

# Hugo Vorlesegerät

## *Bedienungsanleitung und Service Handbuch*

### Inhaltsverzeichnis

Bedienungsanleitung.....	2
Inbetriebnahme.....	2
In Ruhezustand versetzen.....	2
Starten nach Ruhezustand.....	2
Bedienung der Vorlesefunktion.....	2
Vorlesen.....	2
Probleme und Behebung.....	3
Funktionsweise.....	4
Service Handbuch.....	5
Technische Daten.....	5
Öffnen.....	5
Spannungen und Basisfunktion (HW) prüfen.....	5
Funktionsbeschreibung.....	6
Komponenten.....	6
System prüfen.....	6
Option 1 per Monitor, Keyboard, Maus.....	7
Option 2 per SSH.....	7
Login.....	7
Checkliste.....	7

# Bedienungsanleitung

## Inbetriebnahme

Zur Inbetriebnahme bzw. immer nach dem das Gerät vom Stromnetz getrennt war.

- Die Tastatur anschließen (kontrollieren) und den Netzstecker einstecken.
- Nach kurzer Zeit muss die rote LED an der Vorderseite kurz flackern und dann dauerhaft leuchten.
- Der Rechner startet dann und meldet sich innerhalb ca. einer Minute mit dem Begrüßungstext

## In Ruhezustand versetzen

**Das Gerät sollte möglichst nur dann in den Ruhezustand versetzt werden wenn gerade keine Sprachausgabe erfolgt.**

Um das Gerät nach der Benutzung in den Ruhezustand zu versetzen muss die rote Taste an der Vorderseite gedrückt werden. Die LED beginnt langsam zu blinken, nach einigen Sekunden wird der Rechner abgeschaltet und die LED erlischt.

## Starten nach Ruhezustand

Mit der Taste kann das Geräte auch aus dem Ruhezustand aufgeweckt werden. Dazu die Taste kurz drücken, das Gerät schaltet sich ein und startet. Nach ca. einer Minute ertönt der Begrüßungstext.

## Bedienung der Vorlesefunktion

Das Blatt mit dem vorzulesenden Text muss möglichst gerade auf der Glasplatte angelegt werden. Dabei die obere Kante der Seite an der vorderen Kante des Scanners anlegen. Den Hinweisen der gesprochenen Texte folgen. Die Reaktion auf einen Tastendruck erfolgt teilweise erst nach einigen Sekunden.

## Vorlesen

Nach dem Auflegen des Blattes die runde Taste drücken, es folgt der Hinweis etwas zu warten und der Scanner beginnt hörbar zu arbeiten. Der erkannte Text wird dann vorgelesen.

Mit der eckigen Taste kann das Vorlesen des Textes abgebrochen werden.

Wurde der Text vollständig vorgelesen, kommt die Frage ob der Text wiederholt vorgelesen werden soll, dazu die quadratische Taste drücken. Sonst einfach die runde Taste drücken um einen neuen Text einzulesen oder das Gerät ausschalten zu können.

Mit den dreieckigen Tasten kann die Lautstärke verändert werden. Die Taste mit der Spitze nach oben erhöht die Lautstärke, die Andere reduziert sie.

## Probleme und Behebung

1. Text wird sehr undeutlich mit vielen Fehlern vorgelesen:
  1. Das Blatt liegt schief auf dem Scanner.
  2. Der Text ist undeutlich gedruckt oder zu kontrastarm. Z.Bsp. Texte ist grau hinterlegt.
  3. Zu dünnes Papier das beidseitig bedruckt ist kann ebenfalls zu Problemen führen.
  4. Der Text ist n einer ungünstigen Schriftart verfasst. Z.Bsp. Schreibschrift, alte Schriftarten.
2. Der Text wird trotz guter Vorlage komplett falsch gelesen:
  1. Das Blatt ist im Scanner falsch herum aufgelegt
3. Bilder im Textfluss
  1. Bilder im Textfluss stören normalerweise nicht. Unter Umständen werden im Bild erkannte Texte oder Bildunterschriften mitten im Text gesprochen.
  2. Durch Abdecken der Bilder kann das verhindert werden, falls es zu störend ist.
4. Mehrspaltige Texte
  1. Es darf immer nur eine Spalte eingescannt werden da die Software keine Spalten erkennt und alle Spalten als eine Zeile vorliest.
  2. Mit weißem (Kopier-) Papier den Rest abdecken, Für oft verwendete Formate (Zeitungen) kann man sich Schablonen schneiden und damit alles abdecken was nicht gelesen werden soll.

