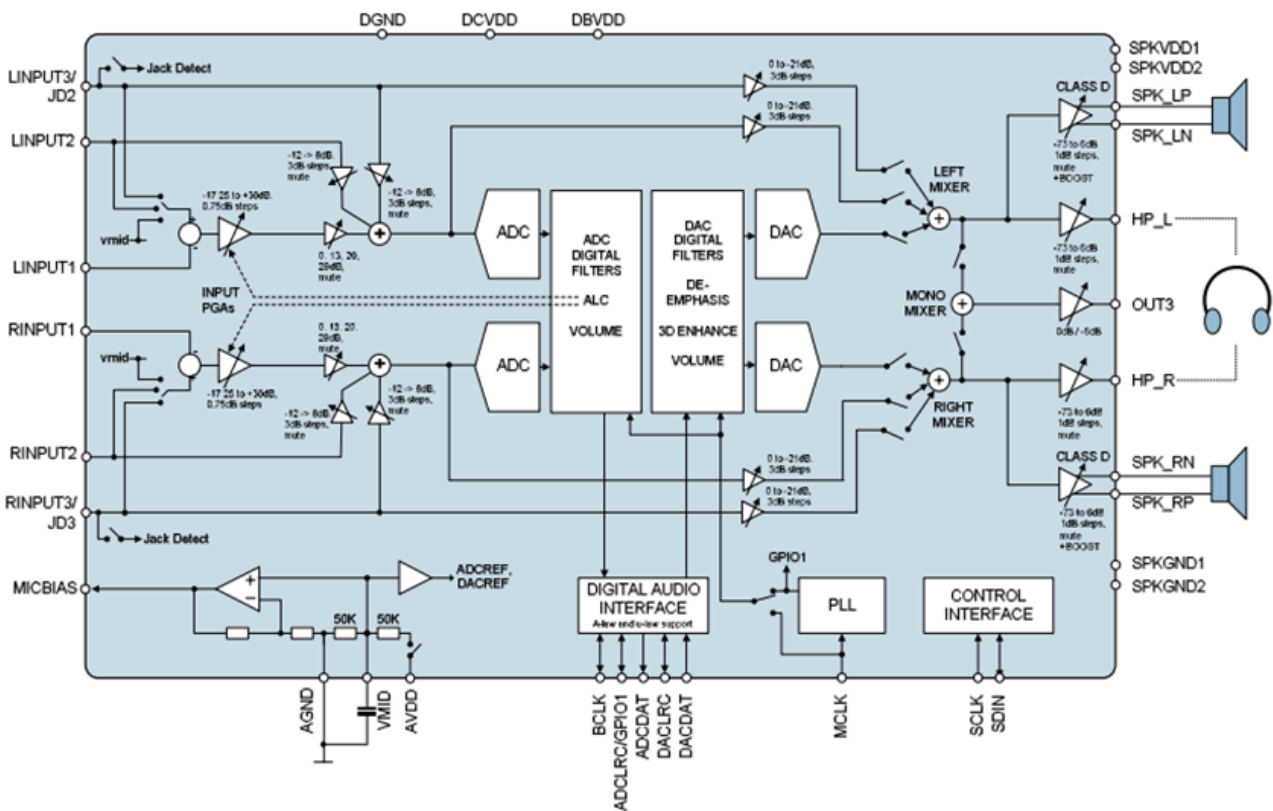


Руководство по применению модуля Mind-Audio-WM8960

1. Введение в модуль

Аудиомодуль Mind-Audio-WM8960 — это высокопроизводительное аудиорешение на базе стереокодека WM8960 с низким энергопотреблением.



Данный модуль поддерживает переключение между двумя источниками сигнала: аудиовходом или входом микрофона. Выход может быть подключен к наушникам или динамикам.

Модуль конфигурируется через интерфейс I2C и использует интерфейс I2S для передачи данных: чтение входных сигналов с АЦП и вывод записанных данных через ЦАП.

Определения выводов модуля:

№	Имя	Описание
1~4	VIN	Входное питание модуля. Поддерживает вход 5В или 3.3В (опционально при поставке)
5~6	NC	Не подключено (свободный вывод)
7~8	DGND	Земля модуля
9	AUD_BCK	BCLK WM8960, тактовый сигнал битовой частоты для аудио
10	AUD_MCLK	Master clock WM8960. Обычно предоставляется встроенным активным кварцевым генератором на 24 МГц, но также может принимать другие тактовые сигналы через этот вывод. Если внешний генератор предоставляет тактовый сигнал через этот вывод, встроенный генератор на 24 МГц необходимо отключить.
11	AUD_DACDAT	DACDAT WM8960, входные данные для ЦАП аудио
12	AUD_DACLRC	DACLRC WM8960, тактовый сигнал левого/правого канала для ЦАП аудио
13	AUD_ADCDAT	ADCDAT WM8960, выходные данные с АЦП аудио
14	AUD_ADCLRC	ADCLRC/GPIO WM8960, конфигурируется как тактовый сигнал левого/правого канала для АЦП аудио или как GPIO. При конфигурации как GPIO, DACLRC служит одновременно тактовым сигналом левого/правого канала и для ЦАП, и для АЦП.
15	AUD_SDA	SDIN WM8960, соответствует линии данных шины I2C для конфигурации WM8960
16	AUD_SCL	SCLK WM8960, соответствует линии тактирования шины I2C для конфигурации WM8960

2. Использование модуля с STM32

При выборе модели STM32, микроконтроллер STM32 должен иметь как минимум один интерфейс I2C (может быть эмулирован через GPIO) и один интерфейс I2S.

