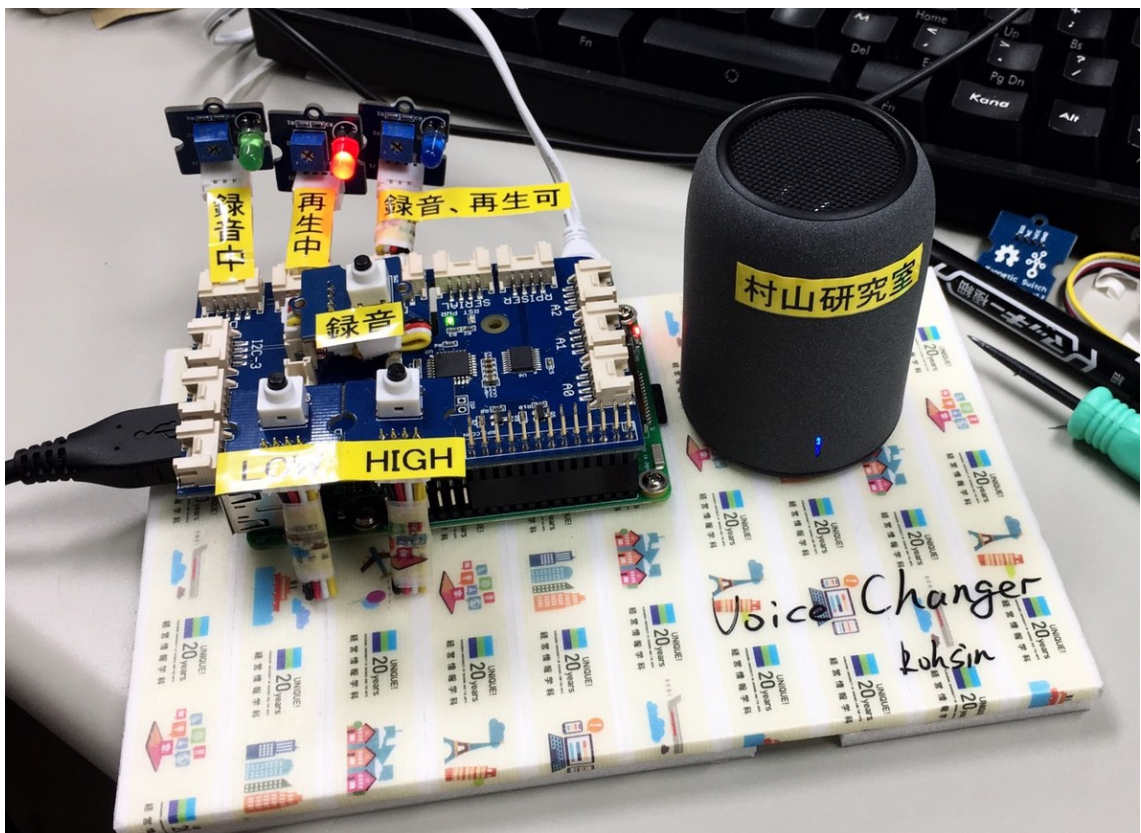


Voice changer

ハードウェア

- ✓ Raspberry Pi 2b (3 bでも対応可能)
- ✓ SDカード(16ギガ以上が勧め)
- ✓ Micro-USB ケーブル
- ✓ USB電源アダプタ
- ✓ AUX 或は Bluetooth スピーカー
- ✓ USB マイク
- ✓ Grove pi
- ✓ Grove LED(青色)
- ✓ Grove LED(赤色)
- ✓ Grove LED(緑色)
- ✓ Grove button(三個)
- ✓ Grove 4pin ケーブル(6本)

① 接続



Voice changer

Grove pi の接続：

D2：青色 LED

D3：赤色 LED

D4：緑色 LED

D6：録音ボタン

D7：再生 HIGH ボタン

D8：再生 LOW ボタン

マウスとキーボード、HDMI、AUX、マイクなどは勿論、電源は最後に接続する。

② ハードウェアの環境設定：

Terminal を開いて：

SSH が必要ならば、設定コマンド：

```
$ raspi-config
```

時刻を手動合わせるコマンドの例：

```
$ date --set='2018/09/27 19:48:20'
```

Grovepi の環境設定：

```
$ sudo curl -kL dexterindustries.com/update_grovepi | bash
```

