

**UNIVERSITÀ DEGLI STUDI DI BRESCIA**

**FACOLTÀ DI INGEGNERIA**

**Dipartimento di Elettronica per l'Automazione**

**Laboratorio di Robotica Avanzata**

**Advanced Robotics Laboratory**

**Corso di Robotica**

**(Prof. Riccardo Cassinis)**

# Webcameras per LDRA

Elaborato di esame di: **Diego Zuccotti**

Consegnato il: **20 novembre 2003**

# 1 Obiettivi

Questo lavoro parte da una esperienza analoga prodotta dai miei colleghi Alessandro Saetti e Sergio Spinoni, il precedente elaborato usufruiva di una webcam su porta parallela per l'acquisizione di immagini alla frequenza di una al minuto e conservava memoria di una immagine ogni 5 minuti per l'ora precedente, il tutto veniva inviato tramite ftp. Le modifiche richieste consistevano nella sostituzione della singola webcam parallela con 3 webcam su usb la cui configurazione si noterà si è rivelata un po' laboriosa, inoltre si intendeva accelerare la frequenza di scatto portandola a frazioni di un minuto, cosa assolutamente infattibile con le metodologie adottate dai miei colleghi i quali, molto elegantemente ricorrevano a degli script lanciati dal demone crond (come sarà senz'altro noto esso permette di impostare l'esecuzione di dati task con la precisione massima di un minuto). Infine si voleva potenziare le capacità di archiviazione prendendo uno scatto ogni minuto dalle 3 webcam e conservando le immagini ottenute di una intera giornata di 24 ore.

## 2 Configurazione sistema

Affronteremo ora passo passo la procedura di configurazione di Linux per il riconoscimento delle periferiche utilizzate. Verrà naturalmente dato per scontato che disponiate di un sistema GNU/Linux correttamente funzionante.

Prima di tutto è importante disporre dei dati relativi alle webcam che si vogliono adoperare, in genere nome del produttore e modello sono sufficienti, se si tratta in particolare di videocamere su porta usb (attualmente è questa la norma) consiglio di visitare la pagina [www.smcc.demon.nl/webcam/index.html](http://www.smcc.demon.nl/webcam/index.html) che costituisce il principale riferimento per il supporto delle webcam usb su linux, in particolare potrete trovare qui il modulo “pwc” che consente al kernel di Linux di riconoscere ed attivare i dispositivi: inizialmente sviluppato per supportare le webcam economiche prodotte da Philips il modulo è stato successivamente potenziato per consentirgli il funzionamento con buona parte delle webcam in circolazione, sul sito indicato potrete trovare la lista completa del hardware supportato.

### 2.1 Acquisizione informazioni

Sarà bene, prima di procedere, compiere una piccola verifica, date i comandi:

***uname -r***

che indica la versione del kernel attualmente installato sulla vostra macchina

***modprobe -l | grep pwc***

che dovrebbe restituire qualcosa come `/lib/modules/2.4.xx../kernel/drivers/usb/pwc.o[.gz]`

Da un certo tempo il modulo pwc è entrato a far parte della branca ufficiale del kernel e pertanto è normalmente incluso nei sorgenti, sul sito summenzionato troverete un elenco delle webcam supportate, a fianco delle stesse si trova segnata la minima versione del modulo necessaria perché esse funzionino, ancora a fianco si indica la minima versione del kernel necessaria affinché il modulo lavori correttamente. Vorrei aggiungere che il modulo è attualmente sviluppato soltanto per la versione 2.5 instabile del kernel e backported alla branca stabile 2.4.xx, i possessori di un kernel della versione 2.2 o precedenti dovranno sostituire i kernel procedura che non possiamo documentare in questa sede in quanto lunga, frustrante e laboriosa. Anche i possessori di un kernel nella versione 2.4.xx potrebbero essere costretti ad un upgrade per abilitare il supporto per il proprio hardware, tale caso si è per altro presentato nel nostro ARL.

Se le verifiche precedenti hanno dato il responso atteso siamo già in possesso di un kernel comprendente il modulo necessario, rimane da verificare se la versione dello stesso sia abbastanza recente per le nostre esigenze. In genere il file `Documentation/usb/philips.txt` contenuto nella directory dei sorgenti del kernel (se si utilizza il kernel accluso ad una distribuzione li troverete senz'altro sui cd di installazione) contiene numerose informazioni, se questo appartiene ad una versione vecchia non darà però la lista delle webcam supportate e nemmeno il numero di versione, si può ricorrere allora ai seguenti

comandi:

```
cd [PATH_DEI_SOERGENTI_DEL_KERNEL]  
less /usr/src/linux/drivers/usb/pwc.h | grep VERSION
```

che dovrebbe restituire qualcosa come:

```
#define PWC_VERSION "8.8"
```

riga nella quale è facile notare il numero di versione.

Come già detto se la versione in vostro possesso è già sufficiente rimane soltanto da caricare il modulo come spiegato più avanti.

## **2.2 Aggiornamento modulo**

Nel caso disponiate di una versione del modulo non abbastanza recente vi trovate di fronte a due possibilità: aggiornare il modulo o l'intero kernel. Paradossalmente la seconda scelta potrebbe rivelarsi la meno macchinosa. Vediamole entrambe nel dettaglio

### **2.2a Aggiornamento del solo modulo**

**!!sconsigliato se state usando kernel in versione non ufficiale come quelli allegati a distribuzioni che notoriamente applicano pesanti modifiche come Mandrake o Redhat**

Scaricate dal sito di prima la più recente versione del modulo che possa essere accettata dal vostro kernel (inutile dire che se tale versione fosse ancora troppo obsoleta per le vostre esigenze non vi rimarrà altra scelta che passare alla seconda procedura), scompattate l'archivio che avete scaricato in un percorso di vostra scelta, entrate nella directory creata e troverete delle sottodirectory identificate dalla versione del kernel per cui i sorgenti sono sviluppati, entrate in quella che vi compete. Sostituite i sorgenti del modulo contenuti in PATH\_SORGENTI\_KERNEL/drivers/usb con quelli che trovate nella directory indicata. Entrate nella directory dei sorgenti del kernel e date

```
make menuconfig
```

assicuratevi che il supporto *Multimedia\_devices* -->*video4linux* e *Usb\_support* --> *Philips cameras* siano abilitati come moduli. Date il comando:

```
make modules
```

se tutto va come dovrebbe troverete nella directory drivers/usb un file

```
pwc.o
```

che potete volendo comprimere con

```
gzip pwc.o
```

ottenendo un [pwc.o.gz](#) che dovrà sostituire l'omonimo file nella directory

/lib/modules/KERNEL\_VERSION/kernel/drivers/usb.

