

JOUONS UN PEU AVEC UNE CAMÉRA IP D'ENTRÉE DE GAMME SOUS GNU/LINUX

par Denis Bodor

Au secours Obi-Wan Kenobi, vous êtes mon seul espoir... Difficile de ne pas voir en cette sympathique petite caméra IP des airs de princesse Leia ou de princesse Vespa, selon les références cinématographiques de chacun. Quoi qu'il en soit, le profil de ce périphérique est très courant et son contenu tout autant. Vendu aussi bien sur eBay directement depuis la Chine que via des détaillants de matériels informatiques de toutes réputations, il s'agit là de périphériques portant bien des noms différents mais partageant, parmi d'autres choses, un élément commun : le prix, bien en dessous de la moyenne du marché.

Dans ce domaine, nous sommes en effet très loin du coût des périphériques vendus par Cisco, Linksys ou Axis. Bien entendu, la qualité est également sensiblement inférieure, en particulier en termes de qualité d'image. Le rapport fonctionnalités/prix en revanche est tout à fait respectable et ce type de matériel (parfois moins de 40 euros) fera une solution de surveillance vidéo acceptable.

Une précision importante concerne ce matériel ou plus exactement cette "famille" de matériel. Ce périphérique existe, nous l'avons dit, sous plusieurs noms et en plusieurs déclinaisons. Ainsi, si l'implémentation de référence est issue du fabricant Foscam sous les noms FI8918W et FI8908W, un grand nombre de clones (et le mot est faible) sont fabriqués et vendus un peu partout. Les points communs sont généralement les suivants :

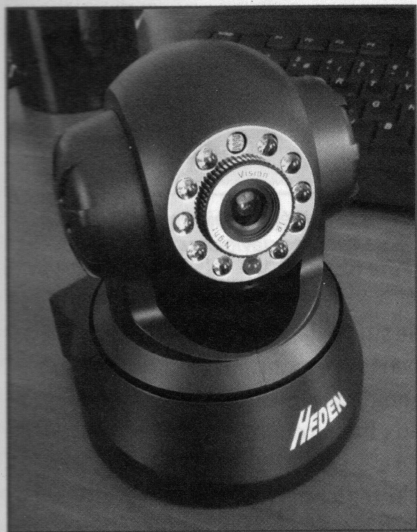
- SoC ARM (Winbond/Nuvoton) ;
- motorisation pan (gauche/droite) et tilt (haut/bas) et parfois zoom ;
- vision nocturne (en réalité le capteur CMOS ne semble simplement pas avoir de filtre IR) ;
- interface ethernet parfois accompagnée d'un adaptateur Wi-Fi ;
- et firmware basé sur Linux.

Les variations enfin portent sur des éléments matériels qui impactent généralement le coût de revient, comme :

- la taille et le type de la mémoire vive ;
- la taille et le type de flash ;
- le firmware plus ou moins récent, bogué et complet ;
- les fonctionnalités annexes (micro, LED IR, etc.) ;
- et en particulier la qualité générale des composants.

Si vous achetez ce type de matériel à un vendeur en Asie (via eBay, DealExtreme, Lightinthebox, etc.), votre principal problème sera de savoir précisément ce que vous avez reçu et dans quelle mesure il sera risqué de faire une mise à jour du *firmware* ou de l'interface web. Nombreux sont les utilisateurs de tous pays ayant acheté ce type de matériel, des clones de Foscam, et ayant littéralement "brické" leur caméra sans vraiment comprendre pourquoi. En effet, les fonctionnalités, l'interface web et l'aspect général du matériel ne sont en rien révélateurs d'une quelconque compatibilité. Les amateurs de systèmes embarqués le savent bien, contrairement à votre PC qui est une configuration générique, un tel système est très proche du matériel. Et cela commence avec son initialisation. Alors que sur un PC, le détesté BIOS, intégré à la carte mère, se charge de l'initialisation et des tests de démarrage (POST pour *Power On Self Tests*),

sur un système embarqué, cette tâche incombe au *bootloader* et aux premières fonctions de démarrage du noyau de l'OS. Le simple remplacement d'une puce de mémoire vive ou de flash et les subtiles différences qu'il sous-entend peuvent rendre le démarrage d'une version du firmware pour un modèle impossible alors que 99% des autres composants sont parfaitement identiques.



La caméra Heden ressemble comme deux gouttes d'eau aux modèles Foscam ainsi qu'aux autres clones, avec leur petit air sympa de princesse Vespa avec son walk-droid.

Les clones, moins chers que la Foscam originale (de parfois 50%), intègrent généralement des éléments moins coûteux à produire et à intégrer, qui rendent ainsi l'installation du firmware Foscam tout bonnement impossible. Ne jouez donc pas à l'apprenti sorcier sans en mesurer pleinement les risques et ce qu'il vous coûtera de corriger le problème. Dans le meilleur des cas, l'ouverture du périphérique et la connexion d'un adaptateur USB/série et dans le pire le recyclage de votre caméra IP en presse-papier SpaceBalls. Vous voilà prévenu !

La légalité de ces clones en termes de propriété industrielle, bien qu'elle ne nous concerne pas directement, est un sujet qui nous semble un peu flou. Le site de Foscam a déjà explicitement averti ses utilisateurs de l'existence de "copies illégales" de ses caméras et de procédures

en cours à l'encontre de certains distributeurs (sur eBay mais également DealExtreme). D'après les discussions sur les sites et forums d'utilisateurs, il semblerait que la qualité d'image de ses clones soit directement proportionnelle au prix du périphérique.

1 Le matériel utilisé et sa prise en main

Voyez cela comme un goût prononcé pour le risque ou pour un signe d'avarice, mais le matériel utilisé pour cet article n'est pas un modèle Foscam original. Il s'agit d'un clone, initialement vendu par LDLC sous le nom de Heden Cam. Pour information, la marque Heden est celle de la société PCA France basée à Noisy le Grand qui "œuvre" dans le domaine informatique depuis plus de 10 ans. Le nom Heden évoque peut-être quelque chose à certains lecteurs en raison de démêlés judiciaires il y a quelques années avec nos confrères de *Canard PC* suite à un article sur les alimentations *noname* premier prix. Quoi qu'il en soit, il ne s'agit pas ici de soumettre le matériel à un test ou des essais, et pour nous, la qualité de ce type de matériel a plus à faire avec sa "hackabilité" que son utilisation en production. Gardez à l'esprit qu'on n'a rien sans rien. Si vous cherchez une caméra IP capable de détecter un chat noir à 3 km lors d'une nuit pluvieuse sans lune, ne comptez pas vous en tirer avec un budget de quelques dizaines d'euros et vous hésitez sans doute à vous en approcher équipé d'un tourne-vis...