終わったら

```
$ sudo reboot
```

ファームウェアをダウンロード：

```
$ mkdir /home/pi/sample
```

```
$ cd /home/pi/sample/
```

```
$ sudo git clone https://github.com/DexterInd/GrovePi.git
```

注意：DexterInd の I は、大文字の i で、小文字の L ではないので間違えないように。

```
$ cd GrovePi/Script/
```

```
$ sudo chmod +x install.sh
```

```
$ sudo ./install.sh
```

GrovePi のファームウェアを最新にアップデート：

```
$ cd /home/pi/sample/GrovePi/Firmware/
```

```
$ sudo ./firmware_update.sh
```

```
$ cd /home/pi/sample/GrovePi/Software/Python/
```

```
$ python grove_firmware_version_check.py
```

もし問題なくできれば「GrovePi has firmware version: x.x.x」と表示される。

Voice changer

ソフトウェア

- 録音・再生ソフト：ALSA (Advanced Linux Sound Architecture)
- 音声転換ソフト：SOX (Sound eXchange)
- Python 2.7.13 (デフォルトにインストール済み。Python3 でもいける、プログラムのなかの「thread」を「_thread」に入れ替えたなら ok)
- 編集ソフト：Emacs (使い方はネットから調べると出る)

ソフトウェアをダウンロードするコマンド：

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install alsa-utils sox libsox-fmt-all
$ sudo apt-get install emacs
```

録音に使用できるデバイス一覧を取得する：

```
$ arecord -l
**** ハードウェアデバイス CAPTURE のリスト ****
カード 1: WUP0210 [WUP-021-0], デバイス 0: USB Audio [USB Audio]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
```

```
$ sudo arecord -r 8000 -d 5 -f S16_LE -D plughw:1,0 input.wav
```

- ① -r 8000の意味：一秒钟、8000回の音のアナログデータを記録する
 - ② -d 5の意味：5秒の録音を設定する、つまり5秒後録音は自動的に閉じる
 - ③ -f S16_LEの意味：ファイルのフォーマット
 - ④ -Dの意味：ファイルを保存する場所
 - ⑤ plughw:1,0の意味：[USB Audio]はカード 1, デバイス 0 のところにあるから、デバイスの番号は環境によって違うかもしれない
 - ⑥ input.wavの意味：記録して作ったファイル名
- ！ 録音を強制終了したい場合はcontrol+Cを押す

Voice changer

音声を転換する：

```
$ sudo sox -v 3 input.wav high.wav pitch 1000
```

- ② high.wavの意味：できた新しいファイル名
- ③ pitch 1000の意味：input.wavの高さを1000にする

音声の再生のデバイス設定：

