« Stoutette » project

Version 1, march 2013

Stout [at] meoz.fr

Playing with my Raspberry pi, discovering GPIO functions, this project came to me.

Index

Goals:

Initial goals :

While traveling abroad, I wasn't satisfied with the way I dealt with my camera / its pictures, spending more time than I need in cyberspaces, to be able to send them to friends, or to feed a blog, …
I usually don't bring any laptop or notebook, firstly to prevent me from getting stuck on it instead of enjoying the country (I know I could spend many hours playing with my pictures, and I don't even imagine if the Internet is available !); but also because I like to travel light, in countries where computers are high priced device that I don't want to carry. (my camera is in fact the only high tech stuff I tolerate).
Briefly : When traveling, i like to leave the technology stuff at home.

So I needed :

- **a device which could resize pictures autonomously** :
  traveling around, I spent plenty of time in cyberspace trying to resize picture to be able to send them to friends ,via email or blog, through low bandwidth networks.
  The device should be able to to that without any other help, in a standard guest house room. (where ac power is available)

- **a device which could copy files from USB device to other USB device** :
  traveling around, I got my camera stolen, included of course its SD card full of memories.
  The device should be able to copy the files, for backup purpose, in a standard guest house room.

I figured that the raspberry pi could do that job : it's a small low power computer, quite cheap, which seems solid (no mechanicals parts). I used it as a media center with Raspbmc, and wanted to play a little bit more with it, especially with the GPIO features.

Looking for devices doing that job, I found pretty expensive stuff, and most of which do not support the resize function:
http://www.bhphotovideo.com/c/buy/Portable-Data-Storage/ci/3369/N/4083163867

Plus the DIY way was very challenging !
(I should precise I'm a computer enthusiast, OK, but it's not my job at all.
The most advanced programming stuff I did was some PHP to customize a Wordpress website, and some scripts in basic when I was younger.
I recently built my own server at home, including mail server and web server, on a netbook running a Debian)

This were the initial functions and goal to be achieved.

Other functions added :

But along the way, other ideas came, implying more software development but no other hardware :

- **A device to make the publication process of photos to a blog much more easy**
  "2 buttons pressed" once the camera is plugged, et voilà, pictures are an the blog"
  The photo blog would of course be hosted on my own web server (an atom notebook running a Debian)

- **A device to make the process of backup easier**
  "plug the backup drive, press 1 button, that's all",
  to backup some files from my NAS to external hard drive that I can store in different places.
  I plan to do this process monthly, an I don't want to spend much time remembering RSYNC commands, sources, destination which are networked mounted drives, or whatsoever.

Many other ideas may pop up : a device running Linux, with a small screen and few buttons can do many many things !!!

Basic Hardware :

So I needed to build a physical interface, with buttons and a small screen to be able to launch severals commands on the raspberry without any other computer, just a power plug.

Many resources are available to build that :

For the screen, a cheap Hitachi LCD seemed easy to connect to the Pi :

http://www.zem.fr/raspberry-pi-et-afficheur-lcd-hitachi-hd44780-1602/ (french !)
http://www.raspberrypi-spy.co.uk/2012/08/16x2-lcd-module-control-with-backlight-switch/
https://projects.drogon.net/raspberry-pi/wiringpi/lcd-library/

Buttons seemed also easy to connect :

http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/buttons_and_switches/

I bought on ebay the things I needed, buying more than I needed for future projects !:

push buttons : 10 pcs for 4,40 €

misc resistors 5 €
solid proto boards 10 for 5,10 €

(bought at LED megashop store : http://stores.ebay.fr/LED-Megashop)

Arduino Breadboard Jumper Cable Wires (65-Cable Pack) : 4 €
A breadboard for testing : 2, 66 €
2 HD44780 LCD screens (2x 16 characters) : 6€