Date infine

**depmod -a**

e, se non ottenete errori, passate sotto per sapere come installare il modulo compilato.

## 2.2b Aggiornamento kernel

questa rimane la procedura consigliata anche perché nel nostro laboratorio è stata l'unica coronata da successo. Disponendo di una Mandrake 9.0 che contiene la versione 8.8 del modulo che non supporta nessuna delle nostre due Logitech, abbiamo tentato la procedura precedente senza successo, siamo poi passati ad una soluzione apparentemente più drastica ma in effetti molto semplice, procuratici i cd di una Mandrake 9.1 che incorpora il pwc 8.10 per noi sufficiente, abbiamo installato il pacchetto [kernel-2.4.21-0.13mdk.rpm](#) con il comando

***rpm -ivh PERCORSODELCD/Mandrake/RPMS/kernel....,***

si può passare a rpm il parametro `-nopost` per evitare che il pacchetto modifichi i link simbolici nella directory `/boot` oppure modificare la voce di lilo relativa al kernel precedente di modo che come kernel e initrd non vengano usati i generici `vmlinuz` e `initrd.img` ma i file appropriati ovvero `vmlinuz-VERSIONE` e [initrd.VERSIONE.img](#), ciò si ottiene modificando il file `/etc/lilo.conf`. In calce a tale file aggiungete le righe che seguono opportunamente adattate

***image=/boot/vmlinuz-2.4.21-0.13mdk*** (o il corretto nome del file del kernel)

***label=2.4.21*** (il nome che volete abbia l'immagine in lilo)

***root=/dev/hda1*** (partizione di root, casomai copiatela dalla stessa riga relativa alla voce `label=linux`)

***initrd=/boot/initrd-2.4.21-0.13mdk.img*** (file initrd corretto)

***append="quiet devfs=mount acpi=off"*** (opzioni commentate più sotto)

***read-only***

Vorrei far notare le due opzioni passate a lilo:

**devfs=mount** si rende necessaria affinché il kernel crei automaticamente i file di dispositivo al caricamento del modulo per le webcam, probabilmente era comunque già abilitata anche nel vostro kernel dato che viene ampiamente utilizzata affinché i kernel della versione 2.4.xx possano creare dinamicamente i file di device all'occorrenza.

**Acpi=off:** in bios che non supportano le specifiche acpi questa opzione è indispensabile perché se non viene passata il modulo relativo si interpone come unico interprete dei

messaggi del bios interpretandoli ovviamente in maniera errata, come se non bastasse si appropriava di tutti gli indirizzi di interrupt supponendo di essere l'unico preposto alla loro distribuzione, tutto questo si traduce nella impossibilità da parte dei dispositivi di rispondere alle assegnazioni di interrupt e di indirizzi di I/O assegnati dal kernel, in altre parole nel nostro caso particolare il sistema rileva la presenza dei dispositivi usb ma quando tenta di assegnare loro un indirizzo attende fino alla fine del timeout e suppone che essi lo abbiano rifiutato, tenta con un altro indirizzo ma ovviamente non ottiene risposta.

**!Problema audio:** Risulta da più parti, ed ho avuto modo di sperimentare personalmente sul mio pc casalingo, una incapacità da parte di linux, almeno nelle versioni attuali, a gestire dispositivi audio usb e non-usb in contemporanea, a nulla vale cambiare l'ordine di chiamata dei moduli, per ora l'unica possibilità si riconduce alla scelta di cosa installare, nel cd allegato troverete in Documentazione, il file [redhat8.0.html](#) che contiene una comoda procedura per disabilitare l'audio usb, nel nostro caso, utilizzando un pc privo di scheda audio e adoperando invece webcam con microfono abbiamo lasciato le cose inalterate, in tal modo funzionano soltanto i dispositivi audio usb. Curioso notare che il problema si presenta soltanto all'avvio del sistema, se una volta caricato il tutto si caricano i moduli nella giusta sequenza o se si avvia senza collegare le webcam e le si collega successivamente, tutto funziona correttamente.

A questo punto date

***lilo -v***

in modo che lilo carichi la nuova configurazione, ora consiglio di effettuare un reboot con entrambe le versioni del kernel, se si ha successo però sarà bene conservare comunque la vecchia versione fino a che non si sia completamente sicuri che tutto funzioni correttamente. Classico problema che si potrebbe incontrare è il mancato avvio del server x se si dispone di una scheda ATI o nVidia per cui siano stati installati i driver accelerati, dal momento che questi contengono un modulo che deve essere necessariamente nella versione appropriata al kernel in uso si renderà necessaria la reinstallazione degli stessi.

Per chi non disponesse di una versione aggiornata della propria distribuzione o volesse comunque adottare l'ultimo kernel disponibile raccomando di scaricare la versione ufficiale e non quella modificata dai produttori di distribuzioni. Configurate a piacere il kernel ponendo attenzione naturalmente che il supporto a video4linux e Philips cameras sia abilitato almeno come modulo. Compilate e installate.

## ***2.3 Caricamento modulo e verifica***

Siamo finalmente pronti a caricare il modulo per le nostre webcam, il comando più semplice è:

***modprobe pwc***

che dovrebbe eseguirsi senza intoppi, a questo punto dando:

***tail -f /var/log/messages***

dovreste notare il riconoscimento della webcam che avete collegato, eventualmente provate a connetterla e disconnetterla un paio di volte mentre il comando è in esecuzione per avere il colpo d'occhio sui messaggi del kernel. Dovreste notare dei messaggi che denotano l'assegnazione di un indirizzo ad un dispositivo usb, inoltre pwc dovrebbe reclamare la paternità della webcam indicando anche il giusto modello.

Dovrebbe ora risultare presente il dispositivo /dev/video0.

Se utilizzate kde 3 o successivi ed una Mandrake noterete la comparsa sul desktop di un link che avvia xawtv leggendo la webcam riconosciuta, potete usare il programma per il test o per valutare la qualità dell'immagine ma raccomando alcune cose:

**Non avviate contemporaneamente più istanze del programma e non avviatelo immediatamente dopo averlo spento** o molto probabilmente provocherà dei conflitti nel modulo, ve ne accorgete perchè se avete più dispositivi ne rimarrà visibile soltanto uno e diverrà impossibile disinstallare il modulo, a questo punto sarete costretti a riavviare.