À présent que les choses sont dites, penchons-nous sur le matériel. La caméra Heden se décline en deux modèles : Wi-Fi ou pas. Nous avons opté pour la version Ethernet simple, mais comme nous le verrons plus tard, la déclinaison Wi-Fi se limite à la présence d'un simple adaptateur USB intégré. La caméra IP Heden dispose d'une connexion intéressante :

- alimentation DC 5V ;
- port Ethernet ;
- connecteur audio ;
- et un connecteur 4 broches libellé "I/O Alarm".

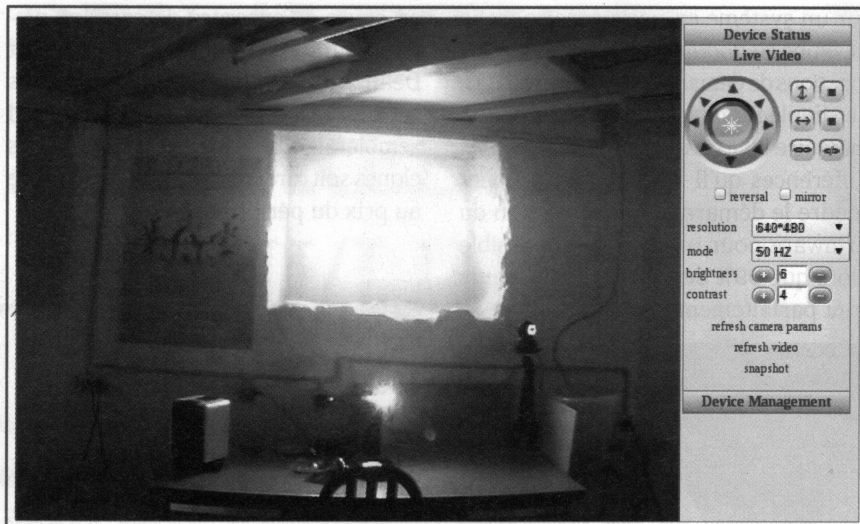
L'optique montée sur deux axes motorisés offre une liberté appréciable de 240° horizontalement et 90° verticalement. La mise au point est manuelle et l'optique est entourée d'un ensemble de 10 LED infra-rouge 850nm permettant la vision nocturne. Le déclenchement des LED est contrôlé par un capteur de lumière (une LDR) et leur fonctionnement est indépendant du firmware. La LED verte placée sous l'optique, en revanche, est contrôlable. Pour information, les quelques exemplaires en notre possession sont tous référencés CAMHED02IP. Il s'agit, semble-t-il, d'un modèle relativement ancien qui n'est plus référencé en tant que tel sur le site d'Heden. On peut cependant estimer sans trop prendre de risque que toutes les explications concernant l'utilisation du matériel sont applicables aux nouveaux modèles ainsi qu'aux autres clones. Le protocole de commandes via CGI, par exemple, fonctionne en grande partie avec nos caméras Heden alors qu'il est initialement donné pour les modèles Foscam FI8908W et FI8918W.

Concernant la première connexion et alors que la documentation recommande l'utilisation d'un outil Windows, il s'avère, et c'est une bonne surprise, que l'adresse IP est directement obtenue via DHCP. Beaucoup d'*appliances* réseau arrivent par défaut avec une adresse fixe et il faut généralement une connexion directe pour changer la configuration. Là, ce n'est pas le cas. Ne faisant pas grands faits du manuel, jeté dans un coin avec l'emballage, on découvre alors, en pointant un navigateur web sur l'IP de la caméra, une interface laide comme il se doit mais offrant la possibilité de choisir entre l'interface "ActiveX Mode" et "Server Push Mode" adaptée à un navigateur plus "normal" que celui de Microsoft.

L'interface de configuration, sous la désignation "Device Management", permet entre autres choses :

- La configuration de l'horloge (RTC).
- La création/modification de huit utilisateurs pouvant être "visitor" (aucune action), "operator" (contrôle de la caméra) ou "administrator" (configuration). Chaque utilisateur peut avoir l'identifiant qu'il souhaite et nous ne sommes pas obligés de conserver le classique "admin".
- Le choix du port pour le serveur HTTP embarqué (80 par défaut).
- La configuration Wi-Fi avec scans des AP en présence avec un niveau de sécurité entre "none" et "WPA2 TKIP".
- La possibilité d'utiliser directement un modem ADSL via PPPoE (un peu désuet actuellement).
- Le choix de lancer ou non un serveur UPNP.
- Un support pour DDNS (DNS dynamique) pour retrouver la caméra sur une connexion avec une adresse IP attribuée dynamiquement.
- Une possibilité pour envoyer des mails d'alerte à jusqu'à 4 destinataires via un serveur SMTP (*smart-host*) configurable.
- La configuration d'un client FTP pour y stocker les captures automatiquement.
- La configuration des alarmes (détection de mouvement et/ou activation via le port "I/O Alarm").
- La gestion de la mise à jour du firmware.
- La sauvegarde et restauration des paramètres de configuration (téléchargement d'un fichier texte).
- L'accès à un journal de connexion à la caméra.

Les fonctions "Live Video", quant à elles, sont celles vers où se porte toute notre attention. L'interface web présente un module de contrôle de mouvements haut/bas/gauche/droite/diagonales ainsi qu'un mode patrouille horizontal et vertical.



L'interface proposée par le serveur HTTP intégré à la caméra IP, bien que totalement affreuse, reste pratique et surtout compatible avec les navigateurs comme Chromium ou Firefox.

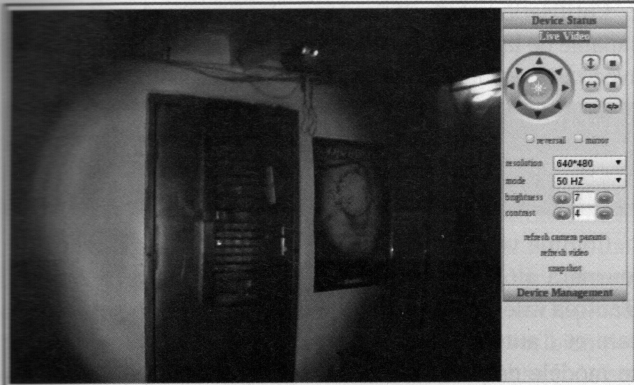
Deux étranges boutons permettent d'activer un relais dont la connectique se trouve sur le port "I/O Alarm". Cette fonctionnalité intéressante ouvre des possibilités dans le contrôle, par exemple, d'un système d'éclairage. Aucune indication du constructeur, cependant, ne précise les limites en termes de tension et de courant pour ce relais (à utiliser donc de manière directe avec prudence).

L'image provenant de la caméra peut être affichée en 640×480 ou en 320×240 avec un contrôle de contraste et de luminosité. Un système de gestion de l'éclairage est également utilisable afin de réduire les interférences des ampoules à filament en choisissant un mode naturel (*outdoor*) ou une alternance de 50Hz ou 60Hz. Enfin, un bouton permet d'afficher une capture fixe de la vision par la caméra, en JPEG.