I already got a solder iron

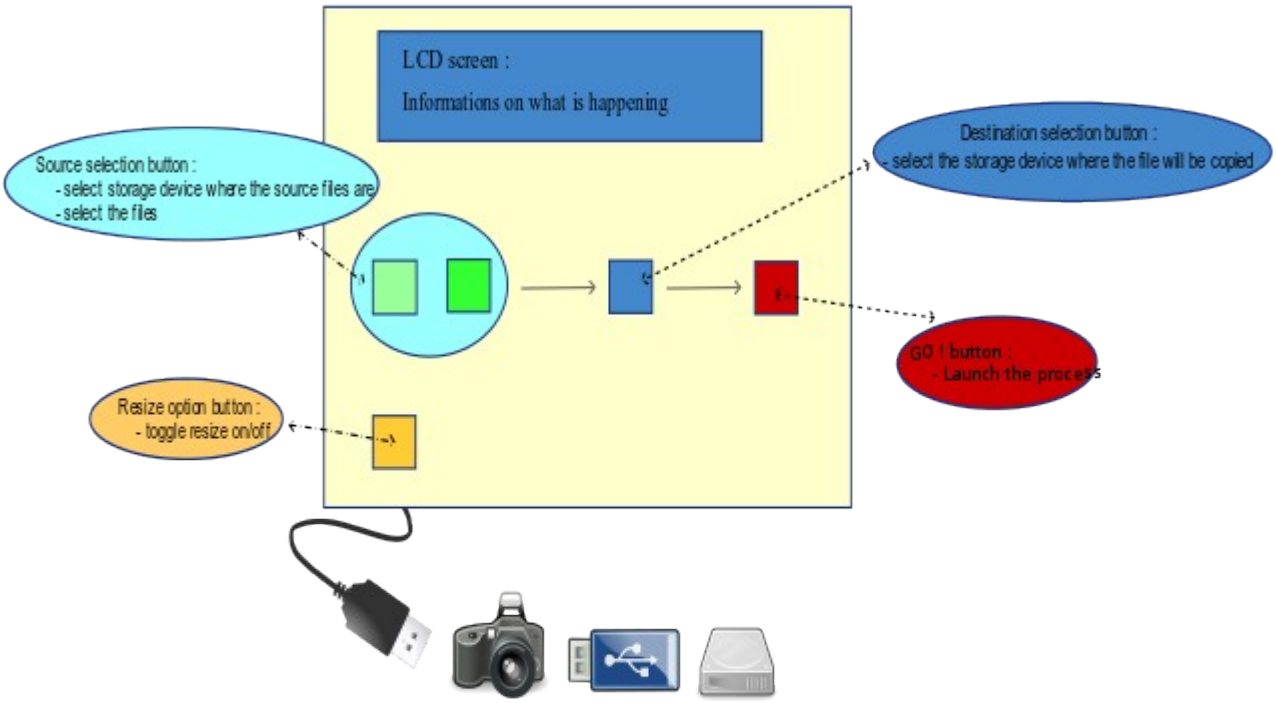
So it's almost 30 € for way more than I need : 2 screens, the breadboard is for early prototype and is a long term use stuff, many solid boards (10 instead of 1 if everything goes well) …

Prototyping :

The main concern for me was to make the device SIMPLE to use :
once correctly configured (built !), just a few button press, and it's OK.

To achieve that, the device should recognize devices plugged so that to offer by default the good function with the right options, and the button pressing is for confirming. (or changing them)

So how do I imagine the device, called "stoutette" ?



Some scenarios :

1) While traveling, no network available :

- I plug my camera and a USB key :
  Obviously, I want to copy pictures from my camera to the USB key.
  So by default, I just want a confirmation of that on the display :
  " camera → USB"
  and if I press the GO! Button, the copy start.

- What if I only want *some* files to be copied ?
  - The files are preselected on the camera : PictBridge function on the camera.
  - I have to select them with the device, by date : file from the XX lasts days/hours
    so I have to be able to input a number of days / hours, by pressing some buttons.

  In both cases, I can toggle the Resize switch :
  - OFF : no resize, copy only original files
  - BOTH : copy both resize and original files
  - ONLY : copy only resized files.

- What if I plug 2 USB keys, to backup some files from one to another ?
  The device cannot predict what I want to do !
  So I have to manually choose the source by pressing the source button, the files with the date selection button, and the destination with the destination button. Then go !

I do not plan to do more sophisticated file selection than
"by date"="file from the last xx days/hours"
because the way I see it,it's the only useful option : I want to backup the photos from today, or from 4 days ago, ….

The PictBridge option is very handy : from my camera, I can very simply (with visualization of the photo) select the photos I want to copy or resize.

2) At home, the devices is connected to a network

- I plug my camera only
  The default behavior would be to send the selected pictures (Pictbridge) to my blog, and publish them automatically, without any other action from me than pressing the GO! Button. The display has to tell me that before I press the button !