Abbiamo appurato che il modulo riconosce ed abilita la webcam

## ***2.4 Installazione modulo modificato***

Potremmo a questo punto ritenerci soddisfatti dato che la webcam funziona ma nell'ambito del progetto di nostra competenza si richiede il funzionamento contemporaneo di almeno 3 webcam preferibilmente configurate in maniera differente.

Per ottenere tale risultato si rendono necessarie ulteriori piccole operazioni:

All'interno del cd fornito in allegato troverete nella directory "pwc\_modified" i sorgenti del modulo pwc nella versione 8.12 (testati soltanto con i kernel 2.4.21 e 2.4.22) opportunamente patchati per consentire la differenziazione delle webcam oltre che per il numero di serie del dispositivo usb anche per il nome esteso che tiene conto di codice produttore e codice prodotto. La modifica è stata resa necessaria dalla incapacità delle webcam in nostro possesso di segnalare un proprio univoco codice usb. Ulteriori dettagli verranno forniti più avanti. **Per ora è importante sapere che la modifica funziona nel nostro progetto ma nel caso le webcam in proprio possesso fossero dei modelli identici il codice usb rimarrebbe l'unico fattore di distinzione** e pertanto se si utilizzano camere che non lo comunicano non c'è modo di discernere dato che per il pc esse sono di fatto indistinguibili. L'unica altra possibilità è fare affidamento sul fatto che se le webcam sono collegate nella stessa maniera, per motivi prettamente elettrici esse vengono abilitate di norma sempre nello stesso ordine.

Tornando alla nostra procedura, come detto si può copiare i nuovi sorgenti sui vecchi nei

proprio sorgenti del kernel e procedere come per l'upgrade del modulo descritto poco sopra, per chi dispone del kernel 2.4.21-0.13mdk fornito con Mandrake 9.1 si può ricorrere alla versione precompilata contenuta nella omonima directory sul nostro cd e piazzandola al posto di /lib/modules/2.4.21-013mdk/kernel/drivers/usb/pwc.o.gz (non preoccupatevi per l'estensione gz, si tratta di un file zippato che può ugualmente essere caricato, per altro noterete che tutti i moduli forniti assieme alle distribuzioni sono in questo formato). Nel caso vogliate procedere manualmente alla modifica del sorgente per utilizzare una versione diversa non dovrete fare altro che aprire il file pwc-if.c e sostituire la riga:

```
if ((device_hint[hint].serial_number[0]== '*') ||
!strcmp(device_hint[hint].serial_number, serial_number))
```

con

```
if ((device_hint[hint].serial_number[0]== '*') ||
!strcmp(device_hint[hint].serial_number, serial_number) ||
!strcmp(device_hint[hint].serial_number, vdev->name))
```

La successiva modifica consiste nell'aggiungere la seguente riga al proprio /etc/modules.conf:

```
options pwc dev_hint=740."Logitech QuickCam Zoom":63,740."Logitech QuickCam Pro 4000":62,740."Philips 740 webcam":61,20,21,22 leds=250,750 fps=10
```

Per una spiegazione dettagliata dei parametri rimando al file philips.txt contenuto nella directory Documentation/usb all'interno dei sorgenti del kernel o sul cd nella stessa cartella pwc\_modified. Ciò che ci interessa in questo momento è che il modulo assegnerà alle webcam di nostro interesse i dispositivi

```
/dev/video61          alla Philips toUcam
/dev/video62          alla Logitech QuickCam 4000 pro
/dev/video63          alla Logitech QuickCam Zoom
/dev/video20,21,22    alle successive 3 webcam eventualmente individuate
```

Dal momento che i dispositivi video4linux possono essere numerati da 0 a 63 abbiamo scelto di assegnare alle nostre webcam gli ultimi disponibili così ci assicuriamo la non interferenza con altri dispositivi video come schede tv o decoder. Le telecamere usano tutte la sezione 740 del modulo e vengono distinte in base al nome assegnato, per la lista completa dei nomi accettati raccomando la lettura del pwc-if.c che fa parte dei sorgenti del kernel, segue la lista completa riferita alla versione 8.12:

Philips 645 webcam

Philips 646 webcam



Philips 675 webcam  
 Askey VC010 webcam  
 Philips 680 webcam  
 Philips 690 webcam  
 Philips 730 webcam  
 Philips 740 webcam  
 Philips 750 webcam  
 Logitech QuickCam Pro 3000  
 Logitech QuickCam Pro 4000  
 Logitech QuickCam Notebook Pro  
 Logitech QuickCam Zoom  
 Samsung MPC-C10  
 Samsung MPC-C30  
 Creative Labs Webcam 5  
 Sotec Afina Eye  
 Visionite VCS-UC300  
 Visionite VCS-UM100

Mi permetto di ricordare che successive release del modulo potrebbero rendere disponibili altri modelli ma si dovrà procedere ad una nuova modifica dei sorgenti.

Mi raccomando di indicare completamente il nome dei dispositivi ponendo attenzione a maiuscole e minuscole.

Come prova, con le webcam collegate rimuovete il modulo pwc (rmmod pwc) e reinstallatelo con:

***modprobe pwc***

modprobe leggerà la riga del modules.conf relativa e caricherà il modulo con le opzioni specificate, nel caso ne vogliate di diverse basta modificare la riga in questione. Se tutto è andato bene dovreste trovare i dispositivi /dev/video61 e successivi.

Ora bisogna copiare i file [CDROM]/etc/hotplug/usb/pwcx-i386.o e pwc in /etc/hotplug/usb/ , questi file consentono il caricamento automatico del modulo di decompressione pwcx al caricamento del pwc da cui dipende, il pwcx non è indispensabile ma consente di sfruttare le capacità di compressione delle webcam scaricando parecchio il bus usb. Nel caso non lo desideraste rimuovete pwcx-i386.o e **rimuovete l'opzione "compression=1" dalla riga di modules.conf** delle webcam

forniranno un segnale inintelligibile.