Une inspection de la page nous indique que le codec vidéo utilisé n'est autre que du MJPEG ou Motion JPEG à l'URL <http://user:pass@IP/videoStream.cgi>. Quelques essais avec le client VideoLan montrent une accessibilité aisée sur flux vidéo, tout comme avec FFmpeg ou Mplayer.

La qualité d'image n'est pas extraordinaire mais suffisante pour un usage

de surveillance simple. Les couleurs affichées ne sont pas très fidèles en raison de la sensibilité du capteur CMOS aux infra-rouges mais les capacités de vision nocturne offrent des fonctionnalités compensatoires. En l'absence de lumière visible, les huit LED IR fournissent un éclairage ("rayonnement" de vrais-je dire puisqu'il ne s'agit pas de lumière visible) correct sur deux ou trois mètres. Le déclenchement d'alarmes par mouvement fonctionne correctement sans être d'une sensibilité exemplaire. Cette fonctionnalité a tendance à jouer des tours puisque d'une part le capteur semble tantôt se réinitialiser ou se recalibrer (fondu au blanc puis retour) et d'autre part les LED IR semblent de piètre qualité, tout comme la LDR (deux des six caméras en place éclairent faiblement et une troisième plus du tout, après quelques centaines d'heures de fonctionnement). Des variations de sensibilité et de rayonnement IR sont détectées comme des mouvements qui déclenchent des alertes. Les premiers mails reçus en pleine nuit incluant des captures ne présentant rien de particulier ne sont pas sans rappeler "Paranormal activity", mais après analyse, nos locaux ne semblent pas être hantés par une entité démoniaque et maléfique autre que votre serveur.



La caméra dans le noir. Attention, ceci ne reflète pas la qualité du système d'éclairage infra-rouge intégré. En effet, le "spot" lumineux provient ici d'une LED IR 850nm de 100mA comportant 5 "dies" en guise d'éclairage complémentaire.

Dans les grandes lignes, nous avons là un équipement tout à fait acceptable en rapport avec son prix et surtout compatible avec des navigateurs open source ainsi que d'autres utilitaires, ce qui n'est pas toujours le cas des périphériques de marque réputée.

2 Aller plus loin et oublier l'interface web

Au-delà de l'utilisation classique avec un navigateur, ce type de matériel permet une utilisation plus intégrée. En effet, alors que le flux vidéo est récupérable via une URL, il en va de même pour un certain nombre d'actions de contrôle et de configuration. Le firmware intégré peut recevoir des ordres en accédant directement à des URL spécifiques provoquant l'exécution de scripts CGI.

La syntaxe est relativement simple. Ainsi, pour le flux vidéo, il nous suffit de pointer **vlc**, par exemple, sur `http://user:pass@IP:port/videostream.cgi`. Notez que deux syntaxes sont utilisables pour passer le nom d'utilisateur et le mot de passe. La seconde est `http://IP:port/videostream.cgi?user=utilisateur&pwd=mot2passe`. D'autres paramètres peuvent être spécifiés, comme la résolution avec **resolution=** suivi de **8** pour 320×240 ou **32** pour 640×480, mais aussi le **framerate** entre le maximum (**rate=0**) et une image toutes les 5 secondes (**rate=23**). À noter qu'il existe également un flux ASF accessible via **videostream.asf** reposant sur les codecs suivants (dixit **ffmpeg**) :

```
Stream #0.0: Video: mjpeg, yuvj422p, 640x480, 25 tbr, 1k tbn, 1k tbc
Stream #0.1: Audio: adpcm_ima_wav, 8000 Hz, 1 channels, s16, 32 kb/s
```

Ce flux présente l'avantage d'intégrer l'audio, contrairement à **videostream.cgi**, mais dispose des mêmes arguments utilisables (résolution).

Une capture fixe se fera via `http://user:pass@IP/snapshot.cgi`. Il devient alors très simple, via un script shell, de régulièrement prendre un cliché de ce que voit la caméra, l'horodater et, par exemple l'envoyer par mail ou le déposer dans un répertoire local ou distant. L'outil **curl** permettra de récupérer les données graphiques qu'on redirigera ensuite vers l'un des outils ImageMagick comme **convert** pour traitement ou **display** pour un affichage immédiat. Exemple :

```
% curl http://user:pass@192.168.0.202:80/snapshot.cgi | \
convert -gravity NorthWest -pointsize 30 \
-font @/home/denis/TrueType/arialbd.ttf -stroke black \
-strokewidth 1 -fill red \
-draw "text 5,5 `date +%d/%m/%Y %H:%M`" - png:- | \
display -
```



Exemple de capture obtenue via une URL de la caméra puis modifiée à la volée avec ImageMagick en incrustant la date et l'heure.

curl récupère l'image, **convert** la transforme et **display** l'affiche. On utilise les arguments d'ImageMagick pour ajouter un texte rouge à bord noir dans le coin supérieur gauche de l'image en ajoutant la date issue de la commande classique **date**. Ceci offre déjà plus de souplesse que ce que peuvent généralement proposer les firmwares de caméra IP de marque en permettant de choisir les couleurs, l'emplacement, et le texte lui-même. L'aspect ludique est également intéressant puisqu'en utilisant ce type de commandes dans une **crontab** ou avec **watch**, il devient possible de créer un **time laps** d'une image toutes les minutes, par exemple, sur quelques jours. On assemblera le tout, ensuite avec **FFmpeg**, par exemple, pour produire une vidéo sympathique.

Nous n'avons, pour l'instant, été passifs que vis-à-vis du périphérique, mais il est également possible de le contrôler via l'URL `http://user:pass@IP/decoder_control.cgi?command=XX`. **XX** est une valeur numéro pouvant être :

- 0 : Mouvement haut activé ;
- 1 : Mouvement haut stop ;
- 2 : Mouvement bas activé ;

- 3 : Mouvement bas stop ;
- 4 : Mouvement gauche activé ;
- 5 : Mouvement gauche stop ;
- 6 : Mouvement droite activé ;
- 7 : Mouvement droite stop ;
- 26 : Patrouille verticale ;
- 27 : Patrouille verticale stop ;
- 28 : Patrouille horizontale ;
- 29 : Patrouille horizontale stop ;
- 94 : activation relais ;
- 95 : désactivation relais.