- I plug an external drive used for my music folder backup :
  Usually, I plug the drive on my computer, mount it, rsync it …
  Here, I want the device to recognize this specific backup drive (via UUID or label or whatever), so that it displays : " music backup ?", and if I press the GO! button, bingo, the backup starts.

Those are the main basic scenarios I built.
Of course, many software details are to be discussed, especially for the blog publishing scenario.

Final hardware :

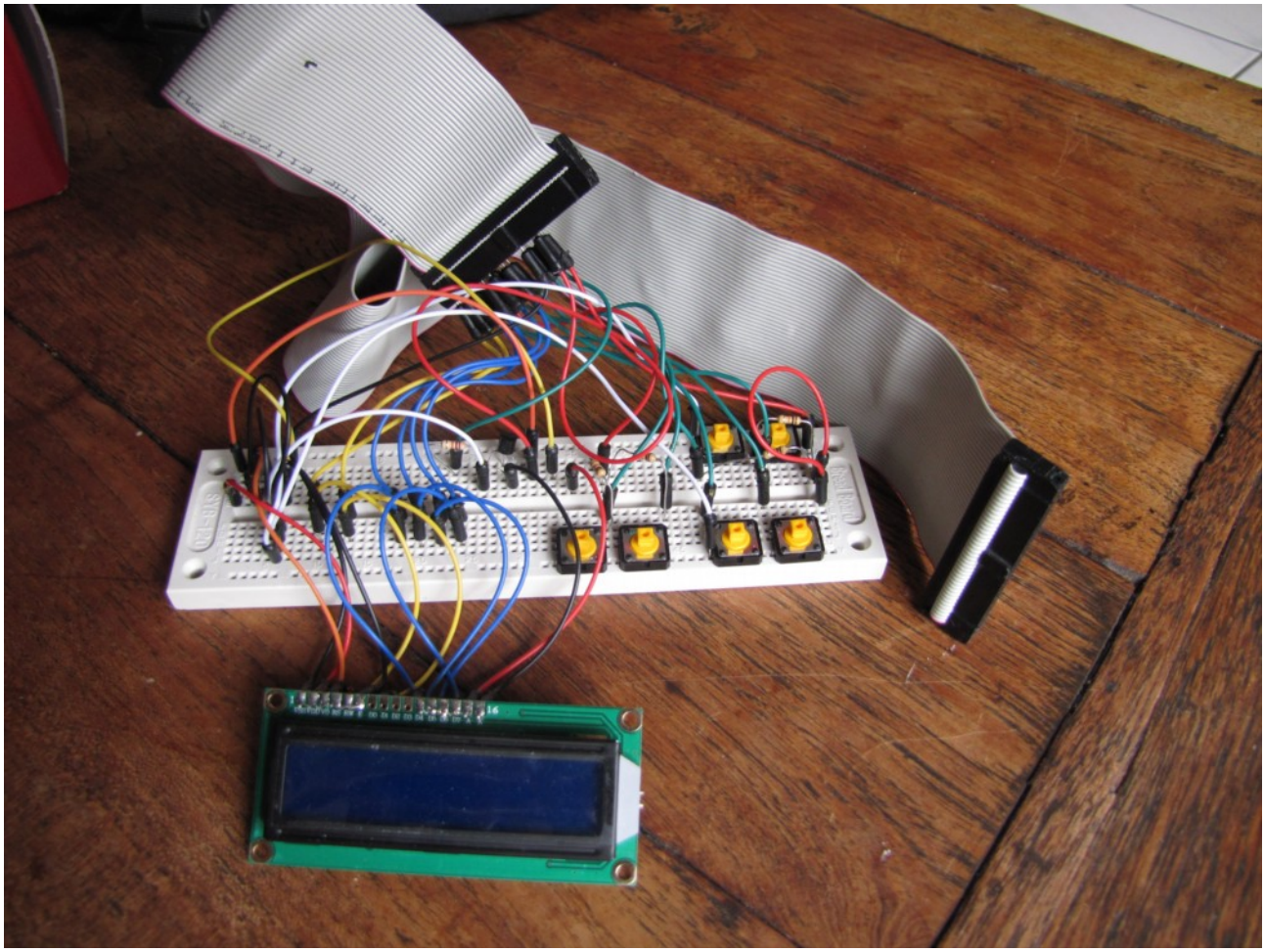
For the hardware, this is it :
I decided to use 6 buttons and the LCD :

- 1 “source selection” : when pressed, cycle through the available sources
- 2 “date selection” : short press add or subtract days, long press add/subtract hours
- 1 “destination selection” : when pressed, cycle through the available destinations
- 1 “go !” button : short press : display the action, long press : do the action
- 1 “resize” toggle button.