## **Funktionsweise**

Das Blatt wird als Bild eingescannt. Der Rechner versucht dann die in der grafischen Information enthaltenen Texte zu erkennen, extrahieren und aufzubereiten.

Dabei wird auf deutsche Wörterbücher zurückgegriffen, ein Teil der Fehler beim Einlesen der Seite wird dabei korrigiert. Fremdsprachige Begriffe werden buchstabengetreu übernommen, Lesefehler in solchen Fremdwörtern werden somit auch nicht korrigiert.

Der erkannte Text wird dann von einer Text-zu-Sprache Software ausgegeben.

Im Inneren werkelt ein Raspberry 4 mit einem Linux Betriebssystem.

# Service Handbuch

## Technische Daten

Stromversorgung	240V
Leistungsaufnahme	ca. 15W im Betrieb
Schutzklasse	1
Systemhardware	Raspberry 4
Betriebssystem	Linux Raspi OS
Power	5V Meanwell Netzteil, Arduino Micro schaltet Raspberry



## Software

Scannen	Sane
Texterkennung	Tesseract
Text-to-Speech	Speak NG Text-to-Speech <a href="https://github.com/espeak-ng/espeak-ng">https://github.com/espeak-ng/espeak-ng</a>
Bedienung	In Python und Linux bash erstelltes Paket readspeak.py steuert alles

## Öffnen

Zum öffnen des Gerätes, dieses zuerst vom Netz trennen.

Dann USB Stecker von Scanner abziehen. Den Scanner nach hinten schieben und entnehmen.

Gehäuseschrauben entfernen.

Auf der Power Control-Platine die Stecker zur LED und dem Taster abziehen. Den Stecker des Lautsprecherkabels am Audio-Modul auf dem Raspberry abziehen.

## Spannungen und Basisfunktion (HW) prüfen

**(Achtung! Am Netzteil sind berührbare Teile mit der Netzspannung vorhanden. Trenntrafo verwenden)**

Am Ausgang Netzteil:	+5V	
Arduino Micro	+5V und 3.3V, blaue LED leuchtet (Pins siehe Abbildung 1)	3,3V werden Arduino intern erzeugt
Relaismodul IN	+5V	
Schraubanschluss Mitte		



# System prüfen

(Linux Kenntnisse erforderlich)

Nach einschalten des Relais muss der Raspberry booten, das ist am flackern der grünen LED auf dem Bord erkennbar.

Gleichmäßiges Blinken im Sekundenabstand deutet darauf hin, dass das Bootmedium nicht gefunden wird. (SD-Card nicht lesbar)

Zur weiteren Prüfung gibt es zwei Optionen :

## Option 1 per Monitor, Keyboard, Maus

HDMI-Monitor Monitor und Tastatur anschließen. Boot Prozess, Fehlermeldungen und laufendes System analysieren.

## Option 2 per SSH

Raspberry per LAN-Kabel mit lokalem Netzwerk verbinden und per SSH einloggen.

## Login

User: service

Passwort: service

## Checkliste

### 1. Bootet das System komplett?

2. Auf Console mit **PS AF | grep python** prüfen ob das Python-Skript **readSpeak.py** gestartet wurde. Dieser sollte durch ein systemd module nach dem Boot-Prozess gestartet werden.

3. **Audio-Module testen.** (Siehe Anlage „WM8960 Audio HAT User Manual“)

### 4. Scanner testen

Der Scanner wird wie unter Linux üblich per SANE eingebunden. Der Check ist mit folgenden Befehlen möglich:

***sudo sane-find-scanner -q***

Der Canon Scanner sollte gefunden werden und in der Ausgabe des Befehls zu finden sein.

