

【Web 版】

Raspberry Pi で作る！
radiko.jp 専用機の実験

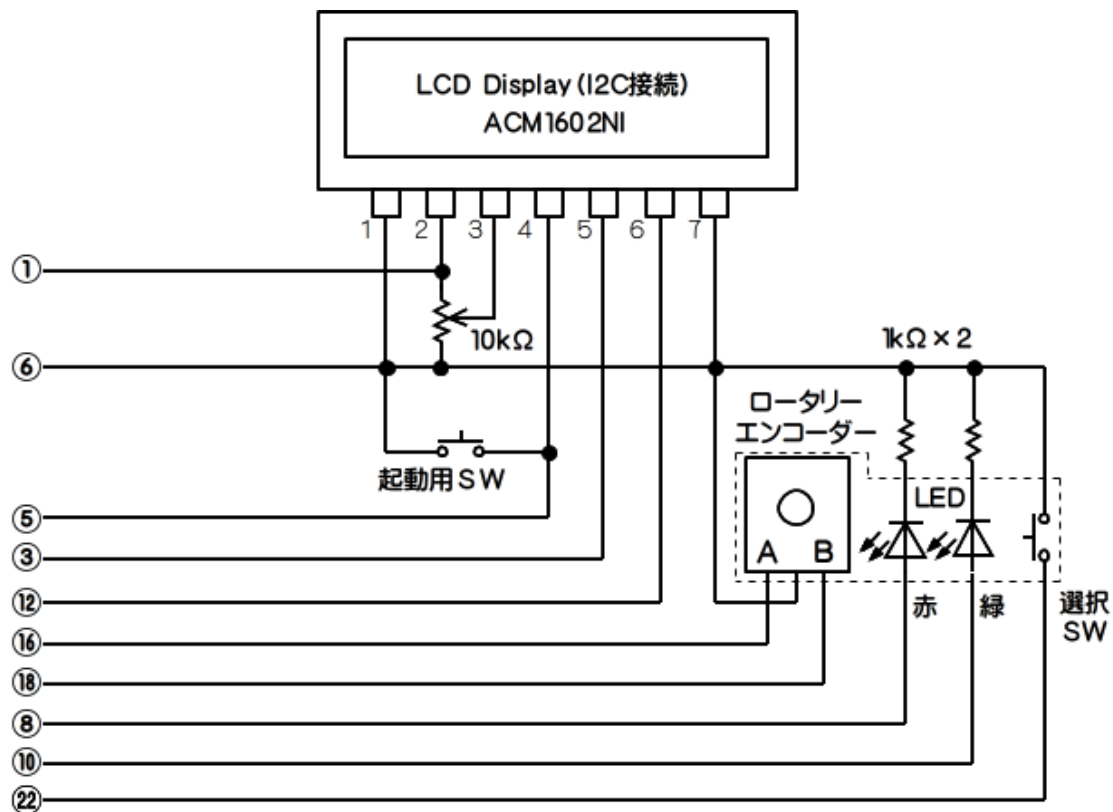
手島伸英（横浜探偵団）

●リード

ラジオマニア 2017 に掲載された「Raspberry Pi で作る！ radiko.jp 専用機の実験」について、執筆時には仮完成だったプログラム本体を追加情報と共に掲載します。

●お詫びと訂正

まずは、お詫びをさせていただきます。これは完全に筆者のミスなのですが、回路図について誤ったものを編集部に送ってしまい、明らかに校了日を過ぎたタイミングまで気が付きませんでした。以下に正しい回路図を掲載しますので、実際に製作される場合にはこちらを参照下さい。



●修正点と変更ポイント

まずは、修正点ですが、記事中で Raspberry Pi と LCD Display ACM1602NI との I2C 通信速度を設定しました。記事執筆時にはデータの送信に際して待ち時間を設定したりして問題なく動作していたのですが、プログラムが大きくなるにつれて「OSError: [Error 121] Remote I/O error」が頻発するようになりました。

色々と調べたところ、I2C の接続速度が正しく変更されていなかったことが原因と分かりました。本来、起動に関連したファイルはいじりたくないのですが、やむを得ず、config.txt を変更したところ、ウソのようにエラーが出なくなりました。具体的には、config.txt に以下の一行を付け加えます。

```
dtparam=i2c_baudrate=50000
```

この時、管理者権限が無いと保存時にはじかれてしまうので、書き換えは管理者権限で行います。vi など、お馴染みのエディターを使ってもかまいませんが、筆者は苦手なので、メモ帳 (leafpad) を使いました。以下のコマンドを打ち込むと、メモ帳から config.txt を変更できます。起動に影響する部分なので、慎重に書き足して下さい。

```
sudo leafpad /boot/config.txt
```