```
$ sudo amixer cset numid=3 1
```

- ① 3 1はアナログ出力の番号
- ② HDMIを使用する場合は、「numid=3 2」
- ③ Bluetoothの場合は、「numid=3」（Grove piをリセットするため、設定したら、Raspberry piを一旦シャットダウンして、電源を切る必要がある。）

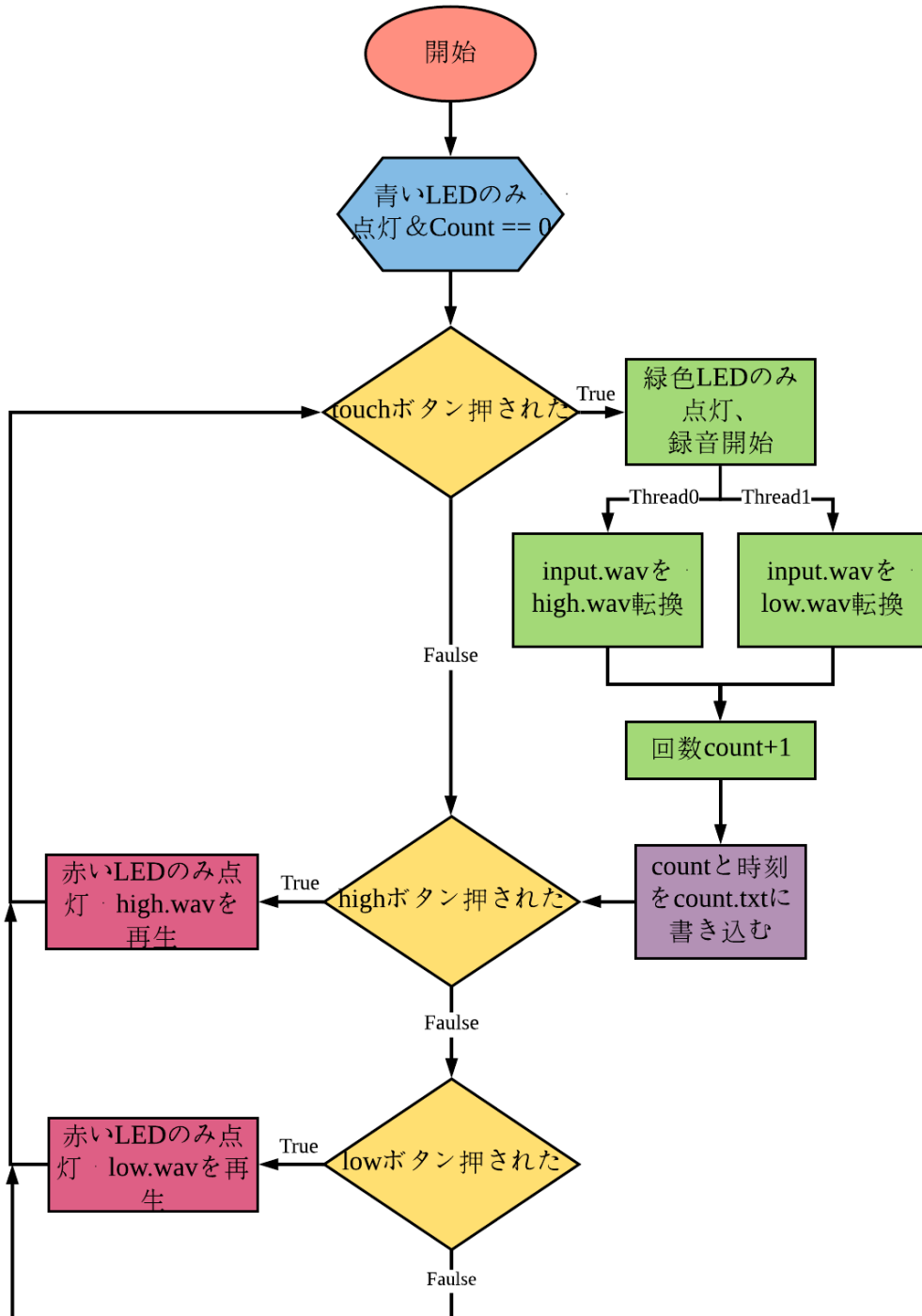
音声の再生：

```
$ sudo aplay high.wav
```

ここでソフトウェアの設定と使い方が終わり。次のページは、Pythonで、ボタンを押すとコマンドを実行できるようなプログラムを作る。

Voice changer

フローチャート



Voice changer

プログラム

```
#!/user/bin/python
#coding: utf-8

#色々な環境の準備#
import time
import grovepi
from grovepi import *
import subprocess as sp
import _thread

#インターフェースの設定#
touch = 6
button_high = 7
button_low = 8
led_red = 3
led_blue = 2
led_green = 4

grovepi.pinMode(touch,"INPUT")
grovepi.pinMode(button_high,"INPUT")
grovepi.pinMode(button_low,"INPUT")
grovepi.pinMode(led_red,"OUTPUT")
grovepi.pinMode(led_blue,"OUTPUT")
grovepi.pinMode(led_green,"OUTPUT")

#色々な命令を定義する#
#ledの状態定義#
def led_red_on():
    digitalWrite(led_blue,0)
    digitalWrite(led_red,1)
    digitalWrite(led_green,0)
def led_blue_on():
    digitalWrite(led_blue,1)
    digitalWrite(led_red,0)
    digitalWrite(led_green,0)
def led_green_on():
    digitalWrite(led_blue,0)
    digitalWrite(led_red,0)
    digitalWrite(led_green,1)
def led_all_off():
    digitalWrite(led_blue,0)
    digitalWrite(led_red,0)
    digitalWrite(led_green,0)
```

Voice changer

```
#録音をする定義#
def record():
    record_args = ['arecord', '-r', '14000', '-d', '5', '-f', 'S16_LE', '-D', 'plughw:1,0', '/home/pi/input.wav']
    sp.call(record_args)

#音を変換する定義#
def process_high():
    process_high_args = ['sudo', 'sox', '-v', '10', '/home/pi/input.wav', '/home/pi/high.wav', 'pitch', '1000', 'speed', '1.2']
    sp.call(process_high_args)

def process_low():
    process_low_args = ['sudo', 'sox', '-v', '10', '/home/pi/input.wav', '/home/pi/low.wav', 'pitch', '-700', 'speed', '0.9']
    sp.call(process_low_args)