Remarque

Il est possible avec la technique URL+ImageMagick d'intégrer dans une image toutes sortes d'informations complémentaires (et même des éléments graphiques) bien plus qu'il n'est possible de le faire avec le firmware original de n'importe quelle caméra IP. Ici, nous avons intégré la date et l'heure mais également les informations en provenance d'un capteur de température et d'hygrométrie accédé en Bluetooth.



```
% curl http://user:pass@192.168.0.202:80/snapshot.cgi | \
convert -gravity NorthWest -pointsize 30 \
-font @/home/denis/TrueType/arialbd.ttf -stroke black \
-strokewidth 1 -fill red \
-draw "text 5,5 'date +%d/%m/%Y %H:%M'" \
-gravity SouthEast -fill green \
-draw "text 5,5 '/home/denis/bin/temp.py 00:19:50:EE:A4:06'" \
-png:- | display -
```

Encore une fois, ce n'est qu'un simple exemple, les informations intégrables peuvent être de toutes natures : dernier mouvement détecté, état de l'environnement, information de positionnement, etc. Vous n'avez que l'embaras du choix en guise de limite ainsi que, bien sûr, l'espace à votre disposition sur l'image.

L'utilisation peut se faire avec n'importe quel outil ou code capable de lancer une requête HTTP. Une utilisation simple, avec **curl**, est la suivante :

```
% curl http://user:pass@192.168.0.202:80/decoder_control.cgi?command=30
ok.
% curl http://user:pass@192.168.0.202:80/decoder_control.cgi?command=452
error: illegal params.
```

Comme vous pouvez le voir, une commande valide retournera **ok**. et un message d'erreur signale un problème. D'autres valeurs semblent pouvoir être utilisées sur les Foscam et d'autres clones, mais n'apportent aucun effet avec ce modèle de caméra. C'est le cas, par exemple, des commandes permettant de définir des positions. Entre **30** et **63**, les valeurs paires fixent la position courante et les valeurs impaires permettent à la caméra de s'y rendre :

- **30** : fixe la position courante comme celle de la position 0.
- **31** : va à la position 0.
- **32** : fixe la position courante comme celle de la position 1.
- **33** : va à la position 0.
- etc.

Ces commandes utilisées avec notre caméra Heden retournent bien **ok**. mais n'ont aucun effet. Enfin, la valeur **25** permet théoriquement de recentrer la caméra après calibration horizontale et verticale. Dans les faits, après quelques mouvements désordonnés, la caméra pointe un endroit qui n'est absolument pas une position centrée.

Une autre URL CGI est **set_misc.cgi?led_mode=**. Celle-ci permet de contrôler l'activité de la LED en façade. La valeur passée en argument peut être :

- **0** : la LED clignote une fois la caméra connectée (réseau configuré).
- **1** : la LED clignote pendant la configuration réseau et une fois connectée.
- **2** : la LED est toujours éteinte. Notez que les LED IR émettent une faible lueur rouge en raison de la proximité du rayonnement avec le spectre de lumière visible. Il n'est pas totalement possible de faire passer la caméra pour inactive sans modification matérielle.

D'autres URL existent, permettant diverses manipulations sans passer par un navigateur, comme :

- la récupération des données de configuration avec **get_misc.cgi** et **get_params.cgi** ;
- l'obtention du journal de connexions avec **get_log.cgi** ;
- le reboot distant (**reboot.cgi**) ;
- ou encore la mise à jour du firmware **upgrade_firmware.cgi**.

Pour une liste complète et peut-être compatible avec votre matériel, consultez le document <http://www.notesco.net/download/ipcamcgisdk21.pdf>, relativement exhaustif.

3 Creuser un peu et prendre des risques

Certains considéreront que c'est maladif ou révélateur d'un important problème psychologique, mais c'est comme ça et je l'assume pleinement : je ne peux que difficilement m'empêcher de démonter et inspecter les entrailles du matériel que j'achète. Ceci est synonyme, bien sûr, d'annulation de garantie, mais le plus souvent, cela vaut vraiment la peine. Dans le cas du matériel ici exploré et comme avec bien des équipements aussi "intelligents", la première phase consiste à trouver une console série.

La méthodologie est la suivante :

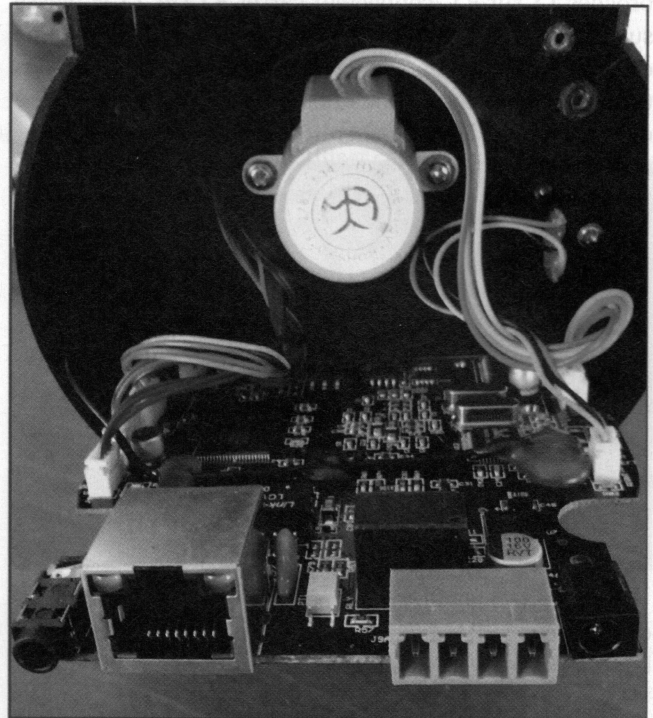
- Après ouverture, chercher sur le circuit un jeu de broches ou d'emplacement à souder.
- Déterminer si parmi les broches en question figure une liaison à la masse et à la tension d'alimentation (Vcc), puis en déduire la tension à utiliser pour la liaison série (généralement 5V ou 3,3V).
- En fonction du nombre de broches à l'utilité indéterminée, trouver TX et RX par tâtonnement.
- S'aider du circuit lui-même en suivant les pistes. Dans bien des cas, les lignes RX et TX sont reliées directement au microcontrôleur/SoC.

Votre meilleur ami est ici le multimètre, à la fois pour mesurer les tensions et en déduire la connectique entre composants. Bien entendu, dans cette procédure, faire la différence entre un port série, un jeu de GPIO ou un port JTAG n'est pas chose facile. Laisser libre court à son instinct est souvent la meilleure solution.

