

プロも使うYocto開発環境で初体験!

ダウンロード・データあります

ラズパイ時代のレベルアップ! MyオリジナルLinuxの作り方

第32回 Webダウンロード対応にして
早送りタイムラプス動画カメラを仕上げる

三ツ木 祐介

作成している早送りタイムラプス動画カメラの構成を図1に示します。

ウェブ・インターフェースの検討

タイムラプス動画を作成しダウンロードするためのウェブ・インターフェースの検討を行います。大まかな仕様は次の通りです。

- ボタンをクリックするとtimelapse.shを呼び出す
- 作成後はダウンロード用のリンクを表示する
- 動画のファイル名はtimelapse_YYYYMMDDHHmmSS.ts
- 動画は最新の1つしか保持しない

ウェブ・アプリケーション・フレームワークFlaskを使うとHTML自体の記述を大幅に減らすことができます。Flaskでプログラムを作成する場合、主に次のものを作成する必要があります。

- サーバ側の処理
- HTMLのテンプレート

● 作るもの1：サーバ側の処理

/home/root/timelapse.pyをリスト1の内容で作成します。@app.routeの行の直後に定義されている関数はブラウザからのアクションをトリガにフレームワークから呼び出されるようになっています。

多くの処理をFlaskのフレームワークが肩代わりし

てくれるので非常に少ない記述でサーバ側の処理を書くことができます。

デフォルトでは他のPCなど外部からのアクセスを受け付けないため、app.run()の引き数に'0.0.0.0'を設定することで外部からのアクセスを許可します。

● 作るもの2：HTMLのテンプレート

FlaskではJinja2記法によってHTMLのテンプレートを処理することができます。テンプレートを使用することで冗長になりやすいHTMLの記述を削減することができます。{% block %}～{% endblock %}の記述がテンプレートとなっており、ブラウザでアクセスするとこの部分が定義済みのHTMLに変換されるようになっています。

▶ contentブロック

主にインターフェースの部分を記述するブロックとなるcontentブロックを作成します。

/home/root/templates/timelapse.htmlをリスト2の内容で作成します。

```
{% extends "bootstrap/base.html" %}
```

ベースとなるHTMLを取り込みます。

本体は{% block content %}～{% endblock content %}の部分で、画面に必要な要素を定義しています。ここが画面のデザインとなります。

▶ scriptブロック

サーバとのやり取りを行う機能を記述するためのscriptブロックを作成します。

/home/root/templates/timelapse.htmlのリスト2の後にリスト3の内容を追加します。サーバとの通信の部分はAjaxを使用しています。サーバの処理が成功すると受け取ったダウンロード用のURLをもとに、リンクを作成するようになっています。サーバの処理中に待ち画面を表示するためにfakeloader⁽⁴⁾を使用しています。最新は2.0.0ですが今回は1.1.0を使用しています。{{ super() }}によってフレームワークが必要とするスクリプトの取り込みを行っています。

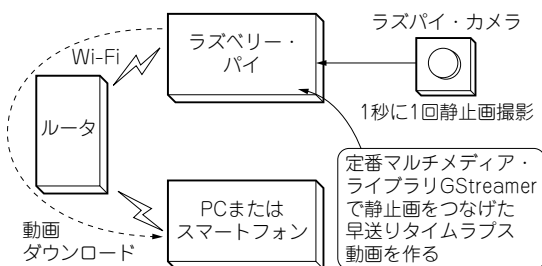


図1 早送りタイムラプス動画カメラを作る