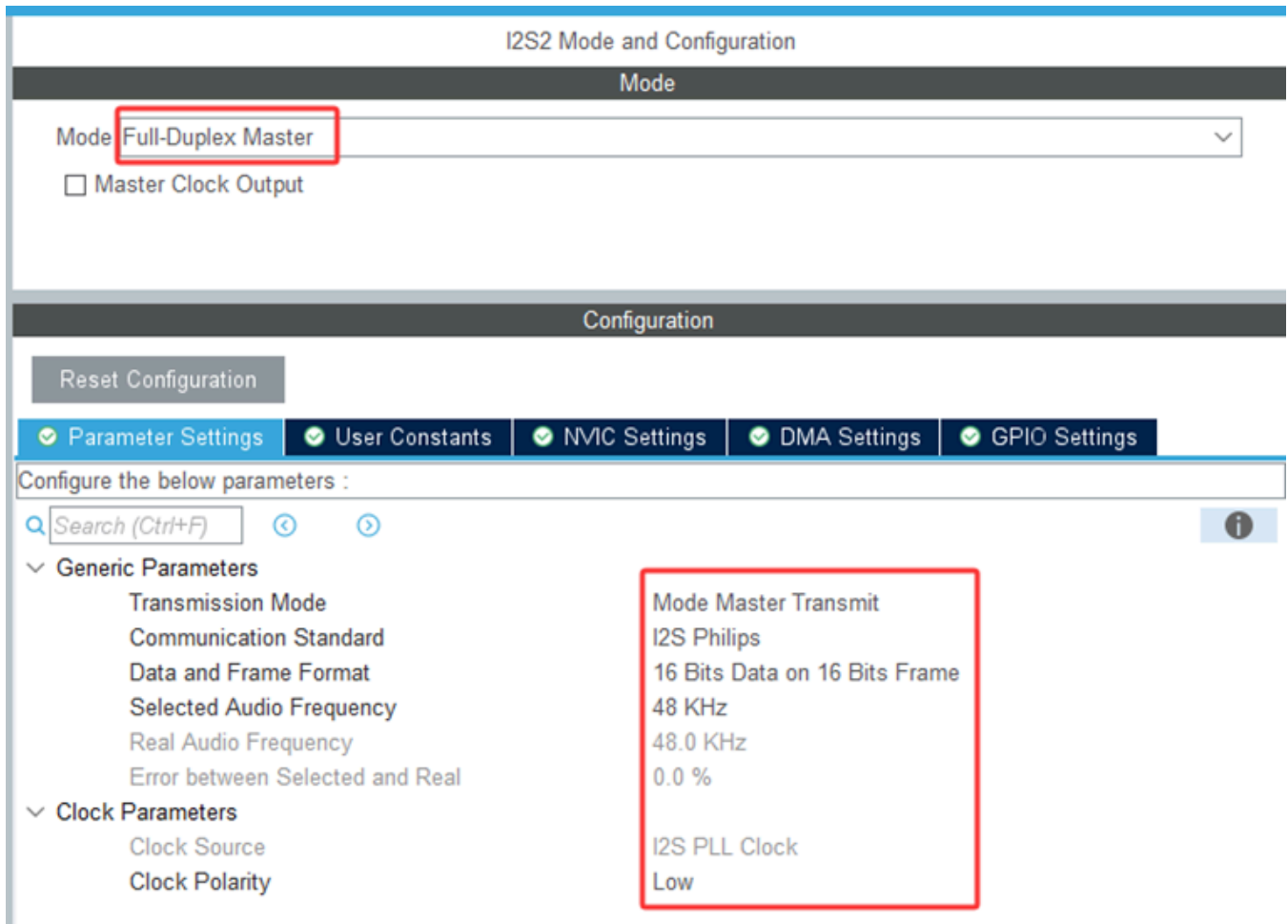
Различные чипы STM32 поддерживают разные режимы работы интерфейса I2S. Здесь в качестве примера используется STM32F429.

STM32F429 поддерживает полно-дуплексный мастер-режим, полно-дуплексный slave-режим, полу-дуплексный мастер-режим и полу-дуплексный slave-режим. Каждый режим имеет два режима передачи: передача и прием.

2.1. Режим мастера на передачу

I2S STM32 действует как мастер и настроен в режиме передачи.

Настройте интерфейс I2S, как показано ниже. Режим может быть выбран как полно-дуплексный или полу-дуплексный. Выбор полу-дуплексного режима позволяет только передачу без приема.



- Режим передачи: Выберите Mode Master Transmit;
- Стандарт связи: Выберите здесь I2S Philips. Соответственно, при настройке WM8960 он также должен быть настроен в этом режиме;
- Формат данных и кадра: Выберите здесь 16bits Data on 16 Bits Frame. Соответственно, при настройке WM8960 формат данных также должен быть настроен как 16bit;

Конфигурация DMA: Выберите Half Word для Data Width.

Parameter Settings | User Constants | NVIC Settings | **DMA Settings** | GPIO Settings

DMA Request	Stream	Direction	Priority
I2S2_EXT_RX	DMA1 Stream 3	Peripheral To Memory	High
SPI2_TX	DMA1 Stream 4	Memory To Peripheral	High

Add Delete

DMA Request Settings

Mode	Normal	Increment Address	<input type="checkbox"/>	Peripheral	<input type="checkbox"/>	Memory	<input checked="" type="checkbox"/>
Use Fifo	<input checked="" type="checkbox"/>	Threshold	Full	Data Width	Half Word	Half Word	Half Word
				Burst Size	Single	Single	Single

Конфигурация GPIO: Сигнал I2S_ext_SD появляется только при настройке в полно-дуплексном режиме.

Parameter Settings | User Constants | NVIC Settings | **DMA Settings** | **GPIO Settings**

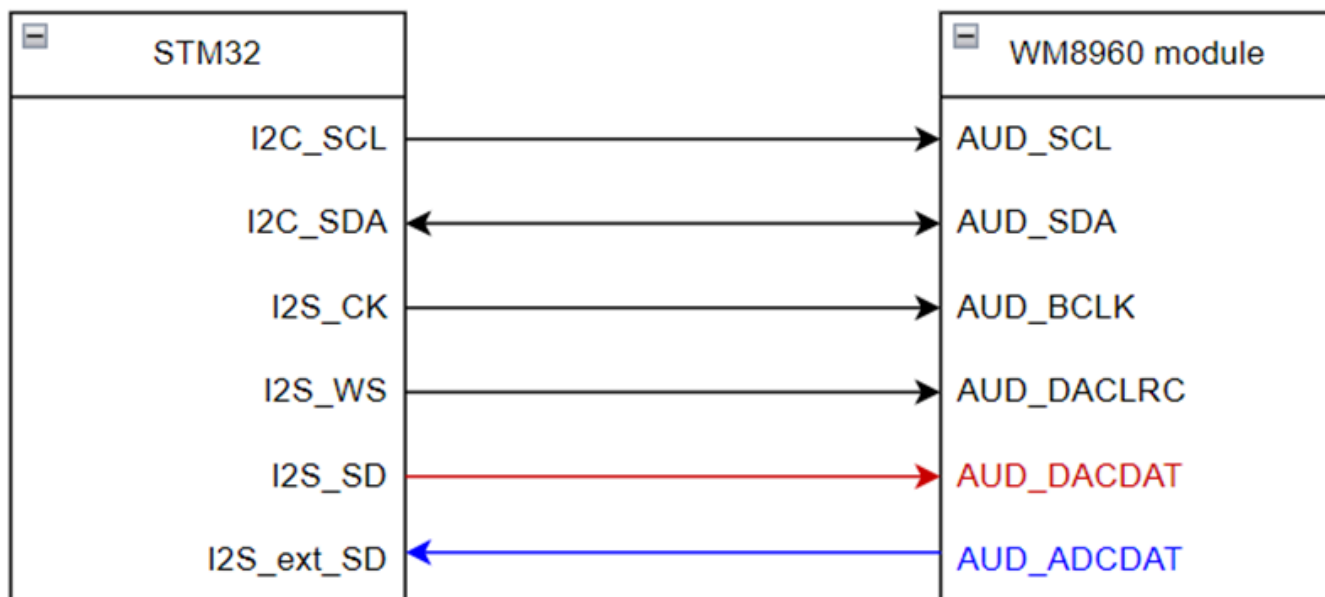
Search Signals

Search (Ctrl+F) Show only Modified Pins

Pin Name	Signal on Pin	GPIO output I..	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PB10	I2S2_CK	n/a	Alternate Fun...	No pull-up an...	Low		<input type="checkbox"/>
PB12	I2S2_WS	n/a	Alternate Fun...	No pull-up an...	Low		<input type="checkbox"/>
PC2	I2S2_ext_SD	n/a	Alternate Fun...	No pull-up an...	Low		<input type="checkbox"/>
PC3	I2S2_SD	n/a	Alternate Fun...	No pull-up an...	Low		<input type="checkbox"/>