Pwc invece è lo script che attiva il caricamento effettivo del modulo pwc in caso di inserimento a caldo delle webcam, inoltre se lo trova carica anche pwcx-i386.o che però deve trovarsi nella directory indicata sopra. Riporto sotto lo script per intero:

```
#!/bin/bash
#
# Philips webcam module setup
#
#Visitate: www.smcc.demon.nl/webcam/support.html per supporto
# www.isthe.com/chongo/tech/comp/pwc/index.html per suggerimenti.
#
# Reso disponibile da Diego Zuccotti jeegsawk@libero.it,
# Ringrazio in particolare Landon Curt Noll per l'utile tutorial
# su cui il seguente script è basato.
#
# chongo (Share and enjoy! :-)) www.isthe.com/chongo/index.html
#
# Viene loggato il caricamento del modulo
#
/usr/bin/logger -p daemon.notice -t hotplug/usb/pwc -i Philips or
compatible webcam setup
#
# Viene caricato il modulo con opportune opzioni
# Raccomando la lettura di philips.txt nei sorgenti del modulo
# per una dettagliata descrizione delle stesse, si noterà comunque che
# si tratta delle stesse inserite in /etc/modules.conf
# seguendo il tutorial allegato all'elaborato
#
/sbin/modprobe pwc dev_hint=740."Logitech QuickCam
Zoom":63,740."Logitech QuickCam 4000":62,740."Philips 740
webcam":61,20,21,22 fps=10 le
ds=250,750
#
# Parte opzionale, nel caso venga trovato pwcx-i386.o nel percorso
# indicato (/etc/hotplug/usb) esso viene installato successivamente
# al pwc ed abilita quindi la decompressione
#
```

```

if [ -f /etc/hotplug/usb/pwcx-i386.o ]; then
    /usr/bin/logger -p daemon.notice -t hotplug/usb/pwc -i pwcx-i386.o
insmod
    /sbin/insmod --force /etc/hotplug/usb/pwcx-i386.o >/dev/null 2>&1
fi
#
# Notate che ad insmod viene passato il parametro --force perché
# pwcx è disponibile soltanto in versione precompilata e difficilmente
# sarà stato creato su versioni del kernel perfettamente combacianti
# con la propria, senza il parametro il caricamento fallirebbe con un
# errore di "version mismatch"
#
# Viene loggato il successo nel caricamento
#
/usr/bin/logger -p daemon.notice -t hotplug/usb/pwc -i Philips webcam
complete

```

Se aveste la necessità di utilizzare parametri particolari al caricamento del modulo, come ad esempio cambiare il livello di compressione utilizzato, la frequenza o altre cose potrete trovare informazioni dettagliate nel file `philips.txt` contenuto in `pwc_modified` sul cd allegato o potete dare

### ***modinfo pwc***

per una visione sintetica dei parametri.

## 3 Configurazione software

### 3.1 Installare Vgrabbj

E' giunto ora il momento di installare il software di cui abbiamo bisogno:

Portatevi nella directory corrispondente al cdrom, si dovrà ora installare vgrabbj:

Se disponete di una distribuzione basata su rpm (RedHat, Suse, Mandrake...) raccomando di utilizzare l'rpm fornito:

come root date il comando :

```
rpm -ivh vgrabbj-0.9.5-1.i586.rpm
```

Se vi venisse segnalata la mancanza delle librerie libpng raccomando di installarle dai propri cd della distribuzione.

Se invece aveste bisogno delle ftplib esse sono disponibili sul cd, copiate il file [ftplib-3.1-src.tar.gz](#) in una directory di vostra scelta, entrate in essa e scompattate con

```
tar xvfz ftplib-3.1-src.tar.gz
```

entrate nella directory ftplib così creata e date in successione

```
./configure
```

```
make
```

passate a root

```
make install
```

Se usate un altro tipo di distribuzione;

copiate in una directory a scelta [vgrabbj-0.9.5\\_binaries.tar.gz](#), entrate nella directory e date:

```
tar xvfz vgrabbj-0.9.5\_binaries.tar.gz
```

entrate nella directory [vgrabbj-0.9.5](#) che si sarà creata e da root date:

```
make install
```

Se intendete compilare vgrabbj dai sorgenti scompattate [vgrabbj-0.9.5.tar.gz](#) in una directory di vostra scelta e, una volta entrati date come root:

```
./configure
```

```
make
```

```
make install
```

Sono indispensabili le librerie libpng, nel caso vi venisse segnalata l'assenza assicuratevi che esista nei percorsi indicati in /etc/ld.so.conf il file libpng.so, nel caso non lo trovaste pur avendo libpng installato cercate [libpng.so.xx](#) e create un link simbolico con

***ln -s [libpng.so.xx](#) libpng.so***

oppure se si trovasse in un percorso non contenuto potete modificare il file /etc/ld.so.conf o esportare la variabile LD\_LIBRARY\_PATH.

Il configure potrebbe segnalare la mancanza delle librerie freetype o di ttf, installate le librerie freetype contenute in ogni distribuzione assicurandovi però che gli include contenuti normalmente in /usr/include/freetype siano quelli relativi ad una versione della serie 1 e non della serie 2 altrimenti viene rilevata la presenza ma si ottiene un errore in compilazione. Inutile dire che dovrete avere freetype nella versione 1.x.x installato eventualmente contemporaneamente alla versione 2. Probabilmente dovrete anche creare un link simbolico con nome /usr/lib/libtff.so che punti a /usr/lib/libtff.so.x. Tutta questa procedura è indispensabile perché il programma compili con supporto al timestamp.

Se necessitate inoltre del supporto al trasferimento tramite ftp (il nostro caso) dovrete necessariamente avere installate le librerie ftplib la cui installazione è illustrata poco sopra.

Segnalo che la versione da noi compilata ha funzionato senza intoppi su ben 4 distribuzioni diverse pertanto ritengo che difficilmente si renda necessaria la compilazione manuale.

### ***3.1a Installare i file di configurazione***

Installato vgrabbj dobbiamo ora posizionare i file di configurazione relativi: nella directory /usr/local/etc/ troverete un file vgrabbj.conf di esempio, il file è ampiamente commentato e di facile comprensione rendendo facile la modifica.

La nostra scelta è stata quella di creare diversi file di configurazione per ogni webcam richiamando di volta in volta quello corretto.

Spostatemi nella directory del cd e date:

***cp -r /etc/vgrabbj /etc***

in modo da copiare i necessari file di configurazione.

### ***3.2 Installare setpwc***

Posizionatevi nella directory del cd e compilate Software/setpwc in /usr/local/bin

Se volete installare il tool dai sorgenti scompattate il file [setpwc-0.4.tar.gz](#) e date:

***make***

dopodichè potete copiarlo nella directory indicata.

