Audio Playback

Introduction to the results of routine experiments

We have recorded audio with K230. Now let's try to play the audio.

Here our speakers are not integrated on K230, and need to be connected externally through the reserved audio output interface

The example code for this section is located in: [Source Code/10.Media/02.play_audio.py]

We use CanMV IDE to open the sample code and connect K230 to the computer via USB.

Click the Run button in the lower left corner of CanMV IDE, and K230 will start playing the audio file [/data/audio/audio.wav]

Code Explanation

[For the completed code and comments, please refer to the code file:]

Let's analyze the structure of this code.

1. Overall architecture

- It mainly contains two functions:
  - `exit_check`: Check exit conditions
  - `play_audio`: Core audio playback function
- Manage speaker hardware with YbSpeaker

1. Initialization of key components

```
spk = YbSpeaker()  # 扬声器实例 Speaker instance
wf = wave.open(filename, 'rb')  # WAV文件读取器 WAV file reader
p = PyAudio()  # 音频处理实例 Audio processing example
MediaManager.init()  # 媒体管理器 Media Manager
```

1. Audio parameter configuration

```
CHUNK = int(wf.get_framerate()/25)  # 计算chunk大小 Calculate chunk size
stream = p.open(
    format=p.get_format_from_width(wf.get_sampwidth()),
    channels=wf.get_channels(),
    rate=wf.get_framerate(),
    output=True,
    frames_per_buffer=CHUNK
)
```

1. Playback process control

```
data = wf.read_frames(CHUNK)
while data:
    stream.write(data)
    data = wf.read_frames(CHUNK)
    if exit_check():
        break
```

1. Resource management mechanism

```
try:
    # Audio playback logic
    # 音频播放逻辑
finally:
    stream.stop_stream()
    stream.close()
    p.terminate()
    wf.close()
    spk.disable()
    MediaManager.deinit()
```

1. Error handling

- Using the try-except-finally structure
- Special exit_check function handles user interruption
- Exception catching and logging