

遠隔地用!!

# WebCam

をつくってみた

Hiroki @ WebCamを愛するもの

## 事の発端

スマートフォンやタブレット端末が普及してきた昨今、小型のデバイスが比較的容易に入手可能となりました。しかし、少し高めのものが多く、基本的には万単位のお金が必要となります。

そんな中、小型のデバイスを安価で提供し、プログラミングの勉強などに使おうという製品、Raspberry Pi<sup>1)</sup>というものが登場しました。お値段なんと3000円ほど。USB、LANポートの他にもHDMI出力が可能でLinux等が動作します。それが手のひらサイズ。これは素晴らしいおもちゃです。

## ああ、届かない...

世界中で話題になったこの製品。安価なので並列動作とかすげえ楽しそう！と買ったのはいいものの、到着予定の10週間を超えても手元に本体がありません。人気過ぎて世界中で生産が追いついてないとか。うおお！ このままでは届かなかったから作ってみた妄想記事になってしまう！！と絶体絶命のピンチに陥ってしまったわけですが、別ルートで輸入したらあっさり2週間ほどで届きました。あれ？ 日本公式……

とりあえず世界的にはだいが需要に供給が追いついてきたようなので、輸入すればそこそこ早く入手できるようになりました、日本の販売代理店の方はまだ予約の処理でいっぱいっぽいようなので、早く手軽に入手可能になることを祈りたいですね。

## WebCamって何？

おはようからおやすみまであなたを見守り続けるWebCamSystem略してWebCamとは、X680x0同好会の部室のセキュリティ対策として導入されている監視システムです。一定時間ごとにWebカメラを使って部室を撮影し、アーカイブ化して保存しています。

私が入学した当初から存在するこのシステムには2つの欠点が存在していました。1つはWebカメラのドライバで、ドライバさえあれば写真を取ることは容易ですが、Linux向けのドライバは少なく、ドライバ探しは大変でした。これに関しては、再開発を行った際に世界共通規格のUVC<sup>2)</sup>を使うことで改善されました（詳しくは2010年のX68通信を参照<sup>3)</sup>）。

もう1つの欠点は、USBによるWebカメラの接続の制限です。USBの規格上の最大長は5m。延長用のケーブルを使っても20m程度が限界で、そういったケーブルは高価でWebカメラと同等程度の費用がかかります。やっつけられません。この記事ではその2つ目の問題をRaspberry Piを使って解消しようと思います。

1) 英国のラズベリーパイ財団が開発したARMを搭載したシングルボードコンピュータ。

2) USB Video Classの略。このドライバに対応したWebカメラであれば指すだけでOSを問わず認識可能。もっと似たような規格が出ることを切に願います。

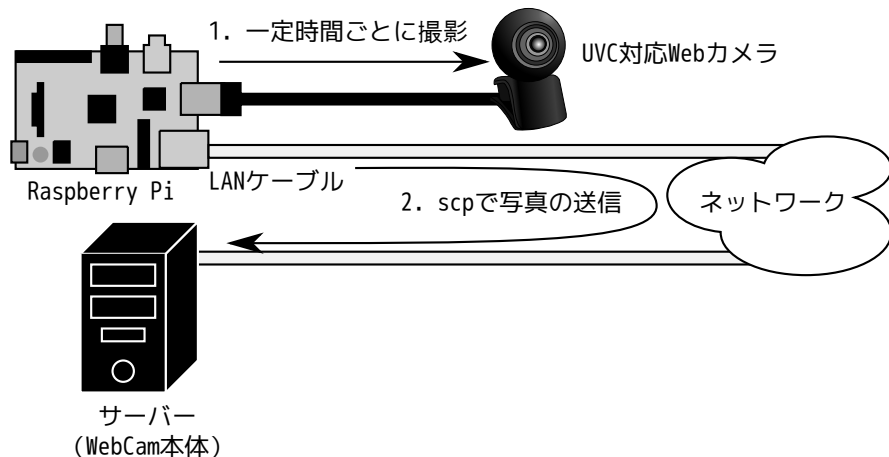
3) X680x0同好会公式サイト( <http://www.x68uec.org/> )のその他の作品 > デジタル版会誌 > 2010年度 (Vol.16) にて読むことができます。記事のURLは <http://www.x68uec.org/other/press/2010/20/index.html>