次に、操作に関して。放送局名リストの中に「Shutdown」メニューを入れるつもりでしたが、勢いで Shutdown を選択しがちなので、勝手ながら「選択ボタンの長押しで Shutdown 実行」と変更させて頂きました。

●ダウンロード

プログラムは、以下のリンクからダウンロードできます。記事中で入手したスクリプト「play_radiko.sh」と同じディレクトリに格納したら、以下のコマンドを入力して起動します。

```
python3 radikotest_103.py
```

【■radikotest_103.py ダウンロードリンクは HP 上にあります】

【■ (参考) play_radiko.sh ダウンロード先】

<https://www.erestage.com/develop/raspberry-pi-radiko/>

●プログラムの説明

冒頭部分は、ほとんどセッティングです。「Config Setting」の値を変えれば、ある程度、好みの動作に変更できます。例えば、BLTIME=10 はバックライトの点灯時間 10 秒のこと。CLICK_TIME=0.02 は 20 ミリ秒選択ボタンが押されたら「選択」とみなす、SHUTDOWN_MODE=0 はシャットダウン時に電源を切るかどうか、SHUTDOWN_TIME=3 は 3 秒長押しでシャットダウン実行といった感じです。

その後、もろもろの初期化を実行して、受信エリアの判定を行います。<http://radiko.jp/area> にアクセスすると Java Script がダウンロードされるのですが、この中にエリア情報が含まれているのです。


内容は、こんな感じ…。

エリアコード「JP14」「KANAGAWA」という内容が読み取れます。

```
document.write('<span class="JP14">KANAGAWA JAPAN</span>');
```

本プログラムでは、この情報を元に放送局リストを取得し、ユーザーのエリアに応じた放送局が表示されるよう設計しています。

【図版：放送局データを取得したところ.png】



```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi: ~ $ python3 radikotest_103.py  
JP14  
stations {'area_id': 'JP14', 'area_name': 'KANAGAWA JAPAN'}  
KANAGAWA  
Station ID = 0 TBS TBS RADIO  
Station ID = 1 QRR JOQR BUNKA HOSO  
Station ID = 2 LFR JOLF NIPPON HOSO  
Station ID = 3 RN1 RADIONIKKEI  
Station ID = 4 RN2 RADIONIKKEI2  
Station ID = 5 INT InterFM897  
Station ID = 6 FMT TOKYO FM  
Station ID = 7 FMJ J-WAVE  
Station ID = 8 JORF RF RADIO NIPPON  
Station ID = 9 BAYFM78 bayfm78  
Station ID = 10 NACK5 NACK5  
Station ID = 11 YFM Fm yokohama 84.7  
Station ID = 12 HOUSOU-DAIGAKU HOUSOU-DAIGAKU
```

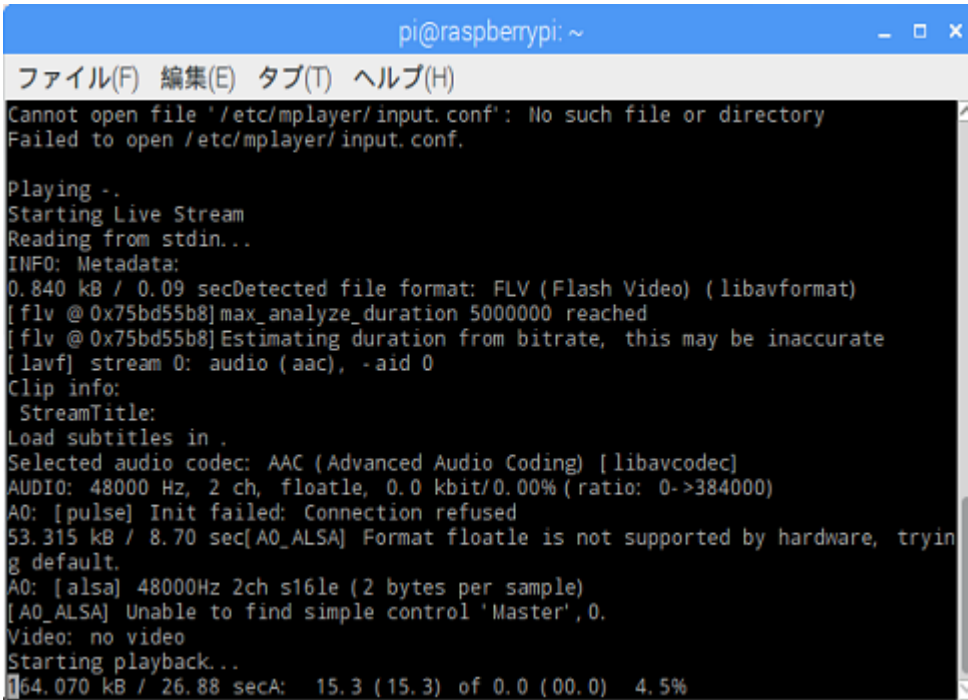
いよいよ、操作系の認識とステータスの表示ですが、ロータリーエンコーダーは「位相」というものを使って回転方向を表現するデバイスです。Raspberry Pi の GPIO には「状態の