### **3.2a Installare i file di configurazione**

Copiate l'intera directory CDROM/etc/setpwc entrando nella directory del cdrom e dando  
***cp -r etc/setpwc /etc***

Nel caso si vogliano dei parametri diversi per il bilanciamento cromatico o del bianco non si dovrà far altro che editare i file /etc/setpwc/XXXX\_conf e XXXX\_night\_conf, si tratta di file di testo che banalmente contengono lo opzioni da passare a setpwc per i vari dispositivi, raccomando di non cancellare la riga nella forma -d /dev/videoXX perchè essa indica il device su cui le modifiche saranno apportate, per la lista dei parametri possibili date

***/usr/local/bin/setpwc -h***

o, se desiderate informazioni più precise collegatevi a [www.smcc.demon.nl/webcam](http://www.smcc.demon.nl/webcam) ovvero il sito del modulo dato che le opzioni possibili per setpwc sono le stesse ed hanno lo stesso significato di quelle accettate al caricamento dal modulo pwc.

Nulla vieta tra l'altro di inserire nel /etc/modules.conf una riga di caricamento per pwc con parametri del tutto generici e di variarli successivamente tramite setpwc, le uniche opzioni realmente indispensabili sono quelle relative a "dev\_hint" che assegna i device alle webcam appropriate, si è scelto di aggiungere poche altre opzioni di base per garantire un funzionamento minimale accettabile ad eventuali altre webcam collegate provvisoriamente.

### **3.3 Installare gli script**

Inoltre copiate webcamstart.sh e webcamstart\_night.sh in /usr/local/bin

Ora copiate webcam\_sysV\_start.sh e no\_pc.sh in /etc/rc.d/init.d/ e create due link nei runlevel 3 e 5 con i comandi:

***ln -s /etc/rc.d/init.d/webcam\_sysV\_start.sh /etc/rc.d/rc3.d/S99webcam\_sysV***

e

***ln -s /etc/rc.d/init.d/webcam\_sysV\_start.sh /etc/rc.d/rc5.d/S99webcam\_sysV***

in questo modo le webcam verranno attivate all'avvio del sistema sia in modalità testo che grafica attivando le giuste configurazioni a seconda dell'ora del giorno in cui si attiva il sistema.

Inoltre date

***ln -s /etc/rc.d/init.d/no\_pc.sh /etc/rc.d/rc3.d/K08no\_pc.sh***

e

***ln -s /etc/rc.d/init.d/no\_pc.sh /etc/rc.d/rc5.d/K08no\_pc.sh***

in tal modo allo spegnimento del pc sia che sia stato avviato in modalità testo o grafica per prima cosa il sistema trasmetterà l'immagine che attesta l'indisponibilità del pc al

server, tale immagine come, come quella relativa all'assenza di una o più delle webcam si deve trovare in /etc/vgrabbj/Images/ a meno che non si decida di cambiare il percorso anche in /etc/np\_pc.txt e /etc/arlX\_off.txt

Riporto sotto lo script relativo, raccomando di cambiare le ore di default per la distinzione fra ore diurne e notturne, al momento vengono ritenute diurne le ore comprese fra le 7:00 e le 18:00, si può modificare le impostazioni semplicemente editando il file che è commentato e di facile comprensione.

Come consiglio generico sarà bene dare anche una scorsa al file /etc/crontab, è risaputo che il demone crond digerisce piuttosto a fatica l'avvio di task periodici contemporanei, si raccomanda perciò di assicurarsi che, nei limiti del possibile, non esistano righe con i primi 5 campi tutti uguali nel file summenzionato.

```
#!/bin/bash
#
ORA=`date '+ %-H'`
MIN=`date '+ %-M'`
let ORA=ORA*100
TEMPO=${ORA+$MIN}
#
# Nella riga che segue sostituire inserire al posto del numero di
# 4 cifre l'ora di inizio giornata nel formato HHMM, per inizio
giornata
# si intende l'ora in cui si vuole che le webcam inizino ad utilizzare
# i settaggi per la luce del giorno
#
if [ $TEMPO -ge 0000 ];
#
# Inserire nello stesso formato in cui si vuole terminare con i
settaggi
# da giorno e passare a quelli notturni.
#
    then if [ $TEMPO -le 2400 ];
#
# Viene eseguito lo script appropriato a seconda dell'ora del giorn
#
        then /usr/local/bin/webcamstart.sh
        fi
        else /usr/local/bin/webcamstart_night.sh
```

fi

Attenzione però che questo file è chiamato soltanto all'avvio del sistema pertanto le impostazioni orarie servono soltanto affinché si sappia con cosa partire, per il funzionamento normale e continuato è indispensabile cambiare le impostazioni del file /etc/crontab, se avete eseguito Install.sh vedrete che esso contiene in fondo due righe relative al passaggio da un tipo di settaggi ad un altro, la prima è relativa al passaggio da notte a giorno, la seconda il viceversa, attualmente troverete impostate le ore 7:00 e 18:00 rispettivamente, non dovete far altro che modificarle ricordando che la prima cifra è quella relativa ai minuti e la seconda quella relativa all'ora nel formato a 24 ore.

Se modificate il crontab è necessario riavviare il demone crond dando il comando:

***service crond restart***

in modo che la nuova configurazione venga letta.

Gli script di attivazione delle webcam (webcamstart.sh e webcamstart\_night.sh) riportati sotto

webcamstart.sh

```
#!/bin/bash
#
BINDIR=/usr/local/bin
CONFDIR=/etc/vgrabbj
SETPWCCONF=/etc/setpwc
#
#
#
# Prima di tutto uccidiamo ogni eventuale istanza di vgrabbj
# in esecuzione
#
killall -9 vgrabbj
#
# Lo script elenca tutti i dispositivi del tipo /dev/video
#
cd /dev
    for i in $(ls video*); do
```



```

        echo item: $i
#
# Le righe successive verificano l'eventuale esistenza dei
# dispositivi /dev/video61,62,63 corrispondenti rispettivamente
# alla Philips toUcam, Logitech QuickCam Zoom e Logitech 4000 pro.
# Se trova tali dispositivi avvia setpwc per settare la webcam
# in questione con gli opportuni parametri dopodi che avvia
# vgrabbj per l'acquisizione con
# un proprio file di configurazione che può essere adattato
# alle proprie esigenze tenendo conto delle condizioni di funzionamento
# di ognuna delle webcam.
#
        if [ "$i" = "video61" ];
            then setpwc -d /dev/$i `cat
$SETPWCCONF/toUcam_conf`
            $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_toUcam.conf

        elif [ "$i" = "video62" ];
            then setpwc -d /dev/$i `cat $SETPWCCONF/4000_conf`
            $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_4000.conf

        elif [ "$i" = "video63" ];
            then setpwc -d /dev/$i `cat $SETPWCCONF/Zoom_conf`
            $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_Zoom.conf
#
# Nel caso venisse individuati ulteriori dispositivi
# non corrispondente ai tre visti si provvede ad avviare
# ugualmente vgrabbj con una configurazione apposita se trovata o di
default
# che abilita per altro il bilanciamento automatico del bianco
# Si raccomanda, nel caso i dispositivi vengano installati
# in pianta stabile di aggiungere uan riga analoga alle precedenti
# e di creare un proprio file di configurazione.
#