- I2S_CK — это тактовый сигнал битовой частоты, соответствует BCLK модуля WM8960
- I2S_WS — это тактовый сигнал левого/правого канала, соответствует DACLRC модуля WM8960
- В режиме передачи I2S_SD — это вывод передачи данных, соответствует DACDAT модуля WM8960
- В режиме приема I2S_ext_SD — это вывод приема данных, соответствует ADCDAT модуля WM8960

В режиме передачи соединение между STM32 и WM8960 выглядит следующим образом:

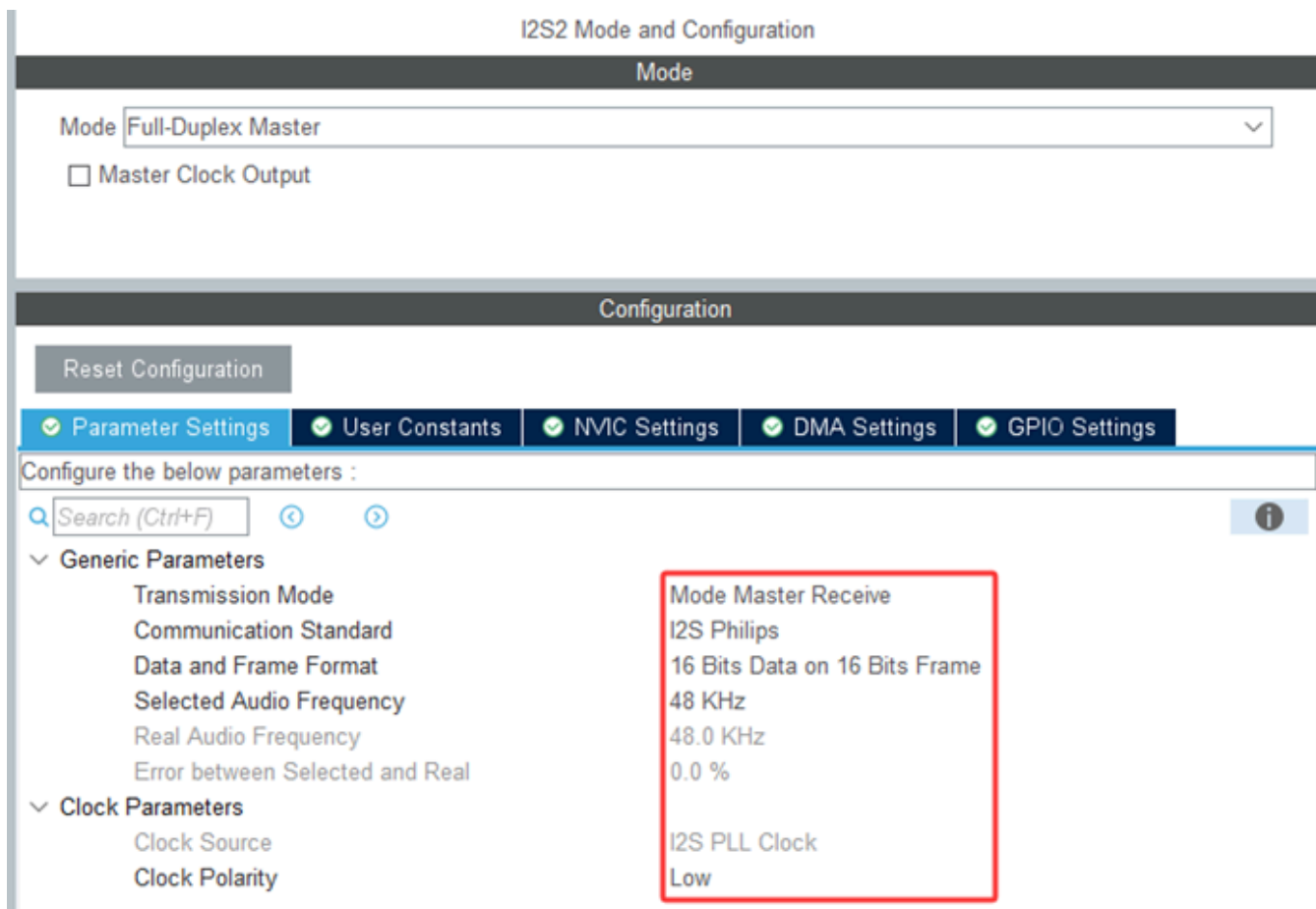


- Вывод MCLK модуля WM8960 можно оставить неподключенным
- ADCLRC модуля WM8960 не подключается
- После сброса вывод ADCLRC WM8960 по умолчанию является тактовым сигналом левого/правого канала АЦП. Однако I2S STM32 предоставляет только один сигнал левого/правого канала: сигнал I2S_WS, который используется совместно и для АЦП, и для ЦАП. Когда ADCLRC WM8960 сконфигурирован как GPIO, DACLR служит одновременно тактовым сигналом левого/правого канала и для ЦАП, и для АЦП. Следовательно, при использовании вывод ADCLRC WM8960 необходимо сконфигурировать как GPIO.

2.2. Режим мастера на прием

I2S STM32 действует как мастер и настроен в режиме приема.

Настройте интерфейс I2S, как показано ниже. Режим может быть выбран как полно-дуплексный или полу-дуплексный. Выбор полу-дуплексного режима позволяет только прием без передачи.

