

耽溺せよ
電子工作

おとな ラズパイリー

第6回

「Raspberry Piをメディアサーバ／プレーヤにしよう(後編)」

おとなラズパイリーは、Raspberry Piを文字どおり「リレー」し、好奇心旺盛なITエンジニアが電子工作をするという企画です。前編で構想を練り、後編で実装します。1年を通してどんなデバイスができるのか?……今回は、話題の新刊も好調な増井先生によるRaspberry Pi B+でGear実装編です。

Writer 増井 俊之(ますい としゆき) 慶應義塾大学 環境情報学部教授

Raspberry Pi 2 デビュー

先月号が発売された直前(2014年2月)にRaspberry Pi 2^{注1}が発表されました(写真1)。従来型のものに比べてCPU性能が格段に向上し、X11のデスクトップ環境が格段に快適になったにもかかわらず価格は据え置きなので迷わずRaspberry Pi2を購入するのが良いと思われます。高速化によって、ますます汎用コンピュータとしての使い勝手が向上したといえるでしょう。

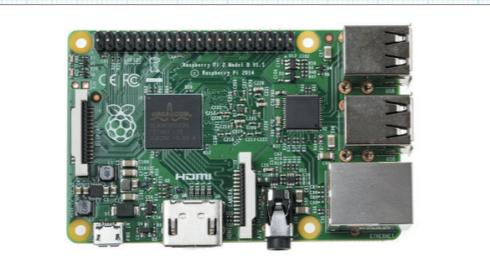
先月号ではRaspberry Piをメディアプレーヤとして活用する「Gear」というシステムの構想を紹介しましたが、今回はこの実装について解説します。

コンテンツの再生

Raspberry Piで動画を再生するのにGPUを使用するomxplayer^{注2}がよく使われています。omxplayerはmp3のような音楽ファイルやネットからの動画ストリーミングも再生できるので、メディアプレーヤとして使う場合はomxplayerだけ用意しておけば充分でしょう。

```
$ sudo apt-get install omxplayer
```

▼写真1 Raspberry Pi 2



メニュー表示方式の選択

Raspberry Piで画面に描画を行うには次の方法があります。

1. フレームバッファに書き込む

Raspberry Piでは、ほかの多くのLinuxシステムと同じように、/dev/fb0にデータを書き込むことによってビットマップ画面を表示できます。Raspbianのコンソール画面はこれを利用して文字を描画しています。

2. GPUでグラフィクスを描画する

Raspberry PiのCPU/GPUなどを1つのチップ上に登載したBCM2835^{注3}というSoCチップ上のGPU(VideoCore)がサポートして

いるOpenGL ES2などを使って高速なグラフィクス表示が可能です。

これらの画面データは独立しており、GPUで描画した画像データを/dev/fd0から読むことはできません。

これらのハードウェア機能を利用して、次のような方法で文字や図形を描画できます。

1. X11を使う
2. フレームバッファに直書きする
3. OpenGL ES2^{注4}を使う
4. SDL^{注5}を使う

これらの選択肢がありますが、omxplayerで動画を表示する場合、GPUベースの描画システムを使うことができません。つまり、OpenGL ES2やSDLでメニューを表示できません。また、X11上の描画システムはomxplayerと共存はできるのですが、ブラウザのJavaScriptでは軽快にメニュー表示できません。Javaなどを使う場合も同様です。

幸いGearのメニューはテキストさえ表示できればそれなりに使えるので、今回はfbterm上に表示するテキストでメニュー操作を行うことにします。残念ながら文字位置を行/桁単位でしか指定できませんし、色も規定された数種類しか利用できず、現代のコンピュータとは思えないメニュー表示になってしまいますが、今回はとりあえずそれで我慢したいと思います。ターミナル上のテキスト表示にはncursesという古典的なライブラリを利用します。