***scanimage -L***

Zeigt weitere Details über den Scanner, sofern der Scanner von SANE gefunden **und** erkannt wird.

***scanimage >test.jpg --format jpeg --resolution=300 -p***

Der Scanner sollte nun arbeiten und ein Datei test.jpg generieren. Mit

***file test.jpg***

kann dies geprüft werden. Es sollte eine Ausgabe die „JPEG image data...“ enthält erfolgen.

Ob der Scanner ein korrektes Bild erzeugt hat kann durch das Öffnen der Datei geprüft werden. (Evtl. per Kopie auf einen anderes Gerät)

**5. Spezielle Eingabetastatur testen.** In der Console sollten folgende Zeichen nach Drücken der Tasten erscheinen:

Runde Taste	-
Quadratische Taste	+
Pfeil nach unten (Leiser)	PgDown
Pfeil nach oben (Lauter)	PgUp



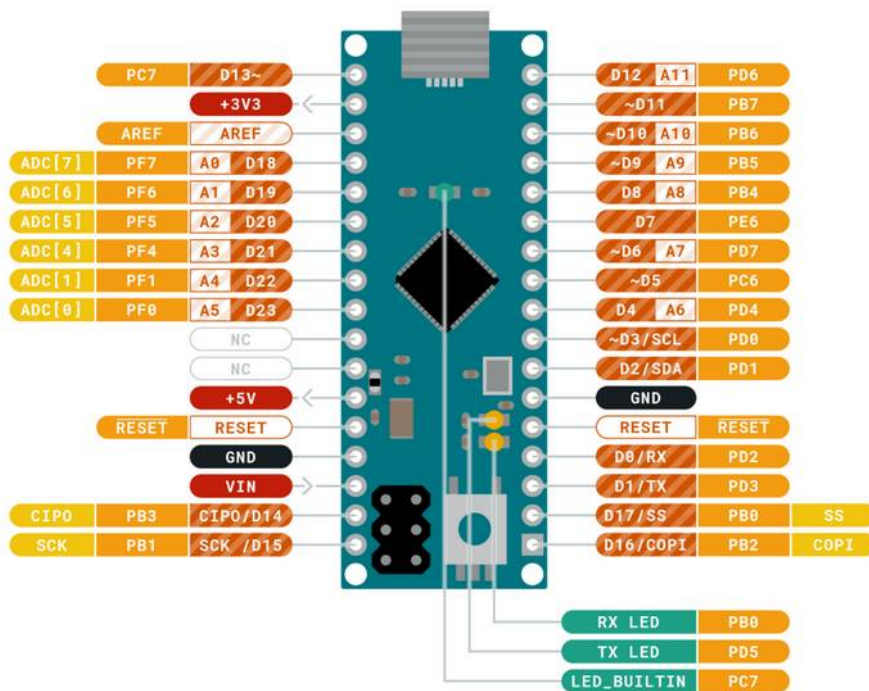


Abbildung 1: Arduino Pinout

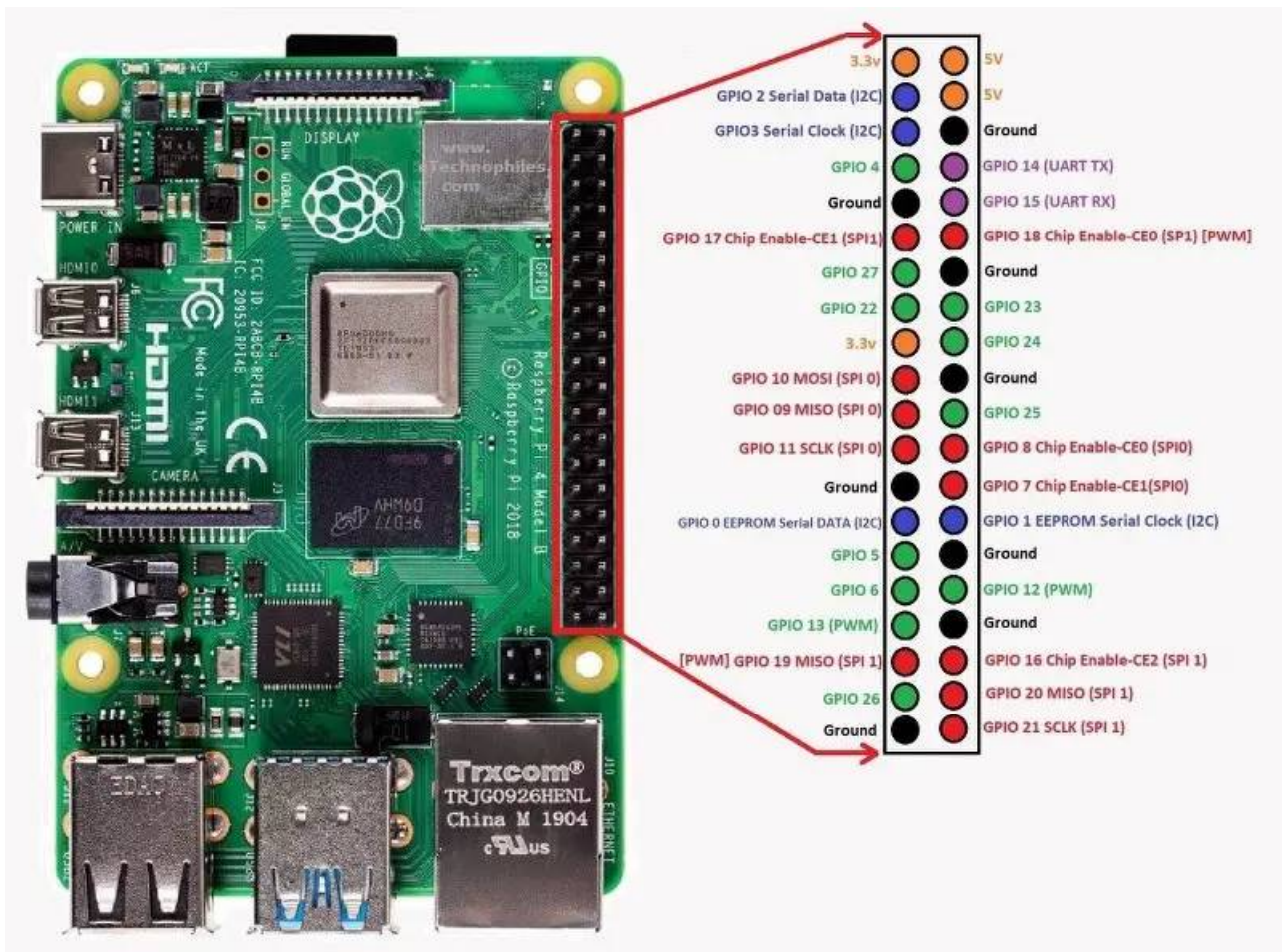


Abbildung 2: Raspberry Pinout

## Single Relay Module 5V(ME010)



### Introduction

The module is used to control high-voltage electrical device. It can be used in interactive projects and can also be used to control the lighting, electrical and other equipment. It can be controlled directly by a wide range of microcontrollers and can be controlled through the digital IO port, such as solenoid valves, lamps, motors and other high current or high voltage devices.

### Specifications:

- Equipped with high-current relay, AC250V 3A ; DC30V 3A
- Standard interface that can be controlled directly by microcontroller ,such as Arduino , 8051, AVR, PIC, DSP, ARM and so on
- Contact independent wiring, safe and reliable
- Platform: the Arduino/ single chip microcomputer
- Port: digital quantity level
- Contact: 1 normally open contact + 1 normally closed contact
- Contact current: < 3 A
- Voltage: 5V
- Weight: 12g
- Size(L\*W\*H): Approx. 41\*26\*15mm/ 1.61\*1.02\*0.59inch

## Pinout

Pin Name	Description
“+”	Power(5V DC)
“-”	GND
“S”	Signal pin, connected with Arduino
“NO”	Normally open terminal
“NC”	Normally closed terminal
“C”(middle pin)	Common terminal, Which connected with the power for the load.