```

```
        elif [ -f $CONFDIR/vgrabbj_${i}.conf ];
            then $BINDIR/vgrabbj -c $CONFDIR_${i}.conf
        else $BINDIR/vgrabbj -c $CONFDIR/vgrabbj_default.conf -d
/dev/${i} -f /root/ar1_${i}.jpg
        fi
    done
```

webcamstar\_night.sh

```
#!/bin/bash
#
BINDIR=/usr/local/bin
CONFDIR=/etc/vgrabbj
SETPWCCONF=/etc/setpwc
#
#
#
# Prima di tutto uccidiamo ogni eventuale istanza di vgrabbj
# in esecuzione
#
killall -9 vgrabbj
#
# Lo script elenca tutti i dispositivi del tipo /dev/video
#
cd /dev
    for i in $(ls video*); do
        echo item: $i
    #
    # Le righe successive verificano l'eventuale esistenza dei
    # dispositivi /dev/video61,62,63 corrispondenti rispettivamente
    # alla Philips toUcam, Logitech QuickCam Zoom e Logitech 4000 pro.
    # Se trova tali dispositivi avvia setpwc con il relativo file di
    # configurazione per settare la telecamera correttamente dopodichè
    # avvia vgrabbj per l'acquisizione con
    # un proprio file di configurazione che può essere adattato
    # alle proprie esigenze tenendo conto delle condizioni di funzionamento
```

```

# di ognuna delle webcam.
#
  if [ "$i" = "video61" ];
    then if [ -f $SETPWCCONF/toUcam_night_conf ]
      then setpwc `cat $SETPWCCONF/toUcam_night_conf`
        $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_toUcam_night.conf
      else setpwc `cat $SETPWCCONF/toUcam_conf`
        $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_toUcam.conf
      fi

    elif [ "$i" = "video62" ];
      then if [ -f $SETPWCCONF/4000_night_conf ]
        then setpwc `cat $SETPWCCONF/4000_night_conf`
          $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_4000_night.conf
        else setpwc `cat $SETPWCCONF/toUcam_conf`
          $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_4000.conf
        fi

      elif [ "$i" = "video63" ];
        then if [ -f $SETPWCCONF/Zoom_night_conf ]
          then setpwc `cat $SETPWCCONF/Zoom_night_conf`
            $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_Zoom_night.conf
          else setpwc `cat $SETPWCCONF/toUcam_conf`
            $BINDIR/vgrabbj -c
$CONFDIR/vgrabbj_Zoom.conf
          fi

#
# Nel caso venisse individuati ulteriori dispositivi
# non corrispondente ai tre visti si provvede ad avviare
# ugualmente vgrabbj con una configurazione apposita se trovato
# un file di tipo $CONFDIR/vgrabbj_videoXX.conf o di default
# che abilita per altro il bilanciamento automatico del bianco
# Si raccomanda, nel caso i dispositivi vengano installati

```

```

# in pianta stabile di aggiungere una riga analoga alle precedenti
# e di creare un proprio file di configurazione.
#
    elif [ -f $CONFDIR/vgrabbj_"$i"_night.conf ]
        then $BINDIR/vgrabbj -c $CONFDIR/vgrabbj_"$i"_night.conf
    elif [ -f $CONFDIR/vgrabbj_"$i".conf ];
        then $BINDIR/vgrabbj -c $CONFDIR/vgrabbj_"$i".conf
    else $BINDIR/vgrabbj -c $CONFDIR/vgrabbj_default.conf -d /dev/$i
-f /root/arl_$i.jpg
    fi
done

```

sono opportunamente creati per l'attivazione delle webcam in dotazione al laboratorio, ciononostante permettono il funzionamento di un numero imprecisato di webcam leggendo all'occorrenza dei file di configurazione nella forma `vgrabbj_videoXX.conf` o `vgrabbj_videoXX_night.conf` da posizionare sempre nella directory `/etc/vgrabbj` e relativi ai dispositivi `/dev/videoXX` dove naturalmente `XX` è un numero di 1 o 2 cifre. (I dispositivi video possono essere al massimo 64). Nel caso tali file di configurazione non fossero stati creati viene utilizzato un file di configurazione di default, tale soluzione però ha senso soltanto per motivi di test dato che se in locale le immagini prodotte vengono differenziate in base al dispositivo in remoto tramite ftp ciò non è possibile dato che `vgrabbj` non accetta parametri relativi all'ftp da riga di comando. L'aggiunta di questa opzione è stata in particolar modo pensata per l'utilizzo provvisorio a scopo di test di alcune ulteriori webcam, in caso di necessità rimane da preferirsi la soluzione di aggiungere agli script delle righe analoghe a quelle specifiche per le 3 webcam di partenza.

In aggiunta a quanto fatto da `webcamstart.sh` il `webcamstart_night.sh` verifica se per i dispositivi sia disponibile una configurazione relativa alla notte e se essa è reperibile li attiva con essa altrimenti ricade nella configurazione per il giorno.

**!!ATTENZIONE:** Il sistema configurato secondo le nostre istruzioni riconosce l'aggiunta a caldo di una o più webcam ciononostante essa non verrà attivata fino all'attivazione da parte di `crond` di uno dei due script: se si desidera l'abilitazione immediata non si deve far altro che far partire uno dei due a seconda dell'ora del giorno o della notte o eseguire nuovamente

*`/etc/rc.d/init.d/webcam_sysV_start.sh`*

o più semplicemente

*`service webcam_sysV_start.sh restart`*

Tuttavia consiglieri, per evitare eventuali conflitti, di fermare preventivamente ogni

istanza di `vgrabbj` in corso con il comando:

***killall -9 vgrabbj***

e solo dopo inserire il dispositivo questo perchè non è da escludersi che la chiamata al modulo per il riconoscimento della nuova webcam, arrivando contemporaneamente ad un accesso in lettura agli altri, possa bloccare il modulo stesso in maniera analoga a quella sperimentata all'avvio di più istanze di `xawtv`.

