XFree86 Video Timings HOWTO

Eric Steven Raymond

Thyrsus Enterprises (http://www.catb.org/~esr/)

esr@thyrsus.com

$Data: 2003/07/14 21:32:42 $

Diario delle Revisioni

Revisione 6.4 2003-07-14 Revisionato da: esr
Corretti URL.
Revisione 6.3 2003-02-22 Revisionato da: esr
Corretti URL.
Revisione 6.2 2002-02-03 Revisionato da: esr
Correzioni minori nella generazione automatica della "modeline".
Revisione 6.1 2001-10-29 Revisionato da: esr
Da notare che la modalità VESA arriva fino a 1920x1440.
Revisione 6.0 2001-08-09 Revisionato da: esr
Una spiegazione più chiara di DDC e EDID. Questo HOWTO è ora fondamentalmente obsoleto.
Revisione 5.0 2000-08-22 Revisionato da: esr
Prima versione DocBook.

Questo HOWTO è effettivamente obsoleto. Le versioni correnti di XFree86 (dalla 4.0.1 in poi) calcolano ottime "modeline" dalla risoluzione che specificate nella sezione "Modes" del vostro file di configurazione X.

Come comporre una "modeline" per la vostra combinazione scheda-video/monitor su XFree86. La distribuzione XFree86 ora ha reso più agevole la configurazione della maggior parte delle combinazioni standard; questo documento è particolarmente utile se state ottimizzando una vostra "modeline" personale per un monitor ad alte prestazioni o per dell'hardware poco comune. Vi darà anche una mano sull'uso dikvideogen per generare le "modeline", o xvidtune per modificare una modalità standard che non è proprio quella adatta per il vostro monitor. Traduzione a cura di Angelo Nardoni <`nardoni.a@TOGLIMI.libero.it`> e revisione a cura di Sandro Cardelli <`sacarde@TOGLIMI.tiscali.it`>.

Sommario

1. Disclaimer

L'uso del presente materiale è *unicamente a vostro rischio*. È possibile danneggiare sia il monitor che voi stessi quando ci si spinge oltre le specifiche del produttore. Leggete Sovraccaricare il Vostro Monitor per precauzioni più dettagliate. Qualsiasi danno a voi o al vostro monitor dovuto al sovraccarico è un problema vostro.

La più recente versione di questo HOWTO può essere trovata alla pagina web Linux Documentation Project (http://www.tldp.org/) .

Siete pregati di indirizzare commenti, critiche e suggerimenti per migliorie a <`esr@snark.thyrsus.com`>. Siete pregati di *non* inviare email chiedendo una soluzione magica al problema del vostro monitor speciale, perché così facendo consumate solo il mio tempo e frustrerà voi -- tutto quello che so sull'argomento è già qui dentro.

2. Perché Questo HOWTO è Obsoleto

Per le versioni 4.0.0 e successive di XFree86 non si devono più generare "modeline" nella maggioranza dei casi. Invece queste sono calcolate internamente dal server all'avvio, basate sulla risoluzione che avete specificato nella parte "Modes" della sezione "Screen" del vostro file di configurazione XFree86 e sulle capacità del monitor che il vostro server X ricava da una query EDID al monitor stesso.

Per cambiare la risoluzione video e la profondità del colore, semplicemente cambiate o create una sezione "Display" che li descriva. Ecco un esempio di descrizione "Screen" dal file di configurazione X del mio laptop:

```
Section "Screen"
 Identifier    "Screen0"
        Device        "ATI Rage Mobility"
        Monitor       "Monitor0"
 DefaultDepth 16

 Subsection "Display"
         Depth       16
                Modes       "1024x768"
 EndSubsection

EndSection
```

Tutto quello che generalmente dovrete fare sarà cambiare i valori nella riga "Modes". X farà il resto. Se specificate una risoluzione impossibile, il server non farà altro che tornare alla approssimazione più vicina ai dati che EDID dice che il monitor può sopportare.