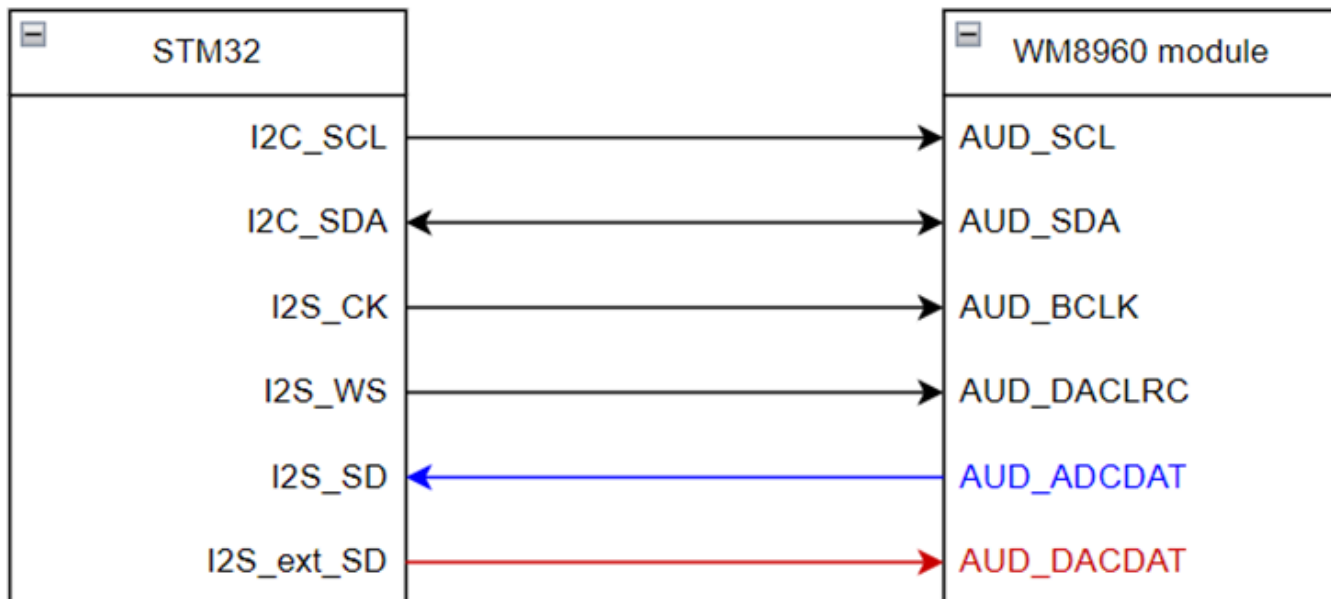


Конфигурация DMA и конфигурация GPIO согласуются с таковыми в режиме передачи.

В режиме приема функции выводов I2S_SD и I2S_ext_SD меняются:

- I2S_CK — это тактовый сигнал битовой частоты, соответствует BCLK модуля WM8960
- I2S_WS — это тактовый сигнал левого/правого канала, соответствует DACLRC модуля WM8960
- В режиме приема I2S_SD — это вывод приема данных, соответствует ADCDAT модуля WM8960
- В режиме приема I2S_ext_SD — это вывод передачи данных, соответствует DACDAT модуля WM8960

В режиме приема соединение между STM32 и WM8960 выглядит следующим образом:



- Вывод MCLK модуля WM8960 можно оставить неподключенным.
- Вывод ADCLRC модуля WM8960 не подключается.

2.3. Структура программного обеспечения

```

Project/
├── Core/
│   ├── Src/
│   │   ├── main.c           # Главная программа
│   │   ├── stm32f4xx_it.c   # Процедуры обслуживания
│   │   прерываний
│   │   ├── dma.c           # Конфигурация DMA
│   │   ├── i2c.c           # Конфигурация интерфейса I2C
│   │   ├── i2s.c           # Конфигурация интерфейса I2S
│   │   ├── wm8960.c        # Драйвер WM8960
│   │   ├── audio_tx.c      # Воспроизведение аудио путем
│   │   включения DMA передачи I2S
│   │   ├── audio_tx_rx.c   # Петля аудио путем включения DMA
│   │   передачи и приема I2S
│   │   ├── freertos.c      # Аудиозадача
│   │   └── Inc/            # Директория заголовочных файлов
├── Drivers/
│   ├── CMSIS/              # Пакет поддержки ядра Cortex-M4
│   ├── STM32F4xx_HAL_Driver/ # HAL-библиотека STM32F4
└── Middlewares/           # Промежуточное ПО (опционально)
  
```


2.4. Настройка среды разработки

2.4.1. Необходимое программное обеспечение

- IAR Embedded Workbench for ARM версии 8.x или выше
- HAL-библиотека STM32CubeF4 (включена в проект)

2.4.2. Настройка проекта

1. Откройте IAR Embedded Workbench
2. Выберите `File → Open workspace`
3. Перейдите к файлу `EWARM/Project.eww`
4. Подтвердите следующие конфигурации:
 - Устройство: STM32F429ZITx
 - Отладчик: ST-LINK (или другой соответствующий отладчик)

2.5. Инструкции по использованию

2.5.1. Аудиодрайвер

```
// инициализировать чип WM8960: выбрать LineIN или MIC в качестве  
источника входного сигнала  
void init_wm8960(ewm8960InputSrc src);  
  
// динамически переключить источник входного сигнала  
void set_wm8960_input_src(ewm8960InputSrc src);  
  
// установить громкость LineOut  
void set_hp_vol(int32_t vol_db);  
  
// установить громкость динамика  
void set_spk_vol(int32_t vol_db);
```

```
// Установить состояние выхода, когда вывод ADCLRC сконфигурирован как GPIO
void set_adclrc_gpio(uint8_t state);
```

2.5.2. Режим петли (Loopback)

Используйте функцию HAL_I2SEx_TransmitReceive_DMA() для одновременной передачи и приема аудио. По завершении вызывается callback-функция HAL_I2SEx_TxRxCpltCallback().

```
// Предоставить два буфера данных в качестве буферов приема и передачи, переключать буферы в callback-функции завершения передачи/приема
void HAL_I2SEx_TxRxCpltCallback(I2S_HandleTypeDef *hi2s)
{
    if (stop_dma != 0)
    {
        tx_cplt = 1;
        return;
    }
    if (ping_pong == 0)
    {
        HAL_I2SEx_TransmitReceive_DMA(hi2s, pong_buf, ping_buf,
BufSize);
        ping_pong = 1;
    }
    else if (ping_pong == 1)
    {
        HAL_I2SEx_TransmitReceive_DMA(hi2s, ping_buf, pong_buf,
BufSize);
        ping_pong = 0;
    }
}
```

2.5.3. Режим передачи

Используйте функцию HAL_I2S_Transmit_DMA() для передачи аудиоданных. По завершении передачи вызывается функция HAL_I2S_TxCpltCallback().