Dans le cas présent, l'ouverture du boîtier ne présente pas de difficulté. Quatre vis sous la caméra permettent de fixer la partie inférieure qui découvre un circuit comportant, entres autres choses, les composants suivants :

- Nuvoton V90N745CDG : un SoC Winbond basé sur un cœur ARM7TDMI à 80 Mhz avec RAM et Flash externe mais intégrant un l'Ethernet, 4 UART, 3 Timers, 32 GPIO, deux contrôleurs USB 1.1 (hôte et esclave), i2c, 4 canaux PWM, un contrôleur audio AC97. Le tout avec gestion d'alimentation et I/O en 3,3V. Bonne nouvelle (pour plus tard), le SoC dispose d'un environnement de compilation basé sur GCC et un BSP Linux pour un kit dévaluation Winbond.
- MEGA MPSV12816FS-6K : mémoire vive introuvable sur le Web ;
- Spansion S29AL016J70TF102 : mémoire flash de 2Mo bootable ;
- Davicom DM9161 : Transceiver Ethernet 10/100 ;
- ALC203 : un DAC audio de RealTek ;

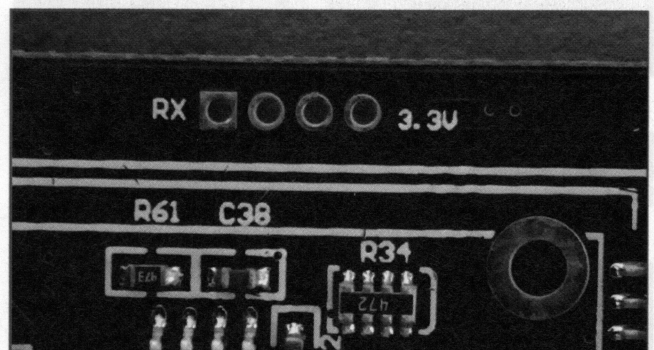
- Golden GY-1C-5L : relais Golden 5V avec pouvoir de coupure de 0,5A/125V alternatif et 1A/24V continu.



La caméra ouverte avec à l'avant de gauche à droite, le port audio, l'Ethernet, l'optocoupleur pour l'entrée alarme, un peu à l'arrière le relais, en vert le connecteur "I/O Alarm" et enfin le connecteur d'alimentation. On voit à l'arrière le moteur pas-à-pas du mouvement horizontal et toute la connectique interne (moteurs, micro, caméra, détecteur de fin de course pour les axes de rotation, etc.).

Une vis supplémentaire maintient le circuit en place qui devient mobile avec son retrait. On déconnectera avec soin les différents câbles (vidéo, audio, moteur, etc.). Sur la face du circuit qui apparaît alors, nous sommes aidés par la sérigraphie. Un connecteur 4 broches en bordure est marqué "RX" et "3.3V". Aucun doute, il s'agit d'un port série. Il se trouve que le marquage référence exactement le brochage, soit :

```
"RX" [rx](tx)(gnd)(Vcc) "3.3V"
```



Le connecteur pour console série sur la carte mère de la caméra est clairement exposé.

Il nous suffit donc de connecter un adaptateur USB/série supportant 3,3 Volts et de connecter respectivement ses broches TX sur RX (connecteur carré), RX sur TX et masse sur masse. Ceci fait, le premier essai est réussi avec une connexion via GNU Screen en 115200 bps. Dès la mise sous tension de la caméra, nous obtenons des informations, à commencer par le bootloader :

```

W90P745 Boot Loader [ Version 1.1 $Revision: 1 $ ] Rebuilt on Aug 19 2009
Memory Size is 0x1000000 Bytes, Flash Size is 0x200000 Bytes
Board designed by Winbond
Hardware support provided at Winbond
Copyright (c) Winbond Limited 2001 - 2006. All rights reserved.
Boot Loader Configuration:

```

```
MAC Address      : 00:60:6E:5D:EB:45
IP Address       : 0.0.0.0
DHCP Client      : Enabled
CACHE            : Enabled
BL buffer base   : 0x00300000
BL buffer size    : 0x00100000
Baud Rate        : -1
USB Interface     : Disabled
Serial Number     : 0xFFFFFFFF
```

On retrouve la taille de la mémoire flash ainsi qu'une indication sur la RAM : 16 Mo ! En laissant le processus de boot se poursuivre, on apprend :

- La version du noyau Linux utilisée et une confirmation de la taille de la RAM :

```
Linux version 2.4.20-uc0 (root@maverick-linux) (gcc version 3.0) #978
EÃ 80Ã 20 01:23:39 CST 2009
Processor: Winbond W90N745 revision 1
Architecture: W90N745
On node 0 totalpages: 4096
zone(0): 0 pages.
zone(1): 4096 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/rom0 rw
Calibrating delay loop... 39.83 BogoMIPS
Memory: 16MB = 16MB total
```

- La prise en charge des ports séries :

```
Winbond W90N745 Serial driver version 1.0 (2005-08-15) with no serial
options enabled
ttyS00 at 0xffff80000 (irq = 9) is a W90N745
Winbond W90N7451 Serial driver version 1.0 (2005-08-15) with no
serial options enabled
ttyS00 at 0xffff80100 (irq = 10) is a W90N7451
```

- Le type de flash supporté et l'Ethernet :

```
AM29LV160DB Flash Detected
01 eth0 initial ok!
```

- La prise en charge USB hôte et audio :

```
Winbond Audio Driver v1.0 Initialization successfully. [...] usb.c:
new USB bus registered, assigned bus number 1 hub.c: USB hub found
hub.c: 2 ports detected usb.c: registered new driver audio audio.c:
v1.0.0:USB Audio Class driver usb.c: registered new driver serial
usbserial.c: USB Serial Driver core v1.4 </code>
```

- La présence d'un support pour chipset Wi-Fi ZyDAS ZD1211 :

[illegible]

- Le montage des systèmes de fichiers :

```
Shell invoked to run file: /bin/init
Command: mount -t proc none /proc
Command: mount -t ramfs none /usr
Command: mount -t ramfs none /swap
Command: mount -t ramfs none /var/run
Command: mount -t ramfs none /etc
Command: mount -t ramfs none /flash
Command: mount -t ramfs none /home
Command: camera&
```

On obtient ensuite un shell utilisable nous permettant de faire le tour du système. Notez l'exécution du binaire **camera** en fin de boot qui est le processus fournissant l'interface web et le contrôle de la caméra. En dehors du système lui-même, l'ensemble du fonctionnement du firmware semble reposer sur ce binaire dont, bien entendu, nous n'avons pas les sources. Les deux éléments (système et WebGui) semblent, de plus, bénéficier de mises à jour différentes sous la forme de deux images, chez tous les constructeurs. Dernier élément en rapport, l'ensemble des données HTML sont placées dans **/home**. En fouillant un peu, on constate également la présence du fichier **rt73.bin** dans **/bin**. Il s'agit d'un firmware pour clé USB Wi-Fi RaLink. On peut en déduire qu'au moins deux adaptateurs intègrent généralement les versions Wi-Fi des caméras, et ce, en USB. Le démarrage du système est laissé à la discrétion du seul fichier **/bin/init** qui est un script montant les systèmes de fichiers, appelant **camera** puis **sh**. Il n'y a pas de services et le contenu de **/bin** ne laisse pas de doute quant à l'utilité de **camera** :

```
-rwxr-xr-x 1 0 0 128715 Jan 01 00:00 camera
-rwxr-xr-x 1 0 0 40713 Jan 01 00:00 dhcpc
-rwxr-xr-x 1 0 0 21610 Jan 01 00:00 ifconfig
-rwxr-xr-x 1 0 0 194 Jan 01 00:00 init
-rwxr-xr-x 1 0 0 38300 Jan 01 00:00 iwconfig
-rwxr-xr-x 1 0 0 33630 Jan 01 00:00 iwpriv
drwxr-xr-x 1 0 0 32 Jan 01 00:00 mypppd
-rwxr-xr-x 1 0 0 28824 Jan 01 00:00 route
-rwxr-xr-x 1 0 0 2048 Jan 01 00:00 rt73.bin
-rwxr-xr-x 1 0 0 31043 Jan 01 00:00 sh
-rwxr-xr-x 1 0 0 41540 Jan 01 00:00 wextcl
-rwxr-xr-x 1 0 0 96317 Jan 01 00:00 wpa_supplicant
```