Quindi, le informazioni a seguire di questo HOWTO sono utili solo se (a) avete un vecchio monitor pre-EDID, o (b) il driver della vostra scheda video non può fare interrogazioni al monitor, o (c) avete una vecchia versione di XFree86 (nel qual caso, potreste risolvere il problema aggiornandolo), o (d) la vostra accoppiata monitor/scheda video può sopportare una risoluzione superiore a 1920 x 1440 o inferiore a 640 x 480, che sarebbe la portata entro la quale XFree86 ha "modeline" scritte.

3. Introduzione

Il server XFree86 permette agli utenti di configurare i loro sottosistemi video e quindi incoraggia ad un miglior uso dell'hardware esistente. Questo documento intende aiutarvi ad imparare a generare i vostri propri valori di temporizzazione per sfruttare al meglio la vostra scheda video e il vostro monitor.

Presenteremo un metodo per ottenere qualcosa che funziona, quindi vi indicheremo come sperimentare partendo da questa base comune per sviluppare delle specifiche ottimizzate alle vostre esigenze.

Se già avete una modalità che funziona discretamente (in particolare, se una delle modalità VESA predefinite vi dà uno schermo stabile ma spostato troppo a sinistra o a destra, o troppo piccolo, o troppo largo) potete andare direttamente alla sezione Risolvere i Problemi con l'Immagine. Questa sezione vi svelerà le possibilità di manipolare i valori di temporizzazione per ottenere effetti particolari.

Non pensiate di dover studiare sin nei minimi particolari tutte le regolazioni delle "modeline" solo perché il vostro X vi appare completamente sballato la prima volta dopo l'installazione; può essere che la maggior parte delle "modeline" fornite dalla distribuzione siano OK e voi siete capitati in una che non si adatta al vostro hardware. Invece, provate tutte le vostre "modeline" con CTRL-ALT-KP+. Se qualcuna di queste sembra OK, provate a renderle inattive commentandole tutte eccetto una 640x480 e controllando che quella funzioni. Se funziona rendete attive scommentandole un paio di altre "modeline", ad esempio una 800x600 e una 1024x768 ad una frequenza che il vostro monitor dovrebbe essere in grado di gestire.

4. Strumenti per il Calcolo Automatico

Se il vostro monitor è stato fabbricato dopo il 1996, probabilmente sopporta le specifiche EDID (http://www.vesa.org/summary/sumeedidrar1.htm). I monitor EDID-compatibili (a volte chiamati "Plug'n'Play" nella letteratura del mercato Microsoft) possono trasmettere le loro capacità al vostro computer.

Diversi moduli driver in XFree86 4.0 supportano la DDC, la VESA Display Data Channel facility (http://www.vesa.org/dload/summary/sumeddcv1.htm). Il modulo di una scheda video DDC-compatibile chiederà al monitor di dargli una descrizione EDID delle sue capacacità e si configurerà in base a quei dati. Così con la 4.0 e un qualsiasi monitor recente, voi non dovrete fare nessuna configurazione.

Se vi capita di avere il modulo della scheda video non DDC-compatibile ma il vostro monitor parla EDID, potete ancora usare il programma read-edid per chiedere al monitor le sue specifiche e calcolare per voi una "modeline". Consultate la read-edid home page (http://john.fremlin.de/programs/linux/read-edid/).

Da XFree86 3.2 in poi, XFree86 fornisce un programma **XF86Setup** che rende facile generare interattivamente una modalità funzionante per il monitor, senza avere a che fare direttamente con i valori delle temporizzazioni video. Così il più delle volte non dovrete calcolare una modalità base per il monitor. Sfortunatamente, **XF86Setup** ha alcune limitazioni; riconosce solo modalità video standard fino a 1280x1024. Se avete un monitor ad alte prestazioni capace di 1600x1200 o più vi dovrete sempre calcolare la modalità base da soli.