In generale ho avuto modo di notare che il modulo, ovviamente ottimizzato per gestire una sola webcam, gestisce in maniera approssimativa gli accessi concorrenti, nel nostro caso non c'è ragione di supporre che si presenteranno difficoltà dato che due collaudi della durata di 4 e 6 ore rispettivamente non hanno avuto difficoltà a concludersi con successo. Certo non mi sento di escludere una possibile instabilità all'aumentare del numero di webcam gestite nonché all'incremento della frequenza di scatto.

Il problema più grosso riscontrato, imputabile secondo esperienze precedenti al sistema NFS, consiste in casuali blocchi dell'intero computer a seguito di problemi anche temporanei nella comunicazione di rete. Il pc rimane attivo ma non è possibile il login, inutile dire che tali difficoltà bloccano anche il sottosistema preposto alla gestione delle webcam visto che tutti i processi di tutti gli utenti (compreso quindi root) risultano in quei casi bloccati. Purtroppo al verificarsi di simili situazioni non si vede altra soluzione che il riavvio.

## **!!ATTENZIONE**

Esiste uno script sul cd che si occupa di installare il modulo `pwcx`, `vgrabbj` e relativi file di configurazione, `setpwc` e relativi file di configurazione, sistema il crontab, installa tutti gli script necessari e crea gli opportuni collegamenti simbolici, se usate un kernel 2.4.21-0.13mdk come la Mandrake 9.1 o lo avete installato come è stato fatto in laboratorio installa anche la versione patchata di [pwc.o.gz](http://pwc.o.gz) nella corretta posizione. Inoltre se usate una distribuzione basata su rpm installa anche `vgrabbj`.

In altre parole su un sistema Mandrake 9.1 o su quello del laboratorio il lancio di **Install.sh** come root dovrebbe rendere il tutto già funzionale.

In caso contrario raccomando di seguire pedissequamente la procedura descritta.

Le uniche modifiche effettivamente necessarie sono quelle da apportare ai file di configurazione di `vgrabbj` locati in `/etc/vgrabbj` in modo da renderli confacenti alle proprie esigenze, discorso analogo per i file situati in `/etc/setpwc`. Potrebbe essere necessario cambiare i parametri di accesso dell'ftp o quelli di regolazione delle webcam in modo da ottenere la migliore resa visiva nelle condizioni definitive di funzionamento.

### 3.4 L'archivio

Come ultima cosa segnalo che la modifica del sistema per supportare la gestione di un archivio delle immagini potrebbe rivelarsi difficoltosa data la limitatezza dei comandi ftp, nel lavoro dei miei predecessori si gestiva un archivio di 12 immagini ed era stato pertanto creato un file di comandi ftp che ruotava le immagini in modo da identificarle sempre con la loro età. La cosa potrebbe essere fattibile nella stessa maniera anche adesso però ricordo che lo script di rotazione non potrebbe eseguirsi più spesso di una volta al minuto, per altro se si intendesse di mantenere una immagine al minuto per ognuna delle 3 webcam per tutta la giornata precedente si dovrebbero gestire 1440\*3 immagini, il file di configurazione conterrebbe migliaia di righe e verrebbero scambiati migliaia di messaggi ogni minuto mentre se la cosa fosse gestita direttamente sul pc destinatario tutto potrebbe essere risolto con uno script di poche righe ricorrendo a dei cicli. Raccomando quest'ultima soluzione che oltre a non caricare la rete si rivelerebbe efficiente perché i file system abitualmente utilizzati in unix non spostano effettivamente i file quando essi vengono rinominati ma si limitano a modificare il referente nel superblocco il quale è per altro in genere cachato in memoria.

Riporto sotto una possibile soluzione che sfrutta il primo metodo

```
FTProtate.sh
#!/bin/bash
#
# Tramite ftp accedo in autologin al server remoto ed eseguo i
  comandi
# ftp contenuti in ftpmacro.txt
#
ftp webmail.ing.unibs.it < /etc/ftpmacro.txt
```

Il file /etc/ftpmacro.txt

```
binary
cd www/docs/webcam
! ./Archive_rotate.sh
exit
#
# Viene lanciato sul server remoto lo script FTProtate, natura
  lmente
# esso deve essere preventivamente stato installato e deve pot
```

```

er essere
# interpretato, ilserver deve cioè disporre di una shell Bourn
e o
# preferibilmente Bourne Again (bash)

```

contiene come istruzioni quelle di portarsi nella directory di destinazione degli scatti e di avviare lo script Archive\_rotate.sh sulla macchina remota.

```

#!/bin/bash
#
# Nel caso che l'archivio sia situato in una posizione diversa
# modificate il path seguente
# FTPDIR deve contenere il percorso in cui si trovano
# le immagini scattate dalle webcam
#
FTPDIR=.
ARCHDIR=$FTPDIR/archivio
#
N=0
cd $ARCHDIR
#
# Eliminazione del file più vecchio
#
if [ -f arla_1440_minuti_fa.jpg ];
    then rm arla_1440_minuti_fa.jpg
fi
if [ -f arlb_1440_minuti_fa.jpg ];
    then rm arlb_1440_minuti_fa.jpg
fi
if [ -f arlc_1440_minuti_fa.jpg ];
    then rm arlc_1440_minuti_fa.jpg
fi
#
# Invecchiamento dei file dell'archivio
#
for j in `seq 1439 1`;
do

```

```

if [ -f ar1a-"$j"_minuti_fa.jpg ]; then
    let N=[ $j + 1 ]
    mv -f ar1a-"$j"_minuti_fa.jpg ar1a-"$N"_minuti_fa.jpg
fi
if [ -f ar1b-"$j"_minuti_fa.jpg ]; then
    let N=[ $j + 1 ]

```

che cancella l'immagine di più di 24 ore fa e sposta indietro tutte le altre contenute nell'archivio indicando correttamente il numero di minuti passati da quando sono state scattate.

Per il buon funzionamento di questi script e del metodo di archiviazione è indispensabile che il pc remoto disponga di una shell Bourne, che si copi sul pc remoto, nella directory di destinazione delle immagini lo script Archive\_rotate.sh contenuto in ftpscripts sul cd, che si crei la cartella di destinazione per l'archivio fotografico e si modifichino di conseguenza i percorsi nello script (di default si può creare una directory archivio sotto a quella in cui si troveranno gli scatti e lo script funzionerà).

Si deve copiare il file ftpscripts/ftpmacro.txt in /etc sul pc locale, ftpscripts/FTProtate.sh in una directory a piacere del pc locale, consiglio /usr/local/bin, aggiungere manualmente al file /etc/crontab la riga seguente:

```

* * * * * root PERCORSO_PER_FTPROTATE/FTProtate.sh

```

Nel nostro caso può andar bene

```

* * * * * root /usr/local/bin/FTProtate.sh

```

Una volta fatto ciò si dovrà riavviare il demone crond con il comando

***service crond restart***

e lo script si eseguirà regolarmente ogni minuto purchè naturalmente siano stati sistemati l'autologin e le cose sul pc di destinazione.