Not all the possibilities are used : some “long press” buttons are not used, and no combination of button are used. These can be used for debugging / advanced configuration purpose later.

Construction of the device :

Using the previously mentioned sources, I built a prototype to test the LCD and the button switching function.
As it was my first experiments with the GPIO, I played with the breadboard a lot before making it worked !



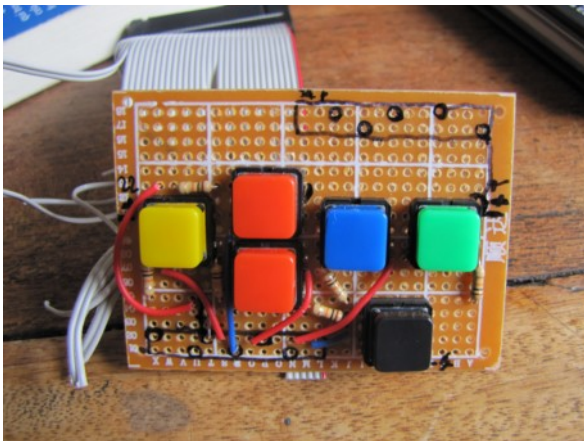
To connect to the Pi GPIO, I used an old IDE cable.

After several tries, it worked ! :

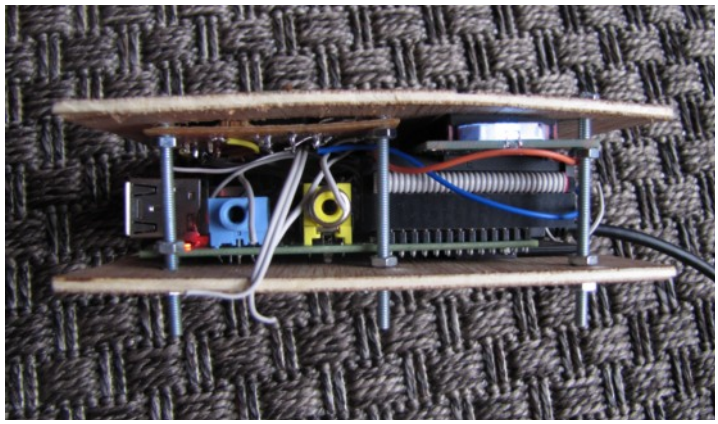
The LCD can display text through bash command lines, with the python librairies.
The button press are detected, and initialized in bash script only, using the Drogon GPIO tools.
(see later for more explanations about thoses)

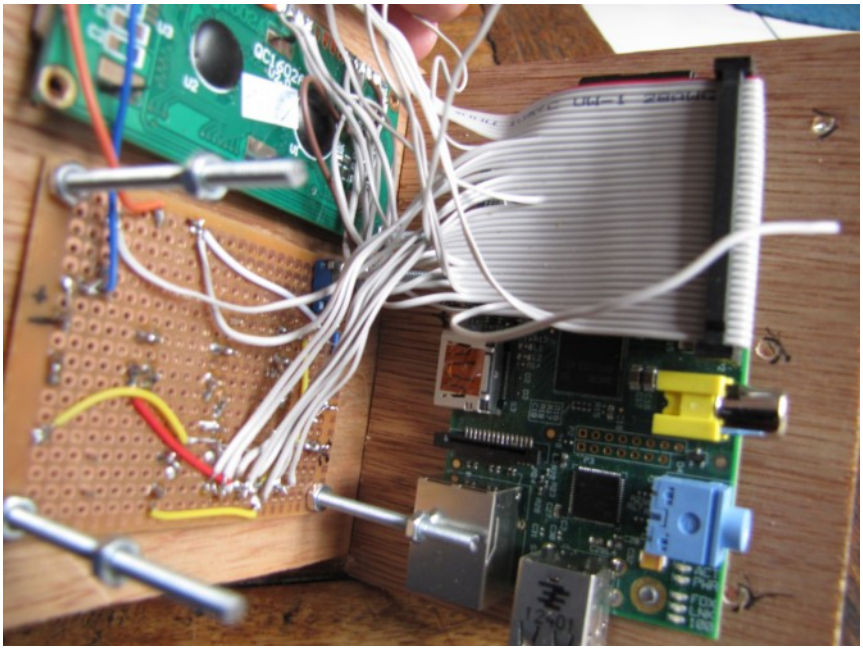
I intend to use only bash script do build the entire program.
I do not know any programing language, I start to become familiar with the shell, so this seemed natural for me to start with bash scripting.
Plus I knew that the functions I need : copy files, detect USB devices, …, are possible to do with bash.