入力装置の選択

普通のパソコンではキーボードとマウスが標準

的な入力装置として利用されていますが、Raspberry Piではこれらの装置を普通に使えることに加え、ボード上のGPIO(General Purpose I/O)端子を入出力に利用することができるので、ちょっとしたスイッチやLEDなどを接続して利用できます。Gearは2個の単純な入力装置があれば実装できるので、GPIOポートにマイクロスイッチなどを直結して利用することもできます。たとえば椅子にマイクロスイッチを接続しておいて、そこからケーブルをGPIO端子に接続すれば、椅子の回転や傾きでGearの操作ができるというわけです。その他、各種のMIDIコントローラのようなUSB接続の入力装置も利用できます。

今回はワイヤレスマウスのホイールを入力装置として使ってみることにします。ワイヤレスマウスは数百円で売られていますし、Raspberry PiではX11を使わなくても/dev/input/event0からマウス操作を取得できるので、ホイールの前方/後方回転でGear操作を行なうことにします^{注6}。

Gearの実装

入力方法と出力方法が決まったので、Raspberry Pi上にGearを実装していきます。すべてをRubyで記述します。

メニュー表示

デフォルトの起動ターミナルは日本語が表示できませんしフォントサイズが小さいので、フレームバッファを利用したターミナルプログラムfbtermを利用して大きな文字を表示します。

```
$ sudo apt-get install fbterm  
$ fbterm --font-size=50
```

ncursesはgemでインストールし、テキスト

注4) http://ja.wikipedia.org/wiki/OpenGL_ES

注5) <http://www.libsdl.org/> <https://wiki.libsdl.org/Introduction>

注6) ちなみにRaspbianで初期設定されている「pi」というアカウントではomxplayerやマウス入力を問題なく利用できますが、異なるアカウントで使おうとする場合は/etc/groupの設定を変更したり/dev/vchiqのパーミッションを変更したりする必要があるので、メディアサーバとしては「pi」アカウントを利用するのが楽なようです。

注1) <http://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

注2) <http://omxplayer.sconde.net/>

注3) <http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

▼リスト1 階層ltsvを使用したコンテンツファイル(<tab>はタブ文字)

```
title:YouTube
title:増井俊之
title:Slime<tab>file:http://www.youtube.com/watch?v=LdyI5UbbSA8
title:Gazz<tab>file:http://www.youtube.com/watch?v=RatK2q6SwFA
title:Dynamic Macro<tab>file:http://www.youtube.com/watch?v=payhP07zi4w
title:音楽
title:John Coltrane
title:My Favorite Things
title:My Favorite Things<tab>file:/home/pi/Music/John Coltrane/My Favorite Things/01 My Favorite Things.mp3
title:Everytime We Say Goodbye<tab>file:/home/pi/Music/John Coltrane/My Favorite Things/02 Everytime We Say Goodbye.mp3
title:Summertime<tab>file:/home/pi/Music/John Coltrane/My Favorite Things/03 Summertime.mp3
title:But Not For Me<tab>file:/home/pi/Music/John Coltrane/My Favorite Things/04 But Not For Me.mp3
title:ムービー
title:Rocky Horror Show<tab>file:/home/pi/Movies/RockyHorrorPictureShow.mp4
title:Planet Earth<tab>file:/home/pi/Movies/PlanetEarthA-1.mp4
```

の表示にはncursesで用意されている次の関数を利用します。

```
require "curses" # ncursesライブラリ
setpos 10, 2 # 10行目の2桁目にカーソル移動
addstr "abc" # カーソル位置に"abc"表示
```

タイムアウト処理

Gearでは、選択中の項目に子要素が存在するときユーザが何も操作を行なわなければ要素を自動的に展開してその最初の項目を選択するようになっています。このようなタイムアウト処理はJavaScriptではsetTimeout()で簡単に指定できますが、Rubyにはタイムアウト関数が用意されていませんので、ruby-concurrencyというgemを使うことになります。10秒後に何かを実行したい場合は次のようにします。明示的にスレッドを起動する必要がないので簡単です。

```
require 'concurrent'
Concurrent::ScheduledTask.execute 10 do
  do_something # 10秒後に実行される処理
end
```