# 設計

WebCamには以下の様な特徴があります。

- ・一定時間ごとに撮影
- ・任意のタイミングで撮影可能
- ・ファイルは一定期間分だけ保存  
(なお、再構築した新しいWebCamは1週間ほどを生で、それ以降1ヶ月分はZIPで保存)

この中でも一番下の項目は、Raspberry Piのような小型でストレージが貧弱なデバイスには辛い仕様となります。そこで、遠隔地用のWebCamとして構築することを考え、以下の様な構造にします。



撮影に関してはスクリプト化することになるので、SSH接続してスクリプトを動作させることで、追加の撮影要求も可能となるでしょう。その場合は固定IPアドレスなどを割り振る必要がありますが、今回はDHCP環境下を想定して、ひとまず単方向に特化させます。

## 必要なもの

今回の遠隔地用WebCamを構築するために使うのは以下のものです。

- ・ Raspberry Pi
- ・ SDカード (2GBは怖いので4GB以上。micro SDをSDに変換しても可能)
- ・ micro USBケーブル (電源用。スマートフォンと同じもの)
- ・ LANケーブル
- ・ HDMIとその入力に対応したディスプレイ
- ・ UVC対応Webカメラ (今回は生協で購入した UCAM-DLA200H を使用)
- ・ キーボード
- ・ WebCamシステムの本体となるサーバー (インストールにも使用。Linux/UNIXであること)
- ・ SDカードライター

基本的なものは安価でそろそろと思います。一番厳しいのはインストール用のサーバーやディスプレイになると思いますが、このシステムを構築する上では必要なので頑張りましょう。

# Raspberry Piのインストール

## Raspberry Piの動作について

Raspberry PiはSDカードをストレージとして使用します。そのためSDカードにOSをインストールする必要があるのですが、普通のOSで使用可能な光学ドライブはなく、USBやネットワークインストールも可能かどうか怪しいです。

そこで、Raspberry Piは公開されているOSのイメージをそのままSDカードに書き込む方法を使ってOSをインストールします。

## OSイメージのコピー

SDカードにイメージを書き込む、と聞いてそのままSDカードにデータを書き込むことを想像した人には残念なお知らせです。そんな簡単にOSがインストールできたら誰も苦労はしません。普通に書き込んでOSのブートはできないので、ファイルシステムをぶち壊してSDカードの始まりからデータを書き込むことで、OSをインストールします。

手順は以下のとおり。

1. OSイメージをダウンロード後、展開。
1. サーバーにSDカードライターを挿し、SDカードを認識
2. デバイス名確認後、SDカードをマウントしていればアンマウント
3. マウントしていないSDカードに対してddでイメージを書き込む

今回のインストールにはCentOS6をインストールした自宅サーバーにやってもらいました。Raspberry Pi側はRaspbianをインストールします。以下はその時の作業です。

1. OSイメージのダウンロードと展開

イメージは <http://www.raspberrypi.org/downloads> からダウンロードしてください。

```
$ wget http://downloads.raspberrypi.org/images/raspbian/2012-09-18-wheezy-raspbian/2012-09-18-wheezy-raspbian.zip # ダウンロード
$ unzip 2012-09-18-wheezy-raspbian.zip # 解凍
Archive: 2012-09-18-wheezy-raspbian.zip
inflating: 2012-09-18-wheezy-raspbian.img # 解凍されたイメージ
```