#high.wavを再生する定義#
def play_high():
    play_high_args = ['/usr/bin/afplay', '/home/pi/high.wav']
    sp.call(play_high_args)

#low.wavを再生する定義#
def play_low():
    play_low_args2 = ['/usr/bin/afplay', '/home/pi/low.wav']
    sp.call(play_low_args2)

#shutdownの定義#
def shutdown():
    shutdown_args = ['sudo', 'shutdown', '-h', 'now']
    sp.call(shutdown_args)

#ここからmain loop#
#録音回数用count#
count = 0
while True:
    try:
        #ledの最初状態#
        led_blue_on()

        #もし録音ボタン押されたら#
        if digitalRead(touch) == 1 :
```

Voice changer

```
led_green_on()
record()
#録音したファイルを変換する#
print("processing")
try:
    led_all_off()
    _thread.start_new_thread(process_high,())
    process_low()
    print("processing finished")
    led_blue_on()
except:
    print("thread ERROR")
#録音した回数と時期をcount.txtに記録
try:
    count = count + 1
    localtime=time.asctime( time.localtime(time.time())
    ) )
    fo = open("count.txt","a")
    fo.write(str(count))
    fo.write('回録音した。')
    fo.write(localtime)
    fo.write('%n')#¥ではなくて、逆スラッシュだ#
    fo.close()

except:
    print("count ERROR")
#もしhighの再生ボタン押されたら#
if digitalRead(button_high) == 1 :
    print("playing_high")
    led_red_on()
    play_high()
#もしlowの再生ボタンを押されたら#
if digitalRead(button_low) == 1 :
    print("playing_low")
    led_red_on()
    play_low()
#もしlowの再生ボタンとhighの再生ボタンを同時に押されたら#
#実際は同時に長押ししないとシャットダウンできない#
if digitalRead(button_low) == 1 and
    digitalRead(button_high) == 1 :
    shutdown()
#loop 終わり#
except IOError:
    pass
```


Voice changer

プログラムのファイルが要ると kohsin1993@icloud.com に連絡
最後に、必要ならば Raspberry pi の電源をいれたらこのプログラムを自動的に起
動させる。Raspberry pi を起動する前に、Bluetooth デバイスを使用する場合
は先にデバイスを起動させた方がいいと思う。