コンテンツの記述

音楽や動画などのコンテンツは内容に従って階層的に分類しておきます。階層的なデータを表現する方法はいろいろありますが、ここでは

▼リスト2 メインルーチン抜粋

```
File.open("/dev/input/event0","rb"){|f|
  while true do
    s = f.read 16
    (time, type, code, value) = s.unpack "qssi"
    if type == 2 and code == 8 then
      move value
    end
  end
}
```

ltsv⁷⁾にインデントを追加した「階層ltsv」を使うことにします。たとえばリスト1の記述では、「YouTube」の「増井俊之」に3個のコンテンツが含まれていることを表現しています。

マウスホイール操作の取得とナビゲーション

メインルーチンでリスト2のようにしてマウス操作を取得します。

階層ltsvで記述したコンテンツ記述ファイルは最初に木構造に変換しておき、ホイール操作があるたびにmoveメソッドが呼ばれて選択項目とメニュー表示が変化し、選択中のコンテンツをomxplayerで再生します。

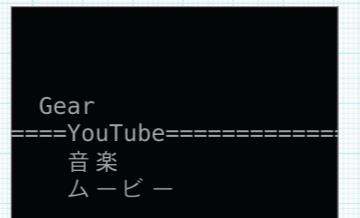
omxplayerによるコンテンツ再生

階層ltsv内の「file:」で指定されているコンテンツを再生するときはリスト3のplayメソッドを呼び出します。YouTubeの場合はyoutube-dlと

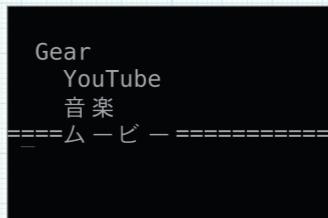
▼リスト3 コンテンツ再生部分のコード

```
def play(file)
  system "killall omxplayer omxplayer.bin > /dev/null 2> /dev/null"
  if file =~ /^http.*youtube.com/ then
    stream = `youtube-dl -g #{file}`
    system "omxplayer '#{stream.chomp}' > /dev/null &"
  elsif file =~ /^$/ then
    system "omxplayer '#{file}' > /dev/null &"
  end
end
```

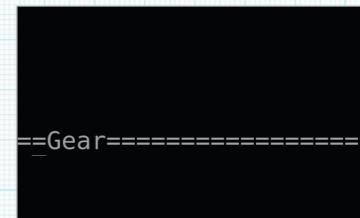
▼図2 メニュー表示



▼図3 選択画面表示



▼図1 Gearの起動



▼図4 メニュー展開画面



いうプログラムを使ってストリーミング元URLを計算してからomxplayerを起動します。

Gearの動き
Gearを起動すると最初に図1のようなテキストが表示されます。何もせずに1秒待つと図2のようにメニューが展開されます。マウスホイールを回転させると図3のように選択項目が移動します。ここで1秒待ってメニューを開かせた後でマウスホイールを回すと図4のように「Planet Earth」を選択することができます。

メニューの見栄えはパッとしませんが、マウスホイールを回すだけでコンテンツを選択して再生できるメディアプレーヤを作ることができます。ソースコードは全部で150行ぐらいで、GitHub⁸⁾に置いてあります。

別の工夫によってメニューをもっと格好良くすることはできるかもしれませんし、Raspberry Pi 2だとブラウザで問題なくプレーヤを構築できるかもしれません。いずれにしても、Raspberry Pi

をGearのようなインターフェースでメディアプレーヤとして使うことは将来性があるだろうと思っていますので、ぜひ試していただければと思います。

おまけ

2月に『スマホに満足しますか?—ユーザインターフェースの心理学—』という本を出した(図5)。ユーザインターフェースにかかる面白い話をたくさん集めていますので読んでみていただければ幸いです。SD

◀ 図5 話題の新刊
(http://www.amazon.co.jp/dp/433403845X)



注7) http://ltsv.org/

注8) https://github.com/masui/RasPiGear