Once I built the first prototype and saw made it basically worked, I built a more “solid” one, so that it looked like something real !



The solid buttons board





You can see some hanging wires from the IDE cable : they correspond to unused GPIO pins.

I haven’t cut them yet because they may become useful if I want to add other stuff : buttons, LED, …

The software :

As I already told, I am not at all a programmer !
I discovered many many things writing this piece of software, which is certainly not “clean” neither “optimized”...
I tried to comment a lot of the code, feel free to ask for some explanation / comment, ...

I won’t detail everything, just a few points that may be interesting for other people:

LCD/Buttons interfacing
- send text to LCD
- long press button detection
- button combination

USB detection
- camera problem : not a mass storage unit !
- specific device detection (backup drives)

Sources/Destination selection process.
- PictBridge
- Date problem

Blog publishing specifics :
- autogenerated static site
- commanded by email

LCD interfacing :

(script 'action_arg2LCD' in the folder 'python')

For the hardware connection, I used the schemes found here :
http://www.raspberrypi-spy.co.uk/2012/08/16x2-lcd-module-control-with-backlight-switch/

I used the LCD python library to send text
https://code.google.com/p/raspberry-gpio-python/
with a python script which get the lines I want to display on the LCD from the command line arguments : first argument is the first line, 2nd is the second line

Buttons interfacing :

(script 'veillehard')

For the hardware connection, see here :
http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/buttons_and_switches/

To detect the buttons states in a bash script, I used the Gordons GPIO utility
https://projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/

The script for this is in fact an infinite loop which check continuously the state of the pins, and launches command when the states change, when the button is pushed.

To be able to detect long press, I wrote a while loop with a counter, increasing while the pin state is down, until a predefined number, when reached, it breaks this loop, otherwise, it's a short press.

For button combinations, for instance button A + B , I just included the pin detection of B just after the one of A : if A is pressed down, it checks if B is pressed down; if B is pressed down, it checks if A is pressed down.

USB detection

File '10-automount.rules in the folder /etc/udec/rules.d
script 'usb-plug'

I used the UDEV service to detect USB plugging/unplugging, and do launch the appropriate stuff. Many tries were needed to find the right syntax, I found some useful resources on the Internet looking for 'UDEV rule writing", "write udev automount rule", …)

- generic USB devices ares mounted in /media/LABEL, label stands for the drive label.
- specific USB devices such as backup drives are identified with their UUID , idVendor, idProduct and mounted in /mnt/SPECIFICNAME
- my camera cannot be mounted like that : it's not a Mass Storage Unit, but uses the Picture Transfer Protocol PTP..
  Hopefully, Gphoto2 and especially gphotofs enable it to be mounted as a standard drive !
  http://www.gphoto.org/
  http://manpages.ubuntu.com/manpages/hardy/man1/gphotofs.1.html
  The package gphotofs needs to be installed

Source and destination definition :

script defsrc and defdest

USB devices

I used bash arrays for those, filled with the directories found in /media or /mnt
a press on the source or destination button select the next one in the array.
If a specifique USB device is present, it's the 1st one in the array => minimise the number of button pushing.

- Network destinations
  If network is up, detected with ifplugd // A DETAILLER , some network destinations are inserted in the destination array : the blog, some emails...

File Selection :

PictBridge :

To be enable to choose pictures directly on my camera, with visualization, I used the Pictbridge function of my camera : designed to send pictures to a printer, it creates a text file with the selected pictures, easily parsable !
This file "AUTPRINT.MRK" is in the folder MOUNTDIR/store_00010001/MISC/ for my camera. I didn't check if it changes with other ones...
I guess the file name won't change, maybe it's location, that could be managed with a 'find' command.