2. SDカードライターの認識

今回は他のメディアも認識する複合型のものを使いました。ポイントは何番目がどのメディアに対応しているかを知ることです。Windowsで調べることも可能ですが、今回はメッセージと認識順からSDカードに対応するデバイス名を見つけました。GUIなどで自動マウントしてしまった場合は、dfコマンドでデバイス名が分かるので楽だと思われます。

```
# cat /var/log/message
Oct 17 22:56:28 beryl kernel: scsi 6:0:0:0: Direct-Access Generic USB SD
Reader 1.00 PQ: 0 ANSI: 0
Oct 17 22:56:28 beryl kernel: scsi 6:0:0:1: Direct-Access Generic USB CF
Reader 1.01 PQ: 0 ANSI: 0
~省略~
Oct 17 22:56:28 beryl kernel: sd 6:0:0:3: Attached scsi generic sg4 type 0
Oct 17 22:56:28 beryl kernel: sd 6:0:0:0: [sdb] Attached SCSI removable disk
Oct 17 22:56:28 beryl kernel: sd 6:0:0:4: Attached scsi generic sg5 type 0
Oct 17 22:56:28 beryl kernel: sd 6:0:0:1: [sdc] Attached SCSI removable disk
```

1番目がSDカードで、初めに割り当てられたデバイス名は sdb です。

# Raspberry Piのインストール

## 3. イメージの書き込み

rootになってddでイメージをコピーします。10分程度はかかるようなので気長に待ちましょう。

```
$ su
Password:
# dd if=2012-09-18-wheezy-raspbian.img of=/dev/sdb
3788800+0 records in
3788800+0 records out
1939865600 bytes (1.9 GB) copied, 763.744 s, 2.5 MB/s
```

これでOSのイメージがSDカードに焼けました。

## Raspberry Piの初期設定

いよいよRaspberry Piを起動させます。SDカード、LANケーブル、キーボード、ディスプレイを接続したら、電源となるmicro USBケーブルを接続しましょう。ちゃんとOSが焼けていればブートが始まります。OS自体の容量も小さめですし、比較的すぐに起動します。

その後CUIのウィンドウが立ち上がり、初期設定が始まります。今回最低限やるべきことはSDカードの容量を全部使い切る設定にすることなので、それだけやって起動します。

キーボードの上下キーで expand\_rootfs を選択し、エンターキーを押してください。これでSDカード全てを使い切るようにパーティションが切られたので、左右キーで移動して一番下の Finish を選択しましょう。Raspbianが起動します。

## Raspbianの環境導入

ようやく環境導入です。インターネットに繋がっていることを確認した後、UVCドライバとWebカメラの撮影を行うfswebcamをインストールします。その前に一般ユーザーを追加したり、特定のユーザーがWebカメラを使えるように設定も行います。

```
raspberrypi login: pi
Password:
$ sudo su
# adduser x68
# passwd x68
# apt-get install v4l2ucp
# apt-get install fswebcam
# vi /etc/group
~省略~
video:x:44:pi,x68
# su - x68
# piユーザーのパスワードはraspberrypi
# root権限に移行
# ユーザー追加
# パスワード追加
# UVCドライバインストール
# fswebcamインストール
# videoグループに対象となるユーザーを
# 追記することで/dev/videoNが使用可能
# 今回は x68 ユーザーを追加
# x68ユーザーに移行
```

問題は次の箇所で、UVC対応Webカメラの接続を行うのですが、上手くデバイス名が表示されません。/var/log/message にも一応UVCで接続したよ！ みたいなメッセージはありますが、どのデバイス名か答えていないので、直に調べます。

```
$ ls /dev/video*
/dev/video0
```

はい。見つかったのでこれを使います（投げやり）。

# WebCamシステムの構築

## Webカメラの撮影テスト

fswebcamコマンドで撮影のテストを行います。videoグループに追加したユーザーであればデバイスにアクセスして撮影が可能です。

```
$ fswebcam /dev/video0 test.jpg
```

写真をWinSCPなどを使ってSSHアクセスしてGUI環境にコピーして確認しておきましょう。

## SSHの鍵準備

scpでファイルをパスワード無しで転送するために、Raspberry Pi側で認証用の鍵を作って公開鍵をWebCam本サーバーに送りつけます。

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/x68/.ssh/id_rsa): # 鍵の名前
Enter passphrase (empty for no passphrase): # パスフレーズ
Enter same passphrase again: # 再入力
```

鍵の名前や位置、パスワードなどは全部デフォルトにします。特に鍵を使うためのパスワードを設定してしまうと、SSH接続時のパスワードは必要無いですが、(鍵を使うための)パスフレーズを入力する必要があり、本末転倒です。

公開鍵の準備ができれば接続先であるWebCam本サーバーに転送するなり、PuTTYなどで公開鍵をコピペできる体制を整えましょう。WebCamのデータを蓄積するユーザーは決まっているはずなのでそのユーザーでログインし、公開鍵を追加します。

