASOUND_THREAD_SAFE=0
```

--- PLAYING Text to speech:

```
speak "Hello, how are you?" -w /home/guido/espeak.wav -s145
aplay -D bluealsa:HCI=hci0,DEV=30:23:23:F4:48:2C,PROFILE=a2dp
/home/guido/espeak.wav
```

--- PLAYING Internetradio:

```
mpg123 -a bluealsa:HCI=hci0,DEV=30:23:23:F4:48:2C,PROFILE=a2dp -@
/home/guido/ffh80s.pls
```

--- ASOUNDRC for PLAYING

~/.asoundrc with the following content:

```
    defaults.bluealsa.interface "hci0"
    defaults.bluealsa.device "30:23:23:F4:48:2C"
    defaults.bluealsa.profile "a2dp"
    defaults.bluealsa.delay 10000
```

Then:

```
mpg123 -a bluealsa -@ /home/guido/ffh80s.pls
```

--- VOLUME

For set the volume of such a device you have to know the "real" name of the device out of the bluetoothctl and the name of the used protocol:

```
    Device: 30:23:23:F4:48:2C TH-S10U
    Protocol: A2DP
amixer -D bluealsa sset 'TH-S10U - A2DP ' 70%
```

--- Ver los devices BT

```
$ echo -e "devices\nquit" | bluetoothctl
```

```
[NEW] Controller 00:1A:7D:DA:71:13 wpi [default]
[NEW] Device 54:E4:3A:1E:FC:92 iPhone
[bluetooth]# devices
Device 54:E4:3A:1E:FC:92 iPhone
[bluetooth]# quit
[DEL] Controller 00:1A:7D:DA:71:13 wpi [default]
```

--- AUTOCONNECT

After starting bluealsa and then a moment later the BT-device will auto-connect if trusted:

```
    Device 30:21:8E:AA:4C:45 ML-28U
    Device 30:23:23:F4:48:2C TH-S10U
    Device 00:11:67:3F:9B:18 BTLS9001
```

If you didnt want to auto-connect or it wouldnt autoconnect you can connect to a paired&trusted BT-Speaker with the command:

```
echo -e "connect 00:11:67:3F:9B:18\nquit" | bluetoothctl
```

Disconnect is the same:

```
echo -e "disconnect 00:11:67:3F:9B:18\nquit" | bluetoothctl
```

You could also change the commandline here for pairing or trusting a device like

```
echo -e "pair 00:11:67:3F:9B:18\trust 00:11:67:3F:9B:18\nquit" | bluetoothctl
```

--- CAPTURING

To capture audio from the connected Bluetooth device:

```
$ arecord -D bluealsa capture.wav
$ arecord -D bluealsa:HCI=hci0,DEV=XX:XX:XX:XX:XX:XX,PROFILE=sco test.wav

$ arecord -D bluealsa:HCI=hci0,DEV=54:E4:3A:1E:FC:92 -r44100 -c2 | aplay -D jack &
```

Nótese que arecord y aplay harán resampling automático salvo que se inhiba.

--- PARA VER SI UN DEVICE ESTÁ CONECTADO

```
$ echo -e "paired-devices\nquit" | bluetoothctl
[NEW] Controller 00:1A:7D:DA:71:13 wpi [default]
[NEW] Device 54:E4:3A:1E:FC:92 iPhone
[bluetooth]# paired-devices
Device 54:E4:3A:1E:FC:92 iPhone
[bluetooth]# quit
[DEL] Controller 00:1A:7D:DA:71:13 wpi [default]
$

$ echo -e "info 54:E4:3A:1E:FC:92\nquit" | bluetoothctl
[NEW] Controller 00:1A:7D:DA:71:13 wpi [default]
[NEW] Device 54:E4:3A:1E:FC:92 iPhone
[bluetooth]# info 54:E4:3A:1E:FC:92
Device 54:E4:3A:1E:FC:92
        Name: iPhone
        Alias: iPhone
        Class: 0x7a020c
        Icon: phone
        Paired: yes
        Trusted: yes
        Blocked: no
        Connected: no
        LegacyPairing: no
        UUID: Vendor specific           (00000000-deca-fade-deca-deafdecacafe)
        UUID: Service Discovery Serve.. (00001000-0000-1000-8000-00805f9b34fb)
        UUID: Audio Source              (0000110a-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
        UUID: Advanced Audio Distribu.. (0000110d-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
        UUID: NAP                       (00001116-0000-1000-8000-00805f9b34fb)
        UUID: Handsfree                 (0000111e-0000-1000-8000-00805f9b34fb)
        UUID: Handsfree Audio Gateway   (0000111f-0000-1000-8000-00805f9b34fb)
        UUID: Phonebook Access Server   (0000112f-0000-1000-8000-00805f9b34fb)
        UUID: Message Access Server     (00001132-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Vendor specific           (02030302-1d19-415f-86f2-22a2106a0a77)
        Modalias: bluetooth:v004Cp6D03d0B20
[bluetooth]# quit
[DEL] Controller 00:1A:7D:DA:71:13 wpi [default]
$
```