Date problem :

When no PictBridge is available, let's say I want to copy some files from a USB stick, I need to be able to select *some* files from the source.

I decided to make a time selection, because I think I will only need this kind :
"I want to copy the files from last week until now".
Problem is : the Raspberry Pi doesn't have any battery to remember the date !!
It is not on time if it's not connected to a network where it can synchronize via ntp...

One way to work that around will answer that need :
"I want to copy the files one week older than the newest file on the key"
Because I think that will be sufficient : if it's from a camera, I just have to take a picture just before plugging the camera, the date will be approximately OK.
So if no network is available, the script finds the newest file on the media, and sets the date of the system to it.
The the user defines a lap of time, in days and hours, with the buttons, the sources files are the ones found with their modification date older than (SYSTEMDATE–USERLAP) (command 'find -mmin')

Blog specific :

I wanted to really minimize the process of publishing pictures on a “photo blog”.

My goal is :

- select the pictures on the camera (pictbridge)
- plug the camera to the device
- press a few buttons to validate the action
- that’s all : the website is updated automatically...

To achieve that, the process consists in :

- an autogenerated website :
  it looks for pictures in specific folder, and generate a nice photo website
  a shell command line should be able to tell it to rebuild itself

- a way to send picture from the device to the website server :
  the device takes the pictures from the camera, and send them via network to the appropriate folder in which the autogeneration website script looks to build itself

- a way to send commands to the website when some photos are sent.

For sending picture AND commands, I chose a very standard way, accessible from almost everywhere : email.
The device sends selected pictures in email to the web server; at a specific email address.
On the web server, a script is launched when an email is received at this address :
it analyses the email, and, depending on its content, do several actions :

- if photos are attached, it creates a folder with them in the website folder, then launch the command to regenerate the site.
- If some special commands are written in it, it launch specifics command : add title, description, …
- if the mail is garbage, it does nothing.

Many tools exists to achieve that.

I chose :

- for the autogenerated photo website : SIGAL http://sigal.saimon.org/en/latest/
  written in python, easily customizable to my needs
  It generates a nice gallery, + enables description from a textfile in MarkDown syntax.

- for the mail sending : sendemail http://caspian.dotconf.net/menu/Software/SendEmail/
  send SMTP email with attachment from command line

- for email receving / script launching :
  I use incron http://inotify.aiken.cz/?section=incron&page=why&lang=en to watch for the maildir folder and to launch a script when a new file is created in the ’new’ folder.
  I plan to use directly postfix, which can do that as I was recently told: http://blog.thecodingmachine.com/fr/content/triggering-php-script-when-your-postfix-server-receives-mail

For email analysing :

- First, ripmime http://www.pldaniels.com/ripmime/ to decode the mail with attachment
- Then, a home made script using mostly 'sed' to extract some keywords and do specific actions if they are present.
Script 'newmailscript'

Main scenarios :

Album creation

1) an email is received on my mail server with some photos attached, and the subject corresponds to the predefined string saying "I am a good email to analyze"
2) the scripts is launched decoding it in a plaintext file containing its text content, and the pictures
2) the textfile is analyzed :
    1. subject string is recognized
    2. a folder named with the date of the email is created in the website folder, where attached photos are copied
    3. if the textfiles contains informations such as Title or Description, they are added to the index file in the same folder, index file analyzed by SIGAL.
3) SIGAL command to regenerate the website is launched/

But how to modify an existing album ?

If the email subject contains a specific string including the allbum ID=its folder name=date of reception of the first email creating it, which is unique, then the script do not create a new folder, but modifies the existing one.
How to know the existing album ID ? That's why I chose the date to name the folder : it's unique, AND it can be displayed as an information in the blog.
So to modify an album, I just copy it's date which is displayed on the blog, and include it in the subject of the email I send, with a specific string...

Many commands can be sent that way to specifics album : to delete them for instance.

What about security ?