変化」を検出する機能が無いため、プログラム側で検出を行いました。わずかな変化の時間差がポイントとなるのでプログラムで追い切れるか心配でしたが、なんとか、実用範囲には収まったと思います。これで、ツマミの回転で放送局が選べるようになりました。

ちなみに、使用したロータリーエンコーダーは、秋月で取り扱っている「EC12PLGRSDVF」という製品。ロータリーエンコーダーとプッシュスイッチ、赤色 LED、緑色 LED が一体となったデバイスです。

再生は、選択ボタン（プッシュスイッチ）が押されたことを検出して、外部のスクリプト「play_radiko.sh」を起動する仕組みです。この時、さきほど取得した放送局リストに含まれている局名コード（例えば、J-WAVE なら「FMJ」）を使って放送局情報を渡しています。

【図版：プレーヤーが起動したところ.png】



```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
Cannot open file '/etc/mplayer/input.conf': No such file or directory
Failed to open /etc/mplayer/input.conf.

Playing -.
Starting Live Stream
Reading from stdin...
INFO: Metadata:
0.840 kB / 0.09 secDetected file format: FLV (Flash Video) (libavformat)
[flv @ 0x75bd55b8]max_analyze_duration 5000000 reached
[flv @ 0x75bd55b8]Estimating duration from bitrate, this may be inaccurate
[lavf] stream 0: audio (aac), -aid 0
Clip info:
StreamTitle:
Load subtitles in .
Selected audio codec: AAC (Advanced Audio Coding) [libavcodec]
AUDIO: 48000 Hz, 2 ch, floatle, 0.0 kbit/0.00% (ratio: 0->384000)
AO: [pulse] Init failed: Connection refused
53.315 kB / 8.70 sec[AO_ALSA] Format floatle is not supported by hardware, trying default.
AO: [alsa] 48000Hz 2ch s16le (2 bytes per sample)
[AO_ALSA] Unable to find simple control 'Master', 0.
Video: no video
Starting playback...
64.070 kB / 26.88 secA: 15.3 (15.3) of 0.0 (00.0) 4.5%
```

再生まではさほど難しくなかったのですが、このまま次に選択ボタンを押すと、どんどんプレーヤーが増えて音が二重三重に聞こえてしまうという怪現象に陥りました。

よくよく調べてみると、本プログラムから呼び出したスクリプトは子プロセスとして動作するのですが、そこから Raspberry Pi 内蔵の mplayer を起動する一連の流れが孫プロセスとして認識されていない様子で、本プログラム自体を終了しても音が鳴り続ける始末。

なので、あまりいい方法ではないのですが、現在実行中のプロセスから「mplayer」という名前のプログラムのプログラム ID (PID) を検出して、直接止めることにしました。結果、うまくいって、怪現象は解消されました。

●操作

起動したら、ツマミを回して「選局」し、選択ボタンを押して「再生」です。プレーヤーが止まっているときにはツマミが「赤」に光り、再生中は「緑」に光ります。音は、ヘッドホンジャックからステレオで流れてきますので、ヘッドホンで聴いても外部スピーカーにつないでも OK。終了するときは選択ボタンを長押し（3 秒）すれば、プログラムが終了します。

●おしまい

エリアの判定機能など、当初は予定に無かった機能を盛り込んだり、けっこう楽しむことができました。Raspberry Pi は、プログラムと追加ハード次第で何にでも化けるデバイスです。機会があれば、エリアフリーへの対応など、機能拡張にチャレンジしていきたいと思っています。ぜひ、編集部宛てに励ましのお便りを！

●参考

【I2C 接続キャラクタ LCD モジュールリンク】

<http://akizukidenshi.com/catalog/g/gP-05693/>

【2色 LED 付スイッチ付ロータリーエンコーダ（赤・緑）リンク】

<http://akizukidenshi.com/catalog/g/gP-05771/>