Непрерывный вызов HAL_I2S_Transmit_DMA() внутри HAL_I2S_TxCpltCallback() позволяет осуществлять непрерывную передачу аудиоданных.

```
// непрерывно отправлять новые аудиоданные в callback-функции
// завершения передачи, пока все аудиоданные не будут отправлены
void HAL_I2S_TxCpltCallback(I2S_HandleTypeDef *hi2s)
{
    if (stop_dma != 0)
    {
        tx_cplt = 1;
        return;
    }

    if (audio_rem_size > 0)
    {
        uint32_t dma_len = (uint32_t)(DMA_MAX(audio_rem_size));

        /* Обновить слой media и включить его для воспроизведения */
        HAL_I2S_Transmit_DMA(hi2s, wav_data_new_pos, dma_len);

        /* Обновить оставшееся количество данных для
        воспроизведения */
        audio_rem_size = audio_rem_size - dma_len;

        /* Обновить текущую позицию указателя аудио */
        wav_data_new_pos = wav_data_new_pos + dma_len;
    }
    else
    {
        tx_cplt = 1;
    }
}
```

2.5.4. Режим приема

Используйте функцию HAL_I2S_Receive_DMA() для приема аудиоданных. По завершении приема вызывается функция HAL_I2S_RxCpltCallback().

Непрерывный вызов HAL_I2S_Receive_DMA() внутри HAL_I2S_RxCpltCallback() позволяет осуществлять непрерывный прием аудиоданных.

2.5.5. Переключение источника аудиовхода

Используйте PC13 (Кнопка) для повторяющегося переключения источника входного сигнала, при этом соответствующий светодиод на PB7 указывает текущее состояние.

```
if (button_pressed)
{
    osDelay(1000);
    button_pressed = 0;
    stop_dma = 1;

    // ждать завершения передачи по DMA
    while(tx_cplt == 0)
    {
        osDelay(1);
    }

    stop_dma = 0;
    osDelay(100);

    if (audio_src == WM8960_INPUT_LINEIN)
    {
        // Переключить на вход микрофона
        set_wm8960_input_src(WM8960_INPUT_MIC);
    }
    else
    {
        // Переключить на линейный вход
        set_wm8960_input_src(WM8960_INPUT_LINEIN);
    }
}
```

```

osDelay(100);

// test_audio_tx();
test_audio_tx_rx();

HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
count = 0;
}

```

3. Использование модуля на Raspberry Pi

3.1. Аппаратное подключение

Подключите плату разработки к WM8960 с помощью перемычек, включая линии питания, линии заземления, I2C SDA, SCL, I2S BCLK, DCLK, DAC и ADC. Обратитесь к следующему изображению для подключения выводов:

Raspberry Pi2 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	■	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	●	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	●	Ground	06
07	GPIO04 (GPIO_GCLK)	●	(TXD0) GPIO14	08
09	Ground	●	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	●	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	●	Ground	14
15	GPIO22 (GPIO_GEN3)	●	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	●	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	●	Ground	20
21	GPIO09 (SPI_MISO)	●	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	●	(SPI_CE0_N) GPIO08	24
25	Ground	●	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	●	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	●	Ground	30
31	GPIO06	●	GPIO12	32
33	GPIO13	●	Ground	34
35	GPIO19	●	GPIO16	36
37	GPIO26	●	GPIO20	38
39	Ground	●	GPIO21	40

Выводы Raspi	Выводы WM8960	Описание
5V	VIN	Вход питания
GND	GND	Земля
GPIO2	SDA	Вход данных I2C
GPIO3	SCL	Вход тактового сигнала I2C
GPIO18	BCLK	Вход тактового сигнала битов I2S
GPIO19	DACLRC	Тактовый сигнал левого/правого канала I2S
GPIO20	ADCDAT	Вход последовательных данных I2S
GPIO21	DACDAT	Выход последовательных данных I2S

3.2. Обновление источников программного обеспечения

```
sudo apt-get -y update  
sudo apt-get -y upgrade
```

3.3. Загрузка драйвера WM8960

Два метода: один — онлайн с использованием git clone; другой — офлайн путем загрузки сжатого файла, его извлечения и загрузки на Raspberry Pi.

```
# Метод 1: Использование git clone
git clone https://github.com/minstru/raspi-wm8960

# Метод 2: Загрузите исходный код с GitHub с помощью браузера,
извлеките его, затем загрузите (через SSH) на Raspberry Pi
# Предположим, audio_wm8960 загружен в папку ~/Downloads
# Измените соответствующие права доступа к файлам
cd raspi_wm8960
chmod +x install.sh
chmod +x uninstall.sh
chmod +x wm8960-soundcard
```

3.4. Установка драйвера

```
cd ~/Downloads/raspi_wm8960
sudo ./install.sh

sudo reboot
```

3.5. Проверка успешности установки драйвера

```
vtino@raspberrypi:~ $ sudo dkms status
wm8960-soundcard/1.0, 6.12.25+rpt-rpi-v7, armv7l: installed
```

3.6. Проверка устройств звуковой карты

```
vtino@raspberrypi:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: vc4hdmi [vc4-hdmi], device 0: MAI PCM i2s-hifi-0 [MAI PCM
i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: wm8960soundcard [wm8960-soundcard], device 0: bcm2835-i2s-
wm8960-hifi wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

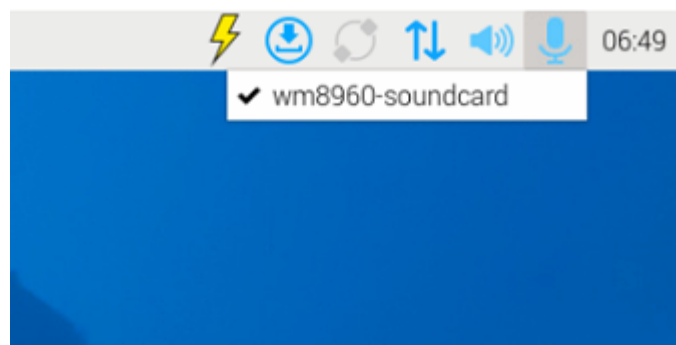
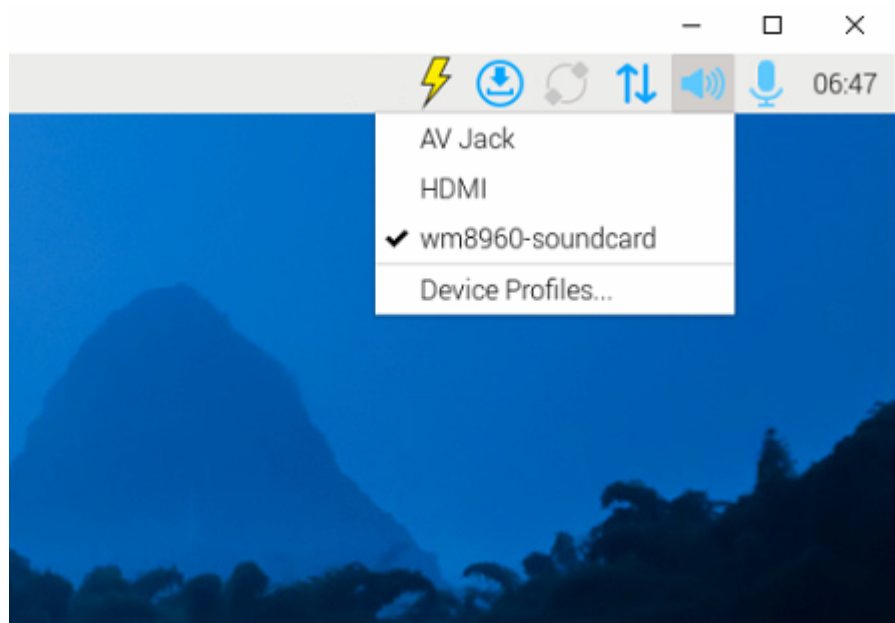
```
card 2: Headphones [bcm2835 Headphones], device 0: bcm2835
Headphones [bcm2835 Headphones]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7

vtino@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: wm8960soundcard [wm8960-soundcard], device 0: bcm2835-i2s-
wm8960-hifi wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3.7. Регулировка громкости

По умолчанию громкость относительно низкая, поэтому необходима регулировка.