### 3. Example

\*\*\*\*\*Code begin\*\*\*\*\*

```
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage
level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the
voltage LOW
  delay(1000);              // wait for a second
}
```

\*\*\*\*\*Code End\*\*\*\*\*



# WM8960 Audio HAT

## User Manual

### OVERVIEW

This is a sound card HAT designed for Raspberry Pi, low power consumption, supports stereo encoding / decoding, features Hi-Fi playing / recording, what's more, it can directly drive speakers to play music.

### FEATURES

- Raspberry Pi connectivity, compatible with Raspberry Pi Zero/Zero W/Zero WH/2B/3B/3B+
- Integrates WM8960 low power stereo CODEC, communicates via I2S interface
- Integrates dual high-quality MEMS silicon Mic, supports left & right double channels recording, nice sound quality
- Onboard standard 3.5mm earphone jack, play music via external earphone
- Onboard dual-channel speaker interface, directly drives speakers
- Supports sound effects such as stereo, 3D surrounding, etc.
- Comes with development resources and manual (python demo code for playing / recording)

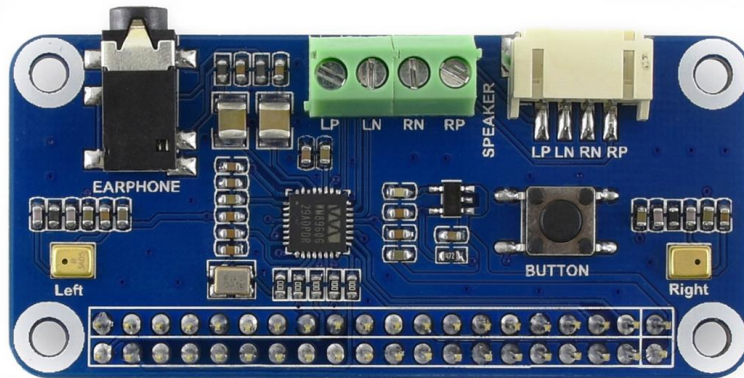
## SPECIFICATIONS

- CODEC: WM8960
- Power supply: 5V
- Logic voltage: 3.3V
- Control interface: I2C
- Audio interface: I2S
- DAC signal-noise ratio: 98dB
- ADC signal-noise ratio: 94dB
- Earphone driver: 40mW (16Ω@3.3V)
- Speaker driver: 1W per channel (8Ω BTL)

## CONTENT

Overview.....	1
Features.....	1
Specifications .....	2
Hardware.....	4
Use GUiDes .....	5
Install driver .....	5
Check Sound Card .....	5
Record and Play.....	6
Volume adjust.....	7
Set default sound card.....	7
Other play tools.....	8
Codse control playing .....	10
Install libraries .....	10

## HARDWARE



LP, LN are positive polar and negative polar of the left speaker separately; RP, RN are positive polar and negative polar of the right speaker.

Pinout:

PIN	RaspberryPi (BCM)	RaspberryPi (Board)	Description
5V	5V	5V	5V
GND	GND	GND	Ground
SDA	P2/SDA	3	I2C Data input
SCL	P2/SDA	5	I2C Clock input
CLK	P18	12	I2S Bit clock input
LRCLK	P19	35	I2S Frame clock input
DAC	P21	40	I2S Data output
ADC	P20	38	I2S Data input
BUTTON	P17	11	Configurable

## USE GUIDES

### INSTALL DRIVER

Insert the WM8960 Audio HAT to Raspberry Pi. Power on Raspberry Pi and networking.

