

1.2.2 Voice Device Introduction and Testing

1. Device Overview

1.1.WonderEchoPro

1.1.1.Introduction

WonderEcho Pro, also known as the AI Voice Interaction Box, is equipped with a high-performance noise-canceling microphone and a high-fidelity speaker. It utilizes a USB-to-audio module, making it plug-and-play with no drivers required, and is compatible with multiple operating systems for both playback and recording.

Integrating various voice processing technologies, WonderEcho Pro features advanced noise suppression algorithms that effectively filter background noise from the environment. It supports a complete voice interaction process—from wake-up to recognition and response. With its modular design, each functional component (e.g., wake word detection, sound detection, speech recognition, and synthesis) can be developed and tested independently.

1.1.2.Features and Specifications

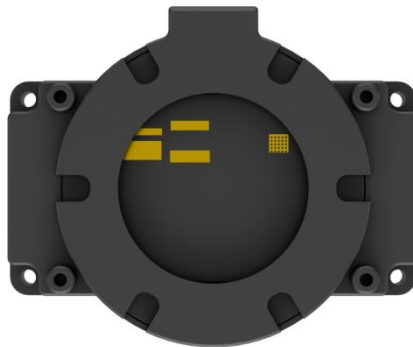
- ① Built-in microphone and speaker interface, supporting both audio input and output
- ② Driver-free, plug-and-play functionality compatible with Windows, macOS, Linux, Android, and more
- ③ Standard USB 2.0 interface

- ④ Control interface: USB
- ⑤ Voice chip model: CL1302
- ⑥ Speaker output: 3.0W per channel (4Ω BTL)
- ⑦ Power supply voltage: 5V

1.2.6-Microphone Circular Array

1.2.1.Introduction

The 6-Microphone Circular Array is a high-sensitivity, high signal-to-noise ratio microphone board. It features six analog silicon microphones arranged in a circular pattern. When paired with a main control board, it supports high-performance Acoustic Echo Cancellation (AEC), environmental noise reduction, and factory-level voice pickup from up to 10 meters.



1.2.2.Features and Specifications

Operating Voltage: 3.3V (typical)

Operating Current: 0.8mA (typical)

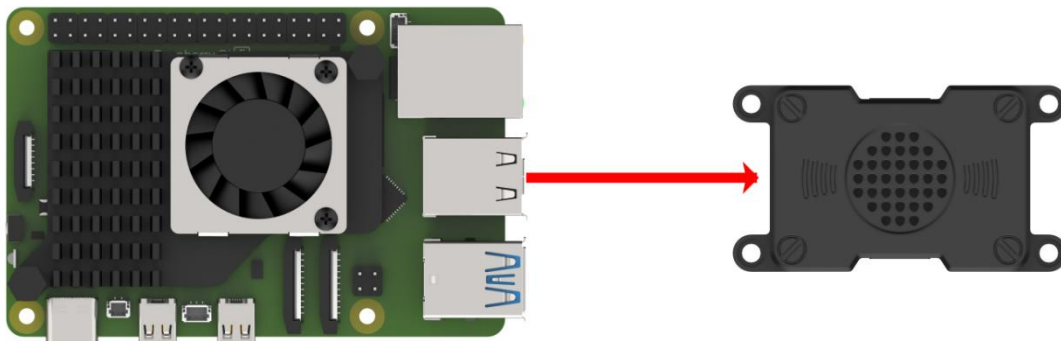
Operating Temperature: -20°C (min), 25°C (typical), 70°C (max)

Operating Humidity: Up to 95% relative humidity (max)

2. Recording and Playback Test

The following demonstration uses the Raspberry Pi 5 as an example. The connection and testing steps are also applicable to other compatible devices such as the Jetson series:

2.1 Connection Illustration and Detection



If the main controller is a Raspberry Pi, you can use VNC remote desktop access (refer to the appendix: Remote Access and File Transfer) to log into the Raspberry Pi system. Once connected, check the upper right corner of the desktop for microphone and speaker icons. As shown in the image below, the presence of these icons indicates a successful connection.



If you're using a NVIDIA Jetson device, you can connect via the NoMachine remote access tool. After logging in, check the upper right corner of the system interface for the speaker icon to confirm successful detection.



2.2 Recording Test

- 1) Next, open a new terminal window and enter the following command to check the available recording devices. Note that the -l option is a lowercase "L". You should see the card number (card) listed—for example, card 0. This is just an example; please refer to your actual query result.

```
arecord -l
```

```
> arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: Device [USB PnP Audio Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

- 2) Then, use the following command to start recording. Replace the red-marked card number (hw:0,0) with the actual number you found in the previous step:

```
arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test.wav
```

- 3) This will create a test.wav audio file in the current directory.

```
arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test.wav
```

- 4) You can record a short 5-second sample, then press Ctrl + C to stop the recording.

2.3 Playback Test

- 1) After the recording is complete, you can check whether the audio file was successfully created by listing the contents of the current

directory:

```
ls
```

```
> ls  
test.wav
```

- 2) If test.wav appears in the list, the recording was successful. To play back the recording, use the following command:

```
aplay test.wav
```

```
> aplay test.wav
```