```
$ sudo emacs ~/.config/lxsession/LXDE-pi/autostart
```

開いたファイルの最後の行に @python /home/pi/pero_last.py を記入。

参考資料：



Arduino によるコンピュータプログラミング入門

分散システム 期末レポート

Grovepi Quick Start Guide

<https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/>

Raspberry Pi と conect+ を Web API でつなげてみた！

<https://www.conect.plus/info/2017/11/965/>

手動で時刻合わせ

<http://katzplus.com/centos-6/ntp/手動で時刻合わせ/>

Grovepi Sample

<https://github.com/DexterInd/GrovePi/tree/master/Projects>

Alsa 問題

<https://bbs.csdn.net/topics/391864071>

Python2.0+Thread

<http://www.runoob.com/python/python-multithreading.html>

Raspberry Pi LED 点滅

<http://www.ic.daito.ac.jp/~mizutani/raspi/blinkingLED.html>

SOX の使い方

<http://computer->

technology.hateblo.jp/entry/20150813/pi#L53%E2%80%9D

<https://qiita.com/moutend/items/50df1706db53cc07f105>

Voice changer

まとめ

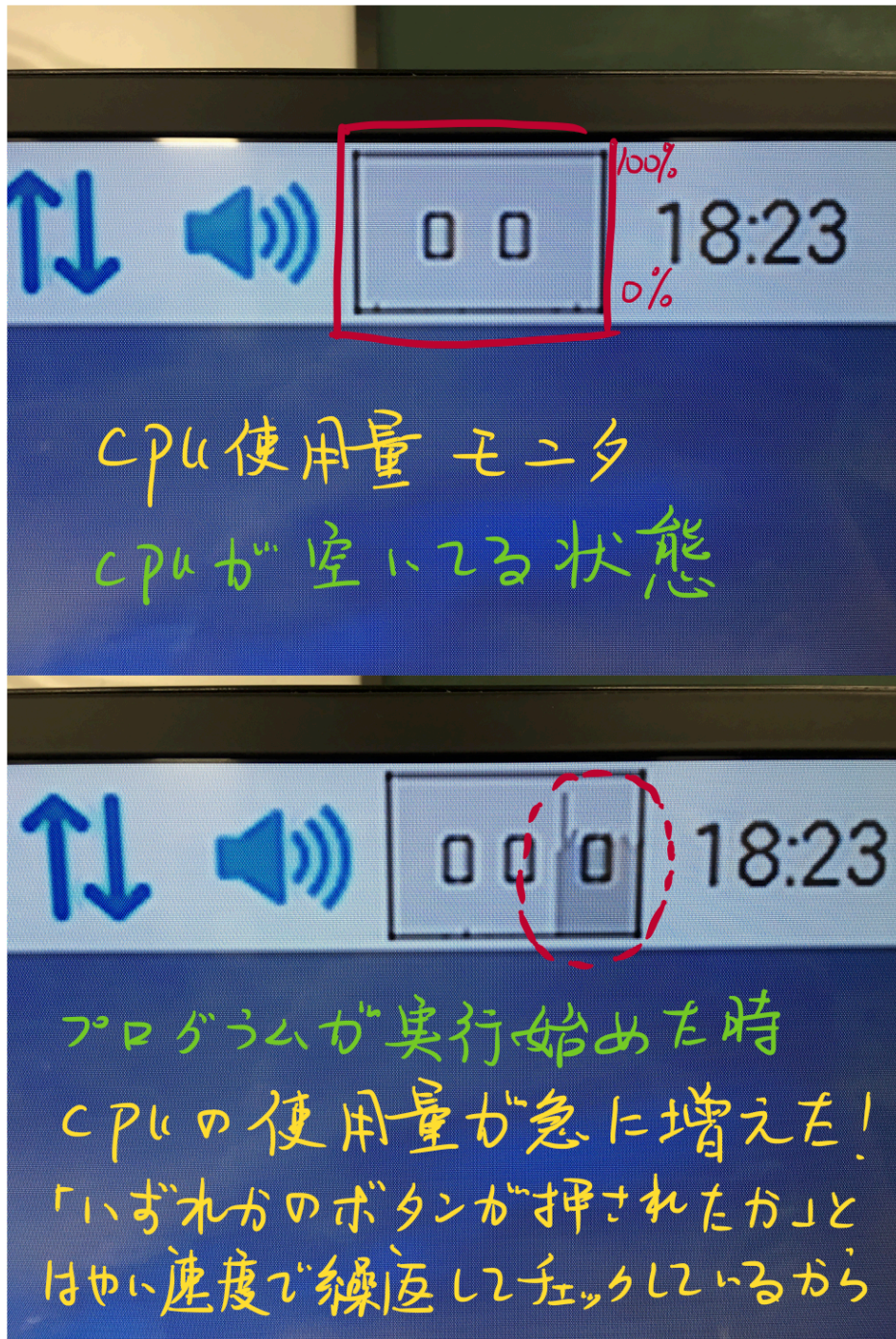