Щелкните правой кнопкой мыши на значке динамика и выберите `wm8960-soundcard` из всплывающих опций; щелкните правой кнопкой мыши на значке записи и выберите `wm8960-soundcard` из всплывающих опций.

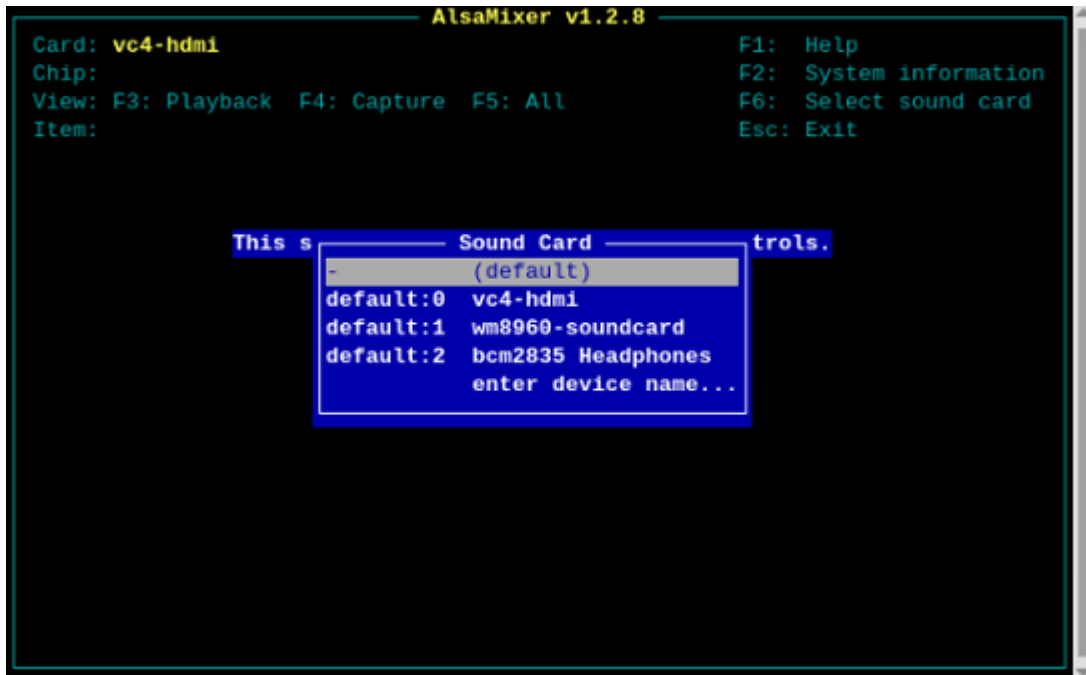


Введите следующую команду и нажмите **F6** во всплывающем интерфейсе, чтобы выбрать устройство звуковой карты.

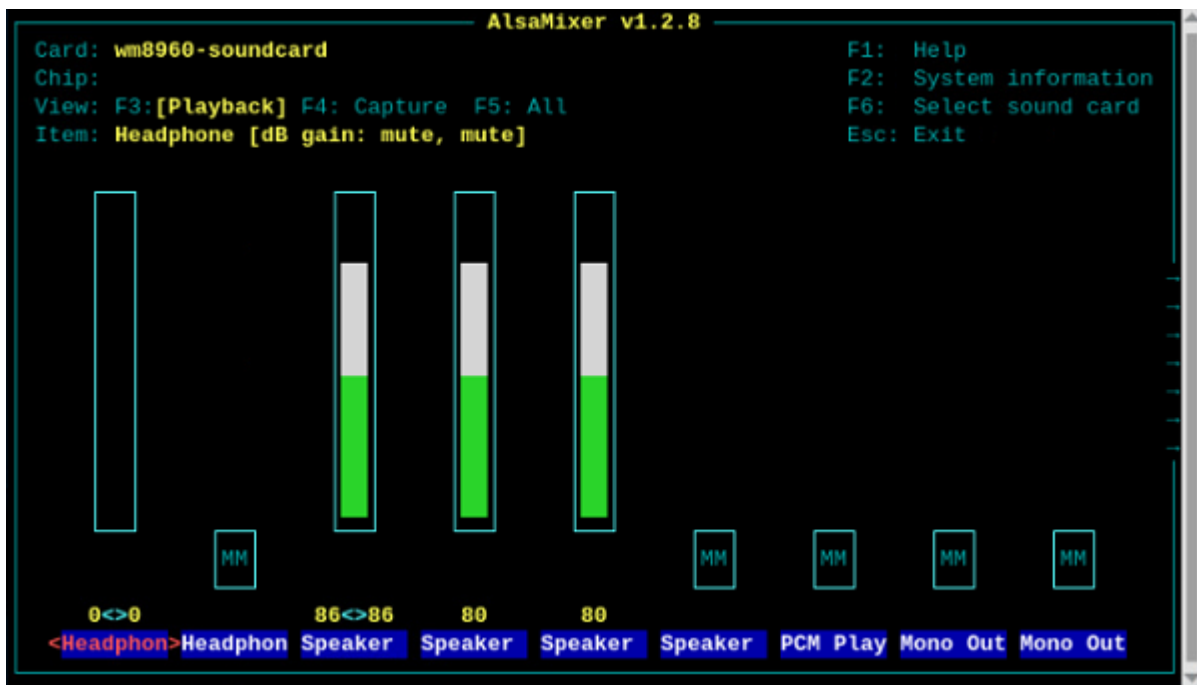
```
sudo alsamixer
```

```
AlsaMixer v1.2.8
Card: vc4-hdmi
Chip:
View: F3: Playback F4: Capture F5: All
Item:
F1: Help
F2: System information
F6: Select sound card
Esc: Exit

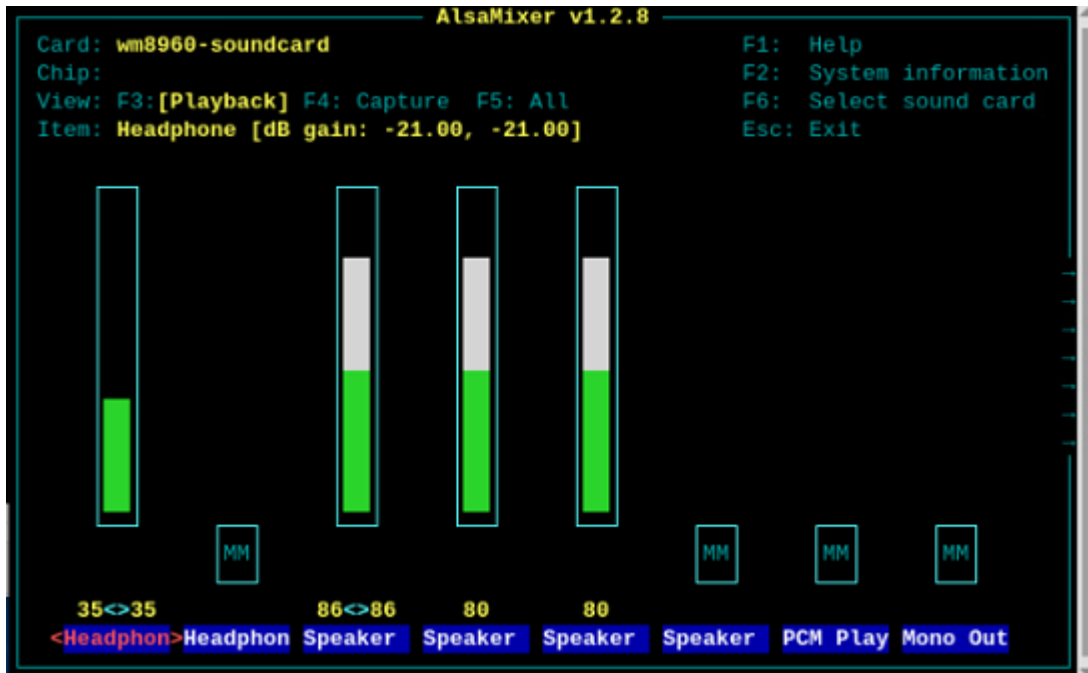
This sound device does not have any controls.
```