### ***3.5 Tutto più facile con Install.sh***

**Lo script Install.sh fa tutto ciò che è descritto precedentemente a parte configurare il pc remoto, modificare /etc/modules.conf (cosa rischiosa da far fare ad uno script) e aggiungere in crontab la riga relativa alla gestione dell'archivio.**

Per le operazioni non eseguite dallo script si deve naturalmente rimboccarsi le maniche così come per la configurazione dei parametri della telecamera.



Per quanto riguarda quest'ultima questione è indispensabile valutare caso per caso tenendo conto comunque che l'unico parametro veramente cruciale è la luminosità, in genere il bilanciamento automatico funziona piuttosto bene per le Logitech, naturalmente il sistema automatico tende a produrre un buon compromesso, se l'area di interesse copre una percentuale limitata dell'immagine ed è illuminata in maniera drasticamente diversa dal resto non si otterrà mai il risultato desiderato. Senza dubbio però il bilanciamento automatico potrebbe rivelarsi l'unico compromesso ragionevole nel caso che le condizioni ambientali siano molto variabili. Se non altro il bilanciamento automatico assorbe rapidamente cambiamenti come l'accensione di una luce frontale e simili.

**ATTENZIONE!** La Philips toUcam installata nel laboratorio sembra ribellarsi all'attivazione del bilanciamento automatico, nella fattispecie se esso viene attivato come prima opzione non ci sono problemi ma nel caso che si passino dei parametri manuali risponde soltanto ad altri parametri manuali e non consente il ripristino del comando automatico senza un riavvio o lo scollegamento dall'alimentazione elettrica della webcam. Non mi sento di escludere che il problema più che alla webcam sia dovuto al modulo ma sta di fatto che esso si presenta invariabilmente.

## 4 Conclusioni e potenzialità di sviluppo

Gli obiettivi prefissi sono stati raggiunti.

Il problema della frequenza di scatto è stato risolto adottando un software caricato come demone che si occupa di svolgere il proprio lavoro con una frequenza impostabile con precisione pari ad un millisecondo, in pratica l'unico limite è dato dalle capacità della webcam e del bus usb.

Per quanto riguarda l'archivio, si è dovuti ricorrere, principalmente per motivi di efficienza ad una soluzione non completamente server based, l'invasività del metodo adottato sembra comunque ragionevole.

Per una maggiore libertà di configurazione si è resa necessaria la modifica ad una sezione modulare del kernel di linux, l'operazione è comunque documentata e decisamente semplice sapendo dove intervenire.

L'unica cosa a cui non è stato possibile porre troppa attenzione è l'automatizzazione delle procedure di configurazione, del resto si presenta la necessità di fornire numerose informazioni, naturalmente si potrebbe pensare di produrre uno script in Perl o Python che faciliti le operazioni e configuri opportunamente tutti i file.

Vorrei segnalare l'esistenza di altri software validi per il nostro scopo, i più interessanti sono senz'altro:

**webcam:** che fa parte del pacchetto xawtv, le sue funzionalità sono pressochè le stesse di quelle offerte da vgrabbj ma in modalità temporizzata, equivalente al daemon mode di vgrabbj, era soggetto ad inspiegabili crash senza la segnalazione di alcun problema.

**Camstream:** programma solido e molto facile da usare che supporta anche molte funzionalità aggiuntive rispetto a vgrabbj ma, a detta dello stesso autore, non ne esiste né ne verrà sviluppata una versione a riga di comando, l'inseparabilità da un sottosistema grafico ci è sembrato un handicap molto gravoso visto che per le nostre esigenze era raccomandabile una struttura che l'utente non dovrebbe nemmeno percepire e che funzionasse in completa autonomia.

## Bibliografia

[Www.smcc.demon.nl/webcam](http://www.smcc.demon.nl/webcam) è il punto di riferimento principale per lo scaricamento del software e per la documentazione di partenza.

**WEB-camera per LDRA** di Alessandro Saetti e Sergio Spinoni da cui questo lavoro ha preso il via

[www.kernel.org](http://www.kernel.org) per versioni aggiornate del kernel GNU/Linux

[www.linux-usb.org](http://www.linux-usb.org) per informazioni e bugtracking di qualsiasi dispositivo usb linux

[vgrabbi.gecius.de/vgrabbi](http://vgrabbi.gecius.de/vgrabbi)

per la versione più recente di vgrabbi e documentazione sul programma

[www.linuxdoc.org](http://www.linuxdoc.org) per tutta la documentazione su Linux, in particolare per tutorial e suggerimenti nella creazione degli script

[nbpfaus.net/~pfau/ftplib/](http://nbpfaus.net/~pfau/ftplib/) per l'ultima versione delle ftplib

[www.libpng.org](http://www.libpng.org) per software e informazioni sulle librerie png

[www.freshmeat.net](http://www.freshmeat.net) per la ricerca di software per linux in genere

[www.bytesex.org/](http://www.bytesex.org/) per tutto ciò che riguarda video4linux e xawtv

**Autore:** Diego Zuccotti

Matricola 025174 Universitas Studiorum Brixiae

Corso di laurea: Ingegneria Elettronica

e-mail: [jeegsawk@libero.it](mailto:jeegsawk@libero.it)



# INDICE

<i>PARAGRAFO</i>	<i>PAG.</i>
1 OBIETTIVI.....	2
2 CONFIGURAZIONE SISTEMA.....	3
2.1 Acquisizione informazioni.....	3
2.2 Aggiornamento modulo.....	4
2.2a Aggiornamento del solo modulo.....	4
2.2b Aggiornamento kernel.....	5
2.3 Caricamento modulo e verifica.....	6
2.4 Installazione modulo modificato.....	7
3 CONFIGURAZIONE SOFTWARE.....	12
3.1 Installare <i>Vgrabbj</i> .....	12
3.1a Installare i file di configurazione.....	13
3.2 Installare <i>setpwc</i> .....	13
3.2a Installare i file di configurazione.....	14
3.3 Installare gli script.....	14
3.4 L'archivio.....	21
3.5 Tutto più facile con <i>Install.sh</i> .....	24
4 CONCLUSIONI E POTENZIALITÀ DI SVILUPPO.....	26
BIBLIOGRAFIA.....	27