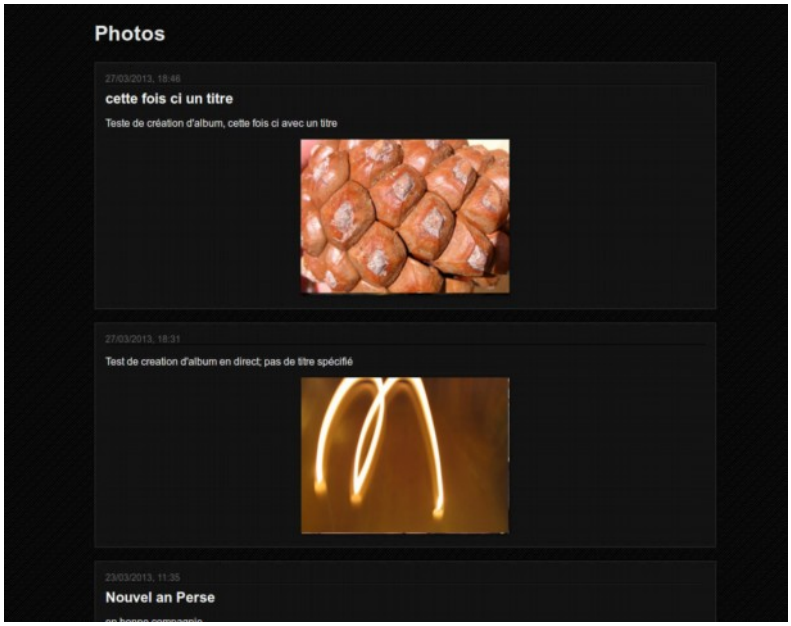
This system is very "open", and not very secured : anyone can send emails which could be interpreted and run commands...
BUT to do that, first one need the good email address, which can be kept secret, then the right syntax in the subject / content of the mail.
Plus it is possible to include a specific secret string in the mail subject / content to be more secure.
Only minor damages can be done : in worst case scenarios, ugly photo albums are added, or some are deleted, or their description is changed...
Considering my photo blog is not made to be popular but to share picture with a restricted public : family / friends, this should not be a problem.

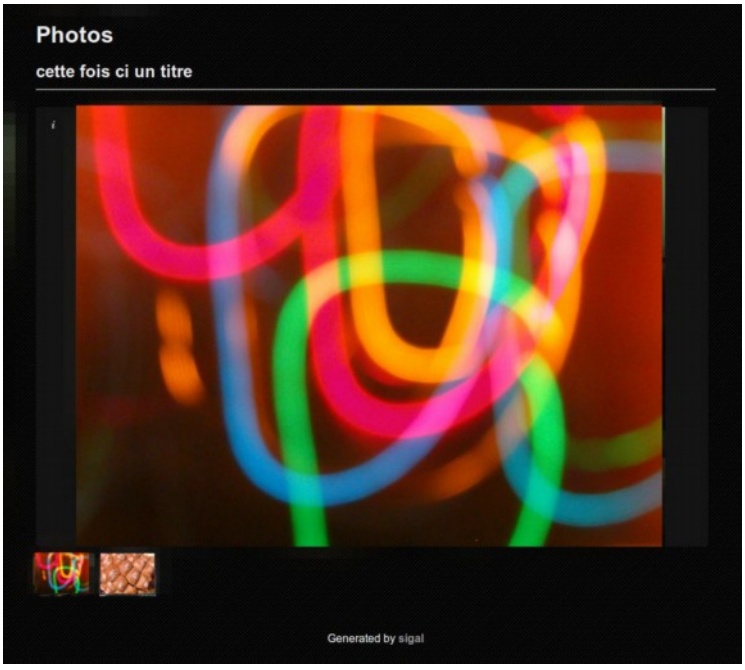
Example :

Main page :

Each post links to an album
The photo displayed is one of the photo from the album

Notice the date, it is the unique ID of the album (corresponding to its folder name), to use in the email to modify, add pictures, change title or description of the album





Album page

(photos taken from http://www.freephotobank.org)

Current development :

It’s almost finished !

A few cleanup is necessary, but the main parts are OK.
Some finishing touch are to be made : clean startup and poweroff with button combination, XBMC toggle with button combination too, …
The box is to be finished, it’s quite rough now !!

A few problems came up though :

The main one : the backup process is VERY long.
Rsync a folder between my network NAS(100mb/s LAN + smb share) ↔ stoutette ↔ usb drive (in ext4) is very very slow, less than 1Mo/s .
I don’t know where is the bottleneck : USB ?
The CPU load is ok on the raspberry...