Используйте клавиши со стрелками, чтобы выбрать `wm8960-soundcard`, и нажмите Enter. Вы можете видеть, что громкость наушников в настоящее время равна 0.



Вы можете увеличить громкость:



3.8. Установка MP3-плеера: mpg123

```
sudo apt-get install mpg123
```

3.9. Тест воспроизведения аудио

Загрузите тестовый MP3-файл в папку Downloads, подключите наушники к аудиовыходному интерфейсу модуля WM8960. Воспроизведите MP3 с помощью mpg123.

```
vtino@raspberrypi:~/Downloads $ ls
test.mp3  WM8960-Audio-HAT-master
vtino@raspberrypi:~/Downloads $ mpg123 test.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and
3
      version 1.31.2; written and copyright by Michael Hipp and
others
      free software (LGPL) without any warranty but with best
wishes

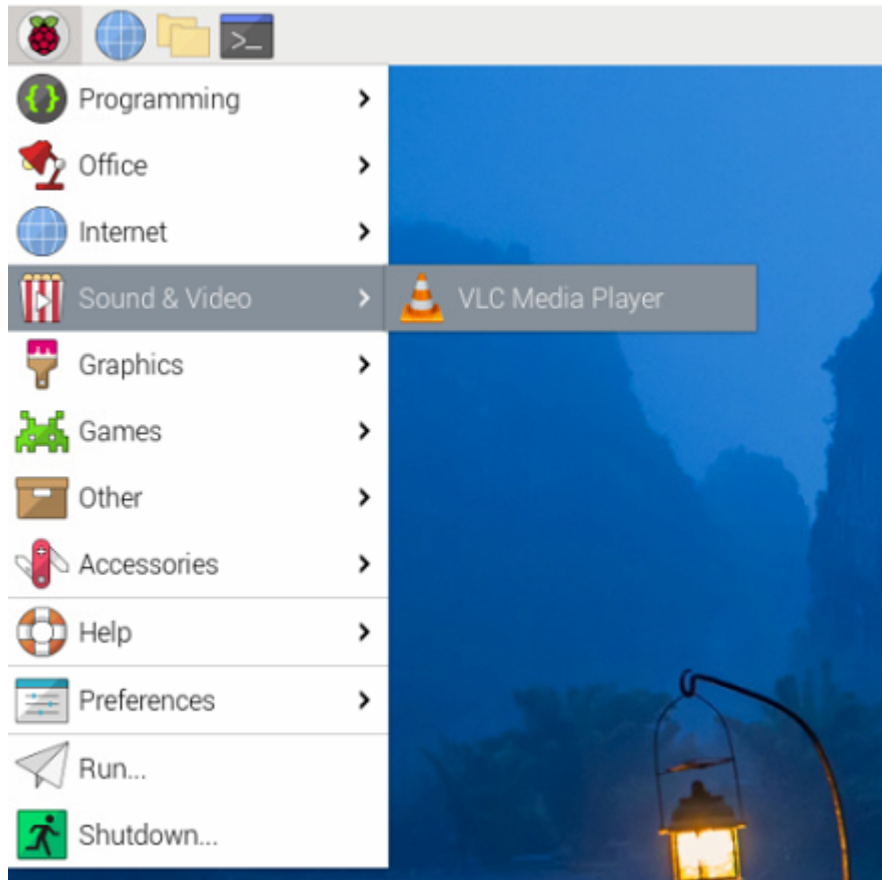
Terminal control enabled, press 'h' for listing of keys and
functions.
```