/usr est vide, tout comme **/etc** et **/flash**. Enfin, le répertoire **/var** contient pour seul et unique élément le fichier PID du client DHCP. Nous avons là affaire à quelque chose de très simple et très fermé en même temps. Il ne nous sera pas possible de modifier simplement le firmware et tout ne peut se jouer que depuis le bootloader. Fort heureusement, celui-ci est un peu plus bavard :

Press ESC to enter debug mode

bootloader >

bootloader > h

W90P745 Command Shell v1.0 Rebuilt on Nov 26 2008 at 15:30:34

H Display the available commands

B Set Baud Rate

D Display memory. D -? for help

E Edit memory. E -? for help

G Goto address

I information

MX Xmodem download

MT TFTP/USB download

FT Program the flash by TFTP/USB. FT -? for help

FX Program the flash by Xmodem. FX -? for help

CP Memory copy

LS List the images in the flash

SET Setting boot loader configuration. SET -? for help

CHK Check the flash

RUN Execute image

DEL DEL the image or flash block

MSET Fill memory

TERM Change the terminal output port

BOOT Reboot the system

CACHE Cache setting

USB USB interface setting

UNZIP Unzip image

ATTRIB Change the image attribution

INTF Print bootloader supported interface, ether USB or MAC

bootloader > LS

Image: 0 name:BOOT INFO base:0x7F010000 size:0x00000038 exec:0x7F010000 -af

Image: 7 name:linux base:0x7F020000 size:0x0000BE60 exec:0x00008000 -acxz

Image: 6 name:romfs base:0x7F0E0000 size:0x00008C00 exec:0x7F0E0000 -a

bootloader > USB

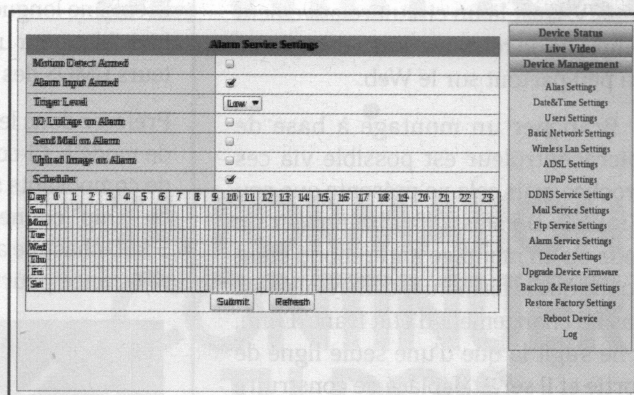
This bootloader doesn't support USB, please use TCP/IP instead

Diverses documentations en ligne comme celles de zoneminder.com, gadgetvictims.com ou openipcam.com indiquent comment charger un firmware. Notez que OpenIPcam fournit un BSP pour le SoC à la fois pour le chip marqué Winbond et celui Nuvoton. La construction d'un système partant de zéro pour ce périphérique est quelque chose qui sort du cadre de cet article, mais nous gardons le sujet sous le coude pour une prochaine fois.

4 Étendre les fonctionnalités de la caméra

Lorsque je parle d'étendre les fonctionnalités, il s'agit principalement d'ajouter des éléments matériels à l'équipement. Dans notre cas, et devant la tâche difficile et *time consuming* de la création d'un nouveau firmware basé sur OpenIPcam, par exemple (qui ne supporte que les systèmes disposant d'une flash de 4Mo), nous nous limiterons à des solutions reposant sur l'utilisation du système dans l'état. L'une des voies qui se prête à cette extension est, bien entendu, l'utilisation du connecteur marqué "I/O alarm". Le déclenchement d'une action par l'alarme semble délicat. En regardant en détail l'interface web de la caméra, on remarque que :

- L'entrée peut être déclenchée au niveau haut ou bas.
- Il est possible d'associer directement la commutation du relais.
- Il est possible d'envoyer un mail que nous pouvons traiter par la suite.
- Nous pouvons envoyer l'image sur un serveur FTP. Là encore, nous pouvons utiliser cet évènement en émulant un serveur FTP pour traiter le déclenchement.
- Il est possible de n'activer cette alarme que sur des plages de temps bien définies.



L'interface de gestion web permet de configurer l'action à déclencher en cas de changement d'état sur un connecteur externe de la caméra.

Étendre le fonctionnement de la caméra par ce biais est donc possible à l'aide d'un dispositif quelconque équipé d'un relais et ouvrant ou fermant la liaison entre la masse et la ligne d'entrée. Les broches à l'arrière de la caméra sont numérotées de 1 à 4 de gauche à droite. 1 et 2 correspondent à la sortie du relais interne. 3 est la ligne d'entrée et 4 est une ligne raccordée à la masse. En étudiant le circuit, il s'avère que la ligne d'entrée est en réalité connectée à un optocoupleur assurant ainsi l'isolation et la protection du reste du circuit, le fait de mettre la broche 3 à la masse alimente la LED de l'optocoupleur. Une tension est donc présente sur cette broche et on évitera d'y connecter un module électronique directement en l'absence de datasheet pour le composant, bien qu'une mesure directe affiche une tension de 5V et un courant de quelque 35mA. L'usage dédié à cette entrée est généralement la connexion d'un capteur infra-rouge de mouvement (PIR sensor). Les modèles courants disposent de relais qu'il suffira alors de connecter directement entre les broches 3 et 4. D'autres utilisations sont donc possibles en utilisant un interrupteur ou un bouton poussoir de manière à obtenir le même comportement. Il devient alors possible de détecter toutes sortes d'évènements physiques relativement simplement.

Les lignes 1 et 2 du connecteur sont plus intéressantes car elles permettent d'agir sur l'environnement physique où se trouve la caméra et ce via l'interface web comme via

la ligne de commandes (cf. `decoder_control.cgi?command=94` et `95`). Il ne faut cependant pas perdre de vue que le relais intégré a ses limites (1A/24VDC ou 0.5A/125VAC) et qu'il est donc absolument hors de question de commander directement un éclairage classique branché au secteur. Il sera alors préférable de connecter les sorties du relais à un autre relais de plus grande capacité (type 10A/240VAC) alimenté en 12V. C'est là un circuit relativement simple qu'on retrouve décrit et expliqué un peu partout sur le Web.