Clone the driver and install it with commands:

```
git clone https://github.com/waveshare/WM8960-Audio-HAT
cd WM8960-Audio-HAT
sudo ./install.sh
sudo reboot
```

Waiting for rebooting, then check the driver

```
pi@raspberrypi:~/WM8960-Audio-HAT $ sudo dkms status
wm8960-soundcard, 1.0, 4.14.71+, armv6l: installed
wm8960-soundcard, 1.0, 4.14.71-v7+, armv6l: installed
```

If the response information doesn't include kernel version as below, you need to try to install it again.

```
pi@raspberrypi:~/WM8960-Audio-HAT $ sudo dkms status
wm8960-soundcard, 1.0: added
```

### CHECK SOUND CARD

Check sound card status of Raspberry Pi with command **aplay -l** and **arecord -l**

```
pi@raspberrypi:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 7/7
  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
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```



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

## RECORD AND PLAY

### Record and play Test

Use arecord to record audio and play: (earphone or speaker is required)

```
sudo arecord -f cd -Dhw:1 | aplay -Dhw:1
```

After running the command, you can hear the sound recorded by mic from earphone or speaker. Note that the speaker should away from the mic to void from noise.

### Record

```
sudo arecord -D hw:1,0 -f S32_LE -r 16000 -c 2 test.wav
```

This command is used to record sounds from device 1.0, two channels, 16000Hz, 32bit, and export as test.wav

```
-D : device
-r : frequency
-c : channel
-t : type
-f : format
```

### Play

```
sudo aplay -D HW:1,0 test.wav
```

Note that aplay tool can only play \*.wav audio file.

## VOLUME ADJUST

To setting sound and adjust volume, you can use alsamixer tool

```
sudo alsamixer
```

If WM8960 is not the default sound card, you should press F6 to choose audio device.



## SET DEFAULT SOUND CARD

```
sudo vi /usr/share/alsa/alsa.conf
```

Open the file and find the statement:

```
defaults.ctl.card 0  
defaults.pcm.card 0
```

Modify these statements from 0 to 1. If WM8690 is device 1, you can modify the

statements to the corresponding device number of WM8690.

## OTHER PLAY TOOLS

aplay can only used to play .wav audio files. If you need to play other audio like MP3, you can use mpg123 software.

Install mpg123

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

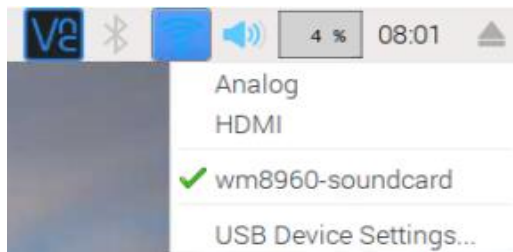
Play audio (Note: you should first set WM8960 as default sound card)