- Subprocess で、python がコマンドする命令ができた。
- -d 5 で、録音は 5 秒に指定して、録音したら止まらないことを解決した。
- Grove Pi で、配線を綺麗になった。
- LED の点滅も工夫した。
- Amixer cset の Bluetooth の設定もできた。
- 録音した回数と時刻も、count.txt に保存した。
- フローチャート図はちゃんと綺麗に描いた。

Thread で、CPU の性能を充分利用して、2 つのファイルを同時に転換する、計算スピードの遅い問題を解決した。

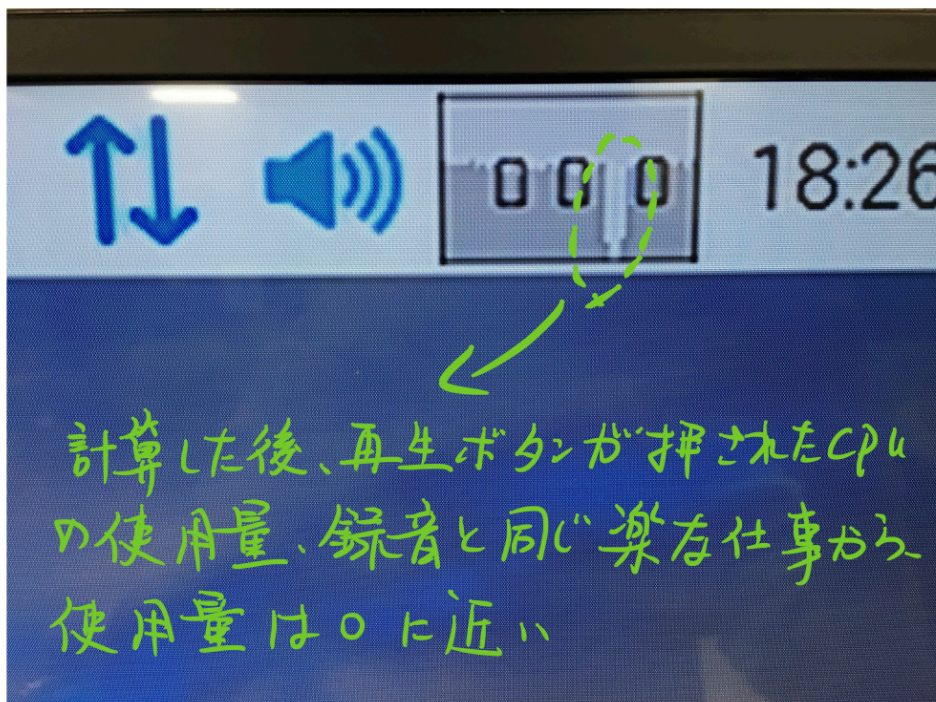
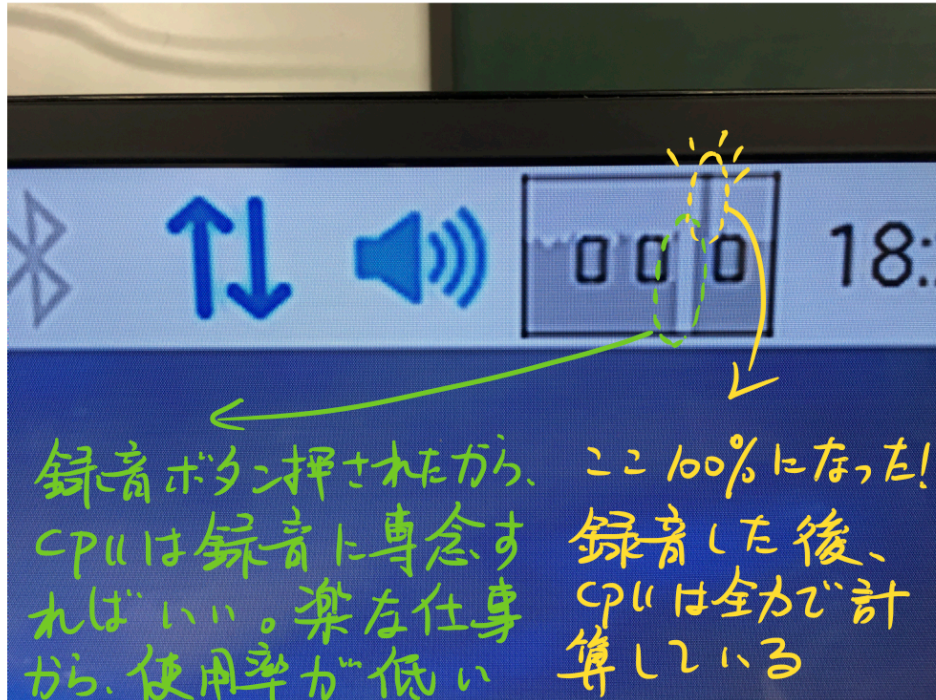
次のページは、このプログラムの実行状態と Raspi 2 B の CPU 使用量の関係の分析：

注：CPU 使用量モニタの中豆腐のような四角は文字化けなので、無視すればよい。文字化けの解決策：`sudo apt-get install fonts-noto` で、`sudo reboot`。

Voice changer



Voice changer



Voice changer

感想：

ただの音声を高く、低くするのは、とても面白いとはいえない、もし、男性の声を自動認識にして、女性の声に転換するなどは面白いと思う。それに関して、sox の使い方を色々調べましたが、公式サイトは英語なのでやめました、英語にもがんばりたいと思う。

この Voice Changer ができるのは、他の人から色々助けてもらった。例えば、Raspberry pi 3b にもプログラムを試してみる時に、沢山の不具合にあった。そのため何時間かかって何回もゼロから繰り返してやっても、無駄でした。最後は、ネットから他人書いたブログを見て、解決できました。

多分プロなプログラマーになるのは、福祉精神も必要だろう。

村 山 研