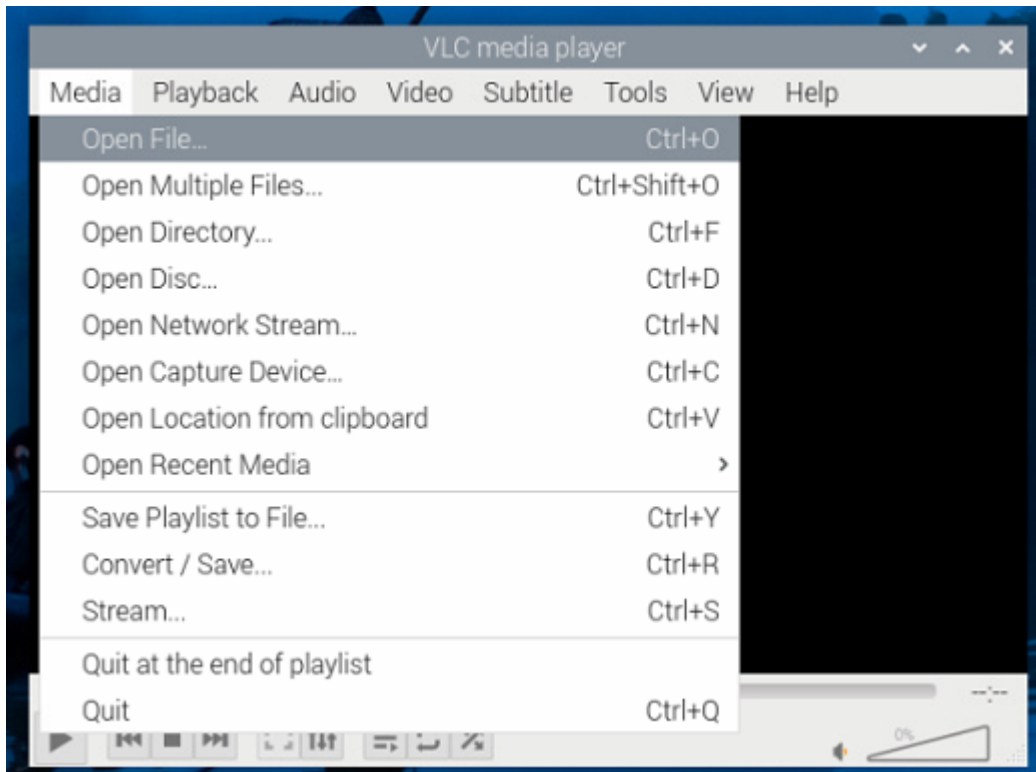
```
Playing MPEG stream 1 of 1: test.mp3 ...
```

```
MPEG 1.0 L III cbr128 44100 stereo
```

```
[1:49] Decoding of test.mp3 finished.
```

Если ваша установленная система имеет графический интерфейс, вы также можете воспроизводить MP3 с помощью VLC.





Откройте соответствующий MP3-файл через Медиа->Открыть файл... чтобы воспроизвести его.

3.10. Тест записи: Использование команды arecord

- Параметры 1,0 после -Dhw соответствуют звуковой карте 1 и устройству 0
- **CTRL+C** остановит запись

```
vtino@raspberrypi:~/Downloads $ ls
test.mp3 test.wav wm8960-Audio-HAT-master
vtino@raspberrypi:~/Downloads $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: wm8960soundcard [wm8960-soundcard], device 0: bcm2835-i2s-
wm8960-hifi wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
vtino@raspberrypi:~/Downloads $ arecord -Dhw:1,0 -f cd test.wav
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate
44100 Hz, Stereo
Aborted by signal Interrupt...
```

```
vtino@raspberrypi:~/Downloads $ ls
test.mp3 test.wav WM8960-Audio-HAT-master
vtino@raspberrypi:~/Downloads $
```

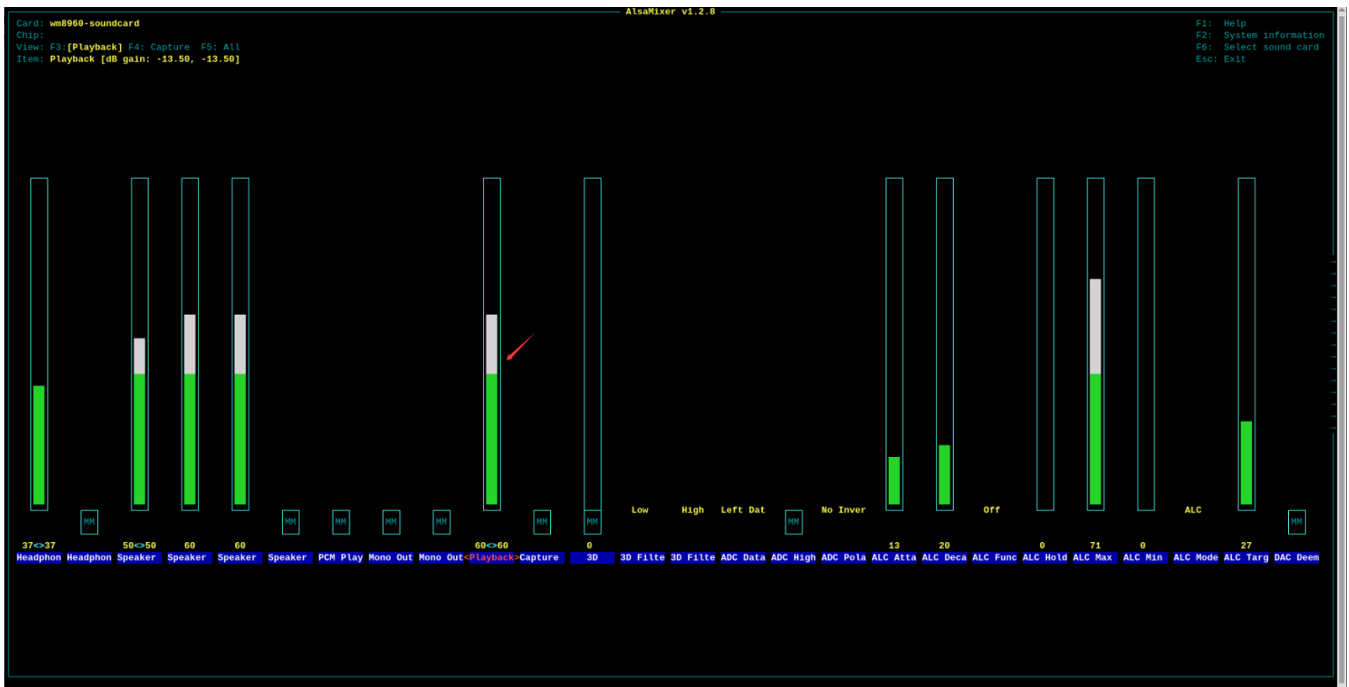
В этот момент в папке `Downloads` будет создан файл `test.wav`. Вы можете воспроизвести этот `wav`-файл с помощью `aplay`.

```
vtino@raspberrypi:~/Downloads $ aplay test.wav
Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100
Hz, Stereo
```

3.11. Сквозное воспроизведение

```
arecord -f cd -Dhw:1,0 | aplay -Dhw:1,0
```

Вам нужно немного уменьшить громкость воспроизведения.



3.12. Удаление драйвера

```
sudo ./unsintall.sh
```

4. Распространенные проблемы

4.1. Не удается обнаружить устройство WM8960

- Проверьте соединения линий I2C
- Используйте метод `set_adclrc_gpio()` для управления светодиодом, подключенным к ADLRC, чтобы проверить успешность операций I2C

4.2. Аудиоданные содержат шум

- Убедитесь, что конфигурации тактовых сигналов MCLK, BCLK и LRCLK верны