```
$ id_rsa.pub >> ~/.ssh/authorized_keys
```

公開鍵はテキストなので、vimなどを使って直接書き込むのもあります。1行1公開鍵となるようにしましょう。また、新規作成する場合はパーミッションを600にしないと正常に動作しないので注意しましょう。

## スクリプト用意

Webカメラで撮影するスクリプトを作成します。作業は次のようになります、

1. fswebcam コマンドで撮影 (tmp.jpgとかにしとく)
2. 日付を元にしたファイル名にリネーム (pic\_20121104.jpgとか)
3. 必要があればリサイズと右下に日付を入れる
4. 公開鍵を使ったscpでWebCam本サーバーに転送
5. ファイルを削除

注意すべきことは、このスクリプトは標準出力、標準エラー出力を出さないように、出力は全て/dev/nullにリダイレクトして破棄します。これにより次の作業で生じるエラーメールやログを残さずに済みます。

さらに、もしストレージに若干の空きがあるのであれば、5の手順は4の失敗時にファイルの破棄を行わないのと、1の手順の前にストレージの空き容量と写真の大体のサイズを考慮して古過ぎるファイルを破棄することで、本WebCamサーバーが落ちてても動作します (scpの転送をワイルドカード指定にすればファイル転送もそんなに考えなくて良い)。

## crontabの用意

次にcrontabに例の скриプトを登録します。ただし、動作前にSSHで一度本サーバーに接続しておきます（初回接続時に初接続のホスト情報を登録する必要があり、キー入力が発生する）。キー入力なしでSSHが通り、スクリプトがキー入力なしで動作することを確認し次第、crontabでの実行を確認しましょう。

## まとめ

今回はRaspberry Piを使って遠隔地用のWebCamシステムを作りました。Raspberry Piは最初のインストールこそ特殊で少し環境を整えたりするのが大変です。軽く調べたところ日本語版のWindowsなどではOSイメージを焼くのにだいぶ苦労するとか。しかしそれさえ超えてしまえばあっさり動きますし、動作も軽いです。さらにSDカードの切替だけでOSを入れ替えることができますし、これから拡張用のパーツなども販売されるとのことなので、まだまだ遊ぶ余地のある面白いハードだと思います。

WebCamSystemの構築に関しては前回の構築に引き続き、Linuxの基礎知識とUVCさえあればさほど苦労せず構築できるものです。今回の目新しさはやはりネットワークを使った分散を視野に入れたことでしょう。それも小型で安価なRaspberry Piが出てきたおかげです。実際にRaspberry Piをたくさん集めてクラウドのように構築した例もあるように、低スペックでも小型で安価、Linuxが動くというようなハードは、個人レベルでの分散システムの開発などに役立ちます（クラウド系の分散は多少ネットワークが貧弱でも成り立つ）。分散ストレージなどの構築は少しむずかしいかもしれませんが、今回の監視カメラの分散等は容易に可能です。あ、今回のシステム（Raspberry Pi+Webカメラ+SDカード）の値段は合わせて約6000円ほどです。6000円くらいで気軽にWebCamSystemを増設可能です。こういう遊びが気軽にできる試みは非常に良いと思うので、今後Raspberry Piのような端末が増えるとIT系の人間にとっては嬉しい限りです。



## おまけ

実はネタ的に美味しいということでキーボード付きAndroidタブレットのTF201(ASUS)にてOSイメージを焼いていました。キーボードドック側のSDカードスロット、USBポートにメモリなどを挿すと外部ストレージとして認識されるので、root化したTerminalIDEというアプリを使って作業可能です。しかしなぜかddがIOエラーで失敗しました。何かddコマンドやストレージ周りに得体のしれない不具合が眠っていそうです。