```
sudo mpg123 test.mp3
```

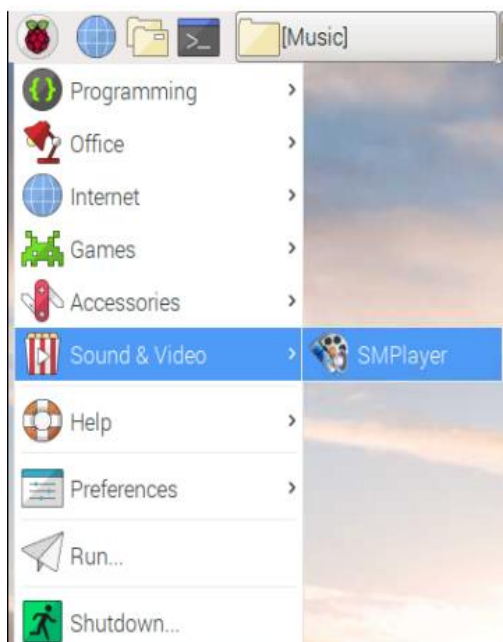
smplayer is a graphic audio playing software

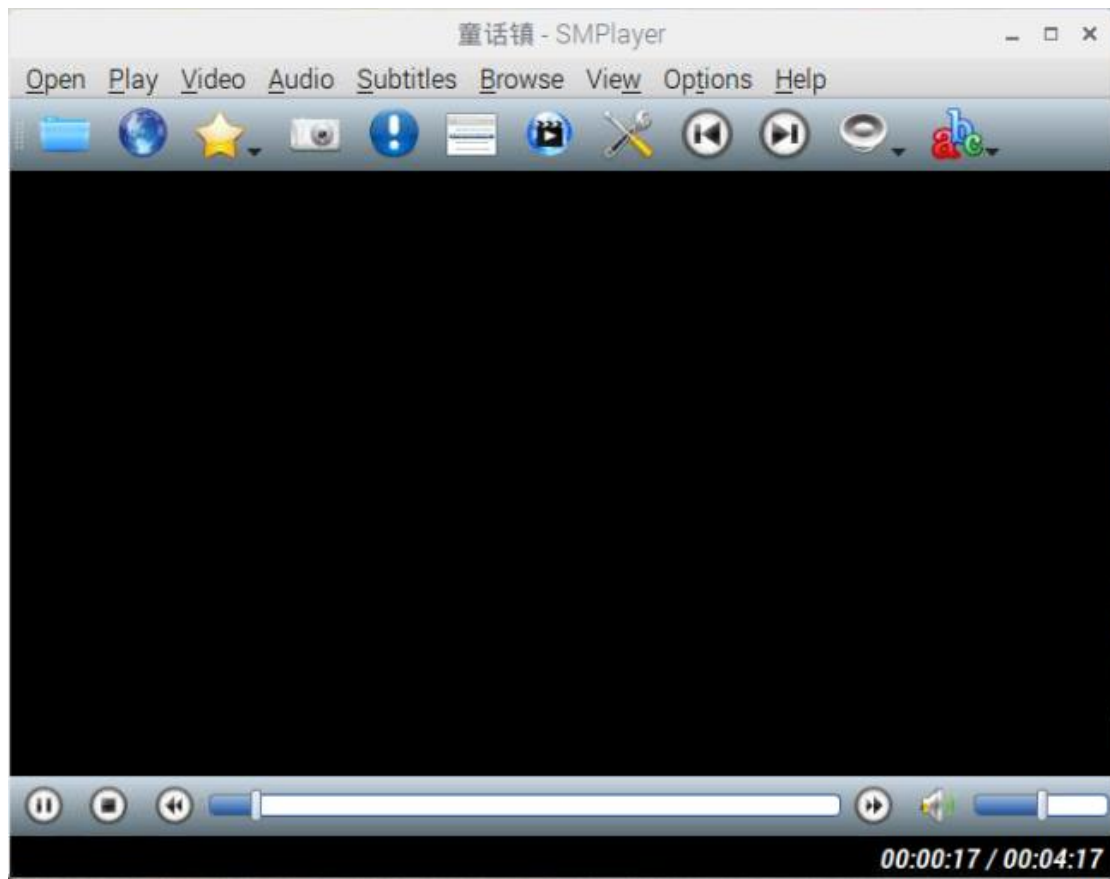
```
sudo apt-get install smplayer
```

Right click and set WM8960-soundcard as default device



Open smplayer software on menu, and open audio files to play.





## CODSE CONTROL PLAYING

### INSTALL LIBRARIES

Install libraries

```
sudo apt-get install libasound2-dev
```

Install pyalsaaudio

```
cd ~  
git clone https://github.com/larsimmsch/pyalsaaudio  
cd pyalsaaudio  
sudo python setup.py build  
sudo python setup.py install
```

Download demo codes from waveshare wiki and copy to raspberry pi

unzip demo codes to pi user directory.

Play:

```
sudo python playwav.py music.wav
```

```
pi@raspberrypi:~/WM8960_Audio_HAT_Code $ sudo python playwav.py music.wav  
2 channels, 44100 sampling rate
```

Record:

```
sudo python recordwav.py out.wav
```

```
pi@raspberrypi:~/WM8960_Audio_HAT_Code $ sudo python recordwav.py out.wav  
2 channels, 44100 sampling rate
```