



# 校内放送自動録音システムの開発

## 1. 動機

校内放送では重要なことが連絡されることがあるので、聞き逃さないことが重要です。しかし、校内放送は周囲の騒音・放送のノイズなどによって聞き取れない場合があります。そこで私たちは校内放送を自動で録音しそのデータをSlackを用いて共有することで重要な連絡の聞き逃しを防止できるのではないかと考えました。

## 2. 目的

校内放送を聞き逃した人が再度、校内放送を聞けるようにすること。

## 3. 環境

Raspberry Pi3B+ Python3.7 Pyaudio VS code Windows10

## 3. 環境

### ( i )Slackを使う理由

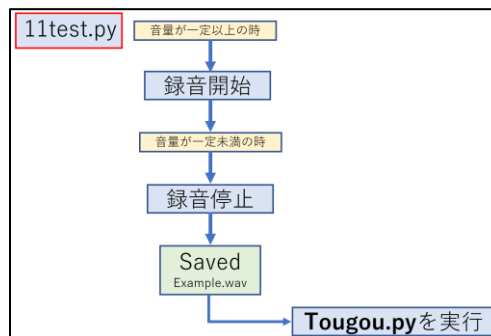
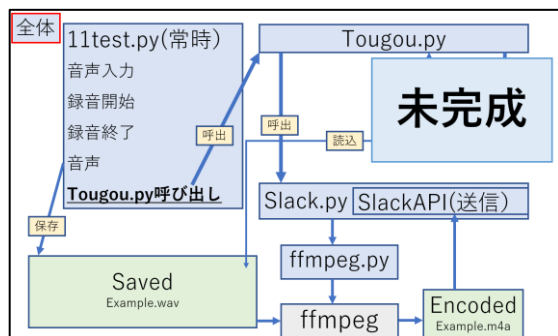
当初に検討していた方式はchromebookからでは利用できないことが分かり、LINE、Discord、Slackなどを使ったほうが互換性が高いのではないかと考えました。そこでそれらの方法を比較した結果Slackを用いた方法が作成、再生ともに簡単なのではないかという結論に行きつきました。

### ( ii )Raspberry Piを使う理由

Raspberry Piは簡易なプログラミング等ができる電子基板です。低消費電力、小型軽量、安価、高い互換性などの特徴を持ちます。今回の私たちの研究ではサーバーとして常時起動するため低消費電力等の観点で選びました。

## 4. 方法

- I. 校内放送のスピーカーの近くにマイクを置き、校内放送の音声を受け取る。
- II. マイクの音声入力が一日以上になったら、録音を開始する。
- III. 音声入力が一以下になったら録音を停止する。
- IV. 音声データをSlack上に公開する。
- V. 生徒からはSlackを介して参照できるようにする。



Tougou.py:11test.pyから受け取ったパスを必要なプログラムに渡す。

Slack.py:Tougou.pyから受け取ったパスをffmpeg.pyに渡したのち返されたパスのファイルのデータをSlackに送信する。

ffmpeg.py:Slack.pyから受け取ったパスのデータを動画圧縮ソフトの「FFMPEG」を用いて圧縮し、Slack.pyにデータを返す。

## 5. 課題

- I チャイム等の誤検知
- II 音源の圧縮形式
- III 圧縮前データや送信後のデータの処理
- IV RaspberryPiの発熱、処理能力
- V 管理の方法
- VI 配置方法

## 6. 展望

- I 音源の圧縮中に録音できるようにする
- II 構成の効率化をする
- III 圧縮後の音声ファイルの保存場所変更をする
- IV 音源の保存先を外部ストレージに変更

[python - file.uploadとrequestsを使用してファイルをSlackにアップロードする方法 \(webdevqa.jp.net\)](#)

[pyaudioを使ってpython上で録音・再生してみよう！ - もろみ先輩の日常 \(hatenablog.com\)](#)