Brancher un montage à base de microcontrôleur est possible via ces broches mais cela ne présente que peu d'intérêt. De plus, il faudra prévoir un code ou un montage anti-rebond étant donné que la fermeture d'un relais n'est pas un changement d'état franc. Enfin, il ne s'agit là que d'une seule ligne de sortie et il serait déplacé de construire un quelconque protocole de communication afin de piloter plusieurs appareils (genre 1 clic = appareil A, 2 clic = appareil B, etc.). Nous avons une autre solution pour cela, presque aussi tordue.

Il existe une URL dont nous n'avons pas encore parlée qui permet d'envoyer des données sur le port série : `Comm_write.cgi`. Les arguments sont les suivants :

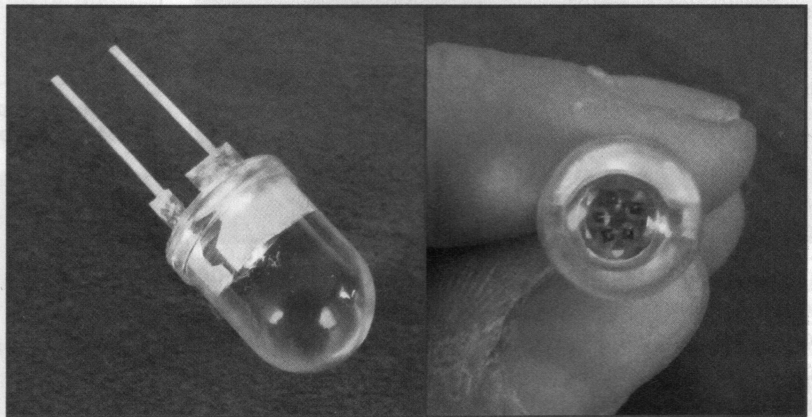
- **port=** : le numéro du port série à commander, **0** étant la console série de Linux.
- **baud=** : qui peut être **9** (1200 bps), **11** (2400 bps), **12** (4800 bps), **13** (9600 bps), **14** (19200 bps), **15** (38400 bps), **4097** (57600 bps), ou **4098** (115200 bps).
- **bytes=** : le nombre total d'octets envoyés.
- **data=** : les caractères à envoyer. Attention, n'oubliez pas qu'il s'agit d'une URL et qu'il faudra donc encoder certains caractères comme l'espace (`%20`), mais que ceux-ci ne comptent que pour un octet.

Nous n'avons pas trouvé sur le circuit d'autres connecteurs correspondant aux trois autres ports séries. Il n'est même

Des histoires d'infra-rouge

Parmi les extensions possibles, vous pouvez choisir d'ajouter un système infra-rouge un peu plus efficace que les 10 LED intégrées à la caméra. Il vous suffira pour cela de mettre en œuvre des LED avec une puissance de rayonnement supérieure qu'on peut trouver dans les boutiques d'électronique en ligne ou sur eBay. Il faut cependant faire attention à la différence existant entre les LED d'éclairage et celles de signalisation. Les premières utilisent une longueur d'onde de 850nm, ce qui les rend légèrement visibles à l'œil nu sous la forme d'un faible petit point rouge. Les secondes sont destinées, par exemple, aux télécommandes ou aux systèmes de transmission de données (IRDA). Elles utilisent une longueur d'onde de 880, 920 ou 940nm, ce qui les rend totalement invisibles pour un humain. Cependant, leur puissance est plus faible et les capteurs CMOS des caméras de vision de nuit y sont moins sensibles.

Préférez donc le 850nm qui, dans cette gamme, propose une vaste sélection de modèle. Ci-contre, une LED 10mm achetée sur eBay auprès d'un vendeur de composants spécialisé dans le tuning de voitures. Elle comporte 5 "dies" ou "chip" connectées en interne en parallèle. C'est donc là l'équivalent de 5 LED chacune alimentée avec 20mA. La mise en œuvre nécessite une simple résistance pour limiter le courant et le tour est joué.



Notez que les LED IR n'affichent pas de puissance lumineuse en MCD puisqu'il ne s'agit pas de lumière visible. Cependant, on peut déduire avec plus ou moins d'exactitude l'efficacité d'une telle LED en fonction de sa puissance. Celle-ci présente une tension à ses bornes de 1.6V, ce qui nous donne $(5 \times 0.02) \times 1.6 = 160$ milliwatts. Certaines LED IR de forte puissance peuvent afficher de 1 à 5 watt avec un angle de diffusion de 140° (contre 40° pour la LED ci-contre). De quoi largement "éclairer" une pièce pour votre caméra. Attention cependant à bien prendre en compte la nécessité d'ajouter un dissipateur thermique (radiateur) car le dégagement de chaleur est très important.

pas certain que le noyau embarqué ait été correctement configuré pour cela. En revanche, nous pouvons utiliser la sortie dédiée à la console série du système. Bien entendu, celle-ci est en quelque sorte multiplexée et il faudra faire avec. L'idée est de connecter un microcontrôleur disposant d'une liaison série à

115200 bps et de le connecter à la console série. Il recevra donc tous les messages de démarrage et deux options s'offrent alors à nous :

- Le laisser faire le tri permanent entre les messages du système et ceux que nous transmettons via `Comm_write.cgi`.

- Utiliser le relais afin de signaler au microcontrôleur qu'il doit être à l'écoute des messages.

Dans les deux cas, nous devons établir un protocole clair permettant de reconnaître les véritables messages que nous envoyons par opposition à d'éventuelles sorties apparaissant tantôt sur la console :

```
ntpc adjust ok
do_file: can not find file favicon.ico
set resolution 4
ntpc adjust ok
unknown resolution
ioctl failed -14
```

Personnellement, j'ai opté pour la syntaxe suivante :
----nombre de caractères:chaîne, le tout préfixé et clos par des CR/LF. Ceci nous donne une URL complète, comme : **Comm_write.cgi?port=0&baud=4098&bytes=16&data=%0a%0d---6:coucou%0a%0d**". Sur la console série apparaît alors :

```
---6:coucou
no support
```

Il est possible de compléter ce protocole en ajoutant, par exemple, une somme de contrôle permettant d'assurer l'intégrité des données reçues et leur validité (somme des valeurs ASCII des caractères modulo 256 en notation hexa, pourquoi pas). En se basant sur cette solution, il devient possible d'implémenter un jeu de commandes avec paramètres qui pourront être interprétés par le montage connecté sur la liaison série. Il est même possible de piloter le shell de cette manière. Là, vous n'avez plus de limitations et vous pouvez aussi bien commander des systèmes mécaniques, que d'éclairages, ou encore un ensemble de servos permettant d'orienter la caméra dans des positions plus exotiques. Ce n'est plus qu'une affaire de code pour le microcontrôleur de votre choix (AVR/Arduino, MSP430, STM8/32, PIC, etc.).

