

測驗題目

- 芯片：Hi3518E V200
- 題目情境：

以 UDP 協議實做一個實時的雙向音頻通訊(上位機 \leftrightarrow 嵌入式)，第一步需先將碼流編碼封裝，發送後接收端收到封包後存入 buffer 解碼，這就是一個簡單的生產者與消費者例子。因嵌入式晶片計算能力有限，且為了提高通訊的效益，通常會將原始的 PCM 碼流封裝為 G726、G711、adpcm 等碼流格式，這邊我們使用 G711 碼流，海思編碼解碼均會在每幀標頭讀取 4bytes 它們自行設計的格式，下圖為海思語音幀介紹。

9.2.2.3 海思语音帧结构

使用海思语音编解码库进行 G711、G726、ADPCM 格式的编码，编码后的码流遵循以下表格中描述的帧结构，即在每帧码流数据的净荷数据之前填充有 4 个字节的帧头；使用语音编解码库进行以上格式的解码时，需要读取相应的帧头信息。

表9-5 海思语音帧结构

参数位置(单位: HI_S16)	参数比特位说明	参数含义
0	[15:8]	数据帧类型标志位。 01: 语音帧; 其他: 保留。
	[7:0]	保留。
1	[15:8]	帧循环计数器: 0~255。
	[7:0]	数据净荷长度(单位: HI_S16)。
2	[15:0]	净荷数据。



参数位置(单位: HI_S16)	参数比特位说明	参数含义
3	[15:0]	净荷数据。
.....	[15:0]	净荷数据。
2+n-1	[15:0]	净荷数据。
2+n	[15:0]	净荷数据。

這邊提供 g711a 一幀的封包範例，我們可以看到每幀前都會加入 00 01 a0 00，4bytes，一幀為 324bytes，扣掉 4bytes 可以得出 data 為

320bytes，所以當上位機編碼完 G711a 碼流後，需在每幀前加入 4bytes 00 01 a0 00，這樣的音頻幀才能順利在嵌入式開發版上播放。

[illegible]

- 自行瞭解：G711、PCM 是什麼，如何撥放。
- 題目：

修改下頁輸出 Wav 的 PCM 碼流格式的 Python 程式，將 PCM 轉換至 g711a，於每幀加入 4bytes 標頭 00 01 a0 00，並將檔案存成 .g711a 檔。

小提醒：frames 為 PCM 碼流，程式最後會在前面加入 44bytes 的 wav 標頭，並輸出 wav 檔，所以只要將 wav 的前 44 個 bytes 刪除就可以修改附檔名得到 .pcm 檔，下圖為 PCM 格式一幀。

[illegible]

- 成果驗收：提供.g711a 檔案與轉換之原始碼，並成功在嵌入式開發版上播放。

```

import pyaudio
import wave

CHUNK = 160
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 8000
RECORD_SECONDS = 5
WAVE_OUTPUT_FILENAME = "demo.wav"

p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,
                 channels=CHANNELS,
                 rate=RATE,
                 input=True,
                 frames_per_buffer=CHUNK)
print("start recording.....")

frames = []

for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    print(data, "\n\n")
    frames.append(data)
print("end!")

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(CHANNELS)
wf.setsampwidth(p.get_sample_size(FORMAT))
wf.setframerate(RATE)
wf.writeframes(b''.join(frames))
wf.close()

```