

BlinkyRecorderについて

2019/03/19 エヴィクサー株式会社 鈴木久晴

ご提供ファイル

blinkyrecorder_release<ビルド日>.zipというファイル名でご提供しております。

ビルド方法

- blinkyrecorder_release<ビルド日>.zipを解凍
- blinkyrecorder_release<ビルド日>フォルダにwifiparameter.hを作成
- Terminal等で、blinkyrecorder_release<ビルド日>フォルダに移動
- makeコマンドでビルド
- make flashコマンドで書き込み
- make monitorでログ確認

wifiparameter.hについて

Wifiのアクセスポイントに接続するための情報と、ネットワーク上のホスト名を設定します。

```
#define HOSTNAME "your hostname"
#define SSID "your ssid"
#define PASSWORD "your password"
```

の3つをご設定ください。

録音方法

- blinkyrecorder_release<ビルド日>/client.pyが録音を行うスクリプトになります。
- コマンド書式
 - python client.py -o <output directory> -b <hostname>:<chmode>:<samplerate>,<hostname>:<chmode>:<samplerate>
 - -o:出力フォルダを指定します
 - -b:mDNSによるホスト名/チャンネルモード/サンプリング周波数でBlinkyRecorderを指定します
 - ホスト名:チャンネルモード:サンプリング周波数を,で接続していただければ複数のBlinkyRecorderから録音を行うことが可能です
- 呼び出し例
 - python client.py -o ./ -b rmic1.local:2:48000,rmic2.local:2:48000
 - カレントディレクトリに、rmic1.local/rmic2.localというBlinkyRecorderからステレオ/48000Hzで録音する
- chmodlには、0/1/2のいずれかの数値を指定します
 - 0:Lchのみ
 - 1:Rchのみ
 - 2:L+Rch
- samplerateには、12000/16000/24000/32000/48000のいずれかをご指定ください

- ホスト名.local_<CHMODE>_<SAMPLRATE>_YYMMDD-hms.wavという名前で録音結果が-oオプションで指定したフォルダに出力されます
- Python2.7で動作確認しております

想定しているシステム構成

- WifiのAP
 - ESP32での接続ですので、b/g/nに対応したものをご用意ください
 - 必要に応じて、APの台数は増やしてください
- BlinkyRecorderを書き込みしたBlinkyボード
 - wifiparameter.hを編集し、ホスト名は各ボードで一意的なものをご設定ください
 - ホスト名.localというのが、mDNSでLAN上にアドバタイズされるホスト名になります
 - wifiparameter.hを編集し、上記WifiのAPに接続を行ってください
- Python2.7の動作するMac or PC
 - 上記BlinkyRecorderに設定したホスト名を指定し、client.pyを実行して録音を行ってください

BlinkyRecorderを複数利用する場合は、有線LANでAPと接続する等で、APの通信量を抑えると性能改善に効果的です

システムの動作

- BlinkyRecorderが、起動するとWifiのAPに接続
- mDNSプロトコルで自分のホスト名をネットワーク全体に通知(mDNSの仕様上、ホスト名.localというのがLAN上の名前となります)
- 8080ポートでclient.pyの接続を待機します
- client.pyは、指定されたホスト名のBlinkyRecorderの8080に接続します
- client.pyは、BlinkyRecorderに対してチャンネル数とサンプリング周波数を設定します
- BlinkyRecorderから、フレームインデックス付きでPCMのrawデータが流れてくるので、それをwavファイルに書き込みます
- フレームのインデックスが連番でないと判断すると、バッファオーバーフローしたと判断し、エラーメッセージが表示されます
- この場合は、BlinkyRecorderの台数、チャンネル数、サンプリング周波数を下げるか、APを増やす等で改善を図ってください

コンパイル調整

システムの限界性能はほぼAPの性能で決まりますので、利用したいBlinkyRecorderの台数に応じて、サンプリング周波数とチャンネル数を変更できるようになっています。

main.cppの冒頭に、以下のパラメータがdefineされており、変更できるようになっています。

```
// AUDIO
```

```
#define AUDIO_CPU_NUMBER (0)
```

```
// Threshold microphone LEVEL for LED illumination
```

```
#define MIN_DB (-45.0f)
```

```
#define MAX_DB (-10.0f)
```

```
// QUEUE Length of PCM buffer from server to client
```

```
// increasing this number may improve overflow
```

```
#define PCM_QUEUE_LEN (30)
```

- AUDIO_CPU_NUMBER
 - ESP32のどのコアでオーディオの処理を実行するかを決定します。
 - Wifi関連のタスクはコア1、オーディオはコア0としております。
 - 今回のファームはぎりぎりのパフォーマンスを狙っているので、変更すると全体のタスク優先度等の変更が必要になると思われます。基本的には変更しないでOKです。
- MIN_DB/MAX_DB
 - LEDを光らせるパラメータです。これまでのものと同様になります。
- PCM_QUEUE_LEN
 - ネットワークのジッターに対応するため、PCM音声をキューにバッファリングしており、そのキュー長です。30に設定しております。

以上