5 Développer son interface et sa surveillance

Nous n'allons pas ici entrer dans le détail du développement d'une application complète et ce pour plusieurs raisons. Étant donné le type de flux video, MJPEG, il existe une quantité très importante de langages, de bibliothèques et de *toolkits* en mesure d'afficher ce type de données. De la même manière, le contrôle des mouvements de la caméra prenant la forme de simples requêtes HTTP est tout aussi facile à implémenter. Mais la raison principale qui justifie le fait de ne pas développer une application cliente est sa relative inutilité. En effet, le fait de disposer d'une application en tant que telle,

MAI THE FORCE BE WITH YOU

TA FORMATION CONTINUER TU DOIS

LE PROGRAMME CI DESSOUS TU TROUVERAS

NOS SESSIONS DE MAI 2012

PARIS

LPI 301	9 au 15 mai
Drupal Themeur	10 au 11 mai
LP101	21 au 24 mai
Drupal Commerce Masterclass	21 au 25 mai
Sécurité des réseaux	21 au 25 mai

TOULOUSE

LPI 102	29 mai au 1er juin
---------	--------------------

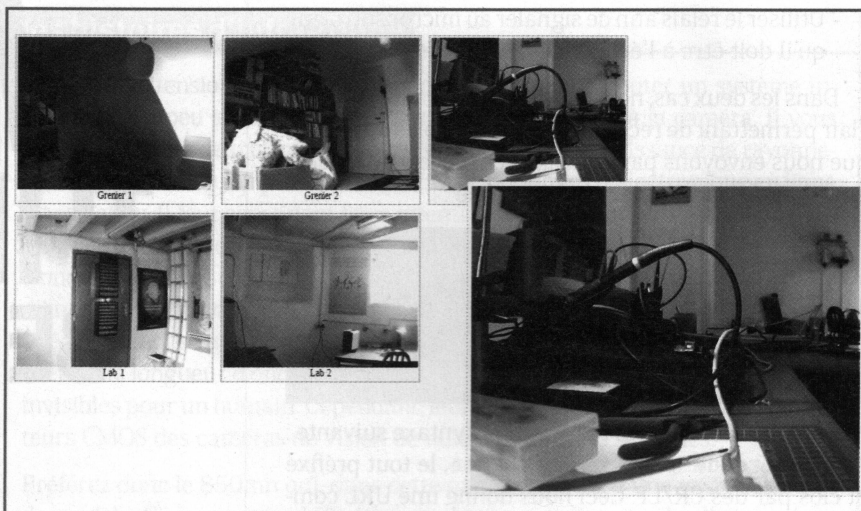
un client lourd, n'apporte pas grand chose en dehors de l'intégration du support de la caméra dans un applicatif existant. Je pense, par exemple, à une éventuelle solution domotique, déjà développée et devant intégrer le flux vidéo. Mais là encore, les cas sont légion et fortement dépendants de l'existant.

Seule exception à ce raisonnement, le développement d'une application pour mobile. Or, des applications de vidéo surveillance pour iPhone ou Android existent déjà et supportent un grand nombre de caméras IP. Quelques solutions sont d'ailleurs open source et très complètes. Inutile de réinventer la roue. L'une de ces applications est celle de David Bromberg qui a encadré des élèves dans leur projet de fin d'étude et a fait le tour de la question de manière relativement exhaustive. Le mémoire ainsi que le code Android développé sont disponibles sur <http://code.google.com/p/android-camera-axis> (Axis n'est pas la seule marque de caméra traitée, contrairement à ce que laisse penser l'URL).

Pour des besoins plus immédiats, l'utilisation de plusieurs caméras du modèle que nous avons analysé ici soulève un problème : comment afficher un écran présentant l'ensemble des flux vidéo en une seule fois ? La réponse tient en deux mots : page web. En effet, il n'est pas nécessaire de se lancer dans le développement d'une application pour si peu. Les flux vidéo sont compatibles avec les navigateurs courant et un peu de HTML sera suffisant pour une base de travail.

Un simple tag `` permettra d'intégrer le MJPEG dans une page et avec un peu d'insistance, on construira un simple tableau présentant les miniatures des images *live* de manière à ce qu'en passant la souris dessus, l'image apparaisse en plus grand (640×480).

Un résultat préliminaire acceptable peut être ceci :



Le contrôle des caméras est un problème plus délicat à régler d'un point de vue client web. En effet, on entre de plein pied dans le domaine de l'AJAX puisqu'il nous faut exécuter des requêtes depuis notre page vers un site distant (le serveur de la caméra). Se pose alors le principal problème rencontré par les développeurs AJAX, j'ai nommé les *cross-domain requests*. Il existe une solution simple supportée par les navigateurs modernes consistant à utiliser **XMLHttpRequest** et le CROS (*Cross-Origin Resource Sharing*). Malheureusement, il faut dans ce cas que le serveur accédé retourne une en-tête **HTTP Access-Control-Allow-Origin** spécifiant quels domaines sont autorisés. Ce n'est bien sûr pas le cas de celui intégré dans la caméra et nous n'avons pas de moyen simple de l'ajouter. Nous arrivons donc à une situation similaire à une tentative d'attaque XSS où nous devons contourner les limitations de sécurité ou éventuellement utiliser un proxy pour traiter les requêtes et les renvoyer aux différentes caméras. Ceci, cependant, alourdit notre simple tentative de regroupement des flux vidéo pour des opérations rares de repositionnement des caméras.

Finalement, la solution la plus simple est d'afficher les flux vidéo et d'ajouter un lien permettant l'accès direct à l'interface web de chaque caméra, ce qui offre déjà une facilité intéressante.

Conclusion

Comme nous l'avons vu, il est relativement simple de construire un environnement de surveillance hackable à base de caméras d'entrée de gamme. De l'utilisation basique à l'ajout de fonctionnalités, l'aspect "complet" de l'architecture ne dépend que du temps et de l'énergie que vous souhaitez investir. Ce genre de matériel, pour un coût minimal, offre davantage de possibilités qu'une solution entièrement "faite maison" (plateforme embarquée ARM, Webcam USB et servo). Bien entendu, rien ne vous interdit de mélanger les genres en utilisant ces caméras IP pour les zones peu sensibles, la solution "faite maison" pour les zones où le mouvement de caméra doit être plus contrôlable, et enfin, des caméras haut de gamme pour les zones sensibles. Le seul regret, finalement, est l'absence d'un projet open source ayant pour objectif de fournir un firmware librement modifiable sur la base d'un environnement comme buildroot, un peu comme ce que l'on trouve pour les routeurs avec OpenWrt/LuCI. Les sources existent pourtant en dehors des BSP disponibles pour le SoC Winbond/Nuvoton car sinon, comment les "cloneurs" du matériel Foscam pourraient-ils fournir des firmwares aussi proches de l'original ? ■