C'è un'applicazione KDE chiamata KVideoGen (http://paranoia.rulez.org/videogen/) che genera le "modeline" dalle caratteristiche del monitor e della scheda video. Ho provato a generare "modeline" con questa applicazione, ma non le ho provate. Notate che i suoi parametri di refresh orizzontale e verticale sono gli stessi delle frequenze di sincronismo HSF e VSF che descriviamo in seguito. L'impulso di sincronismo orizzontale sembra essere una larghezza di impulso di sincronia in microsecondi, HSP (con l'applicazione che dà per fissi il 'piedistallo frontale' HGT1 e il 'piedistallo secondario' HGT2). Se non conoscete il valore dell'impulso di sincronia orizzontale è più sicuro usare quello di default.

Esiste un altro generatore di "modeline" XFree86 qui (http://zaph.com/Modeline). Potete sia scaricare lo script Python o usare il form CGI messo a disposizione.

Versioni recenti di XFree86 offrono uno strumento chiamato xvidtune che troverete probabilmente abbastanza utile per testare e perfezionare le modalità video. Parte con uno spaventoso avviso sulle possibili conseguenze se si fanno errori usandolo. Se prestate un'accurata attenzione a questo documento e imparate cosa c'è dietro quei simpatici numeri nei riquadri di xvidtune, saprete usare effettivamente e con confidenza xvidtune.

Se avete xvidtune(1),potrete provare nuove modalità al volo, senza modificare i vostri file di configurazione per X o resettare il server X. Altrimenti, XFree86 vi permette di spostarvi con la tastiera fra differenti modalità definite in Xconfig (vedi XFree86.man per dettagli). Usate questa capacità per risparmiarvi problemi! Quando volete provare una nuova modalità, chiamatela in modo univoco e aggiungetela alla *fine* della vostra lista di combinazioni di tasti. Lasciate una modalità che sapete che funziona bene come default per poter tornare lì se la modalità che state provando non funziona.

Verso la fine di questo documento, includeremo uno script 'modeplot' che potrete usare per produrre un grafico analogico delle possibili "modeline". Ciò non è direttamente utile per generare "modeline", ma vi può aiutare per capire meglio le relazioni che le definiscono.

5. Come Funziona lo Schermo

Sapere come funziona lo video è fondamentale per capire quali valori mettere nei vari campi del file Xconfig. Questi valori sono usati al più basso livello di controllo dello schermo dal server XFree86.

Lo schermo genera un quadro con quello che considerate essere una serie di punti raster. I punti sono disposti da sinistra a destra per formare delle linee. Le linee sono disposte dall'alto in basso per formare il quadro. I punti emettono luce quando sono colpiti dal fascio elettronico dentro lo schermo. Per far sì che il fascio colpisca ciascun punto per un uguale periodo di tempo, il fascio è diretto sullo schermo con un percorso costante, detto raster.

Abbiamo detto "quello che considerate essere una serie di punti" perché questi punti raster non corrispondono propriamente ai fosfori. I fosfori sono molto più piccoli dei punti raster -- lo devono essere, altrimenti lo schermo mostrerebbe gravi effetti-moiré. I punti raster sono in realtà campioni del segnale analogico, e si vedono come una griglia di punti solo perché i picchi nel segnale sono molto regolari e spaziati con precisione.

Il percorso parte in alto a sinistra sullo schermo, attraversa lo schermo verso destra con una linea diritta, muovendosi leggerissimamente "in discesa" (lo spostamento verticale è troppo esiguo da essere percepito). Poi il fascio è riportato indietro sul lato sinistro, ripartendo con una nuova linea. La nuova linea va da sinistra a destra proprio come la prima. Questo percorso è ripetuto fino a che l'ultima linea in fondo allo schermo è stata passata. Allora il fascio viene portato dall'angolo in basso a destra (spazzando avanti e indietro un pò di volte) all'angolo in alto a sinistra, e il percorso viene ripetuto.

C'è una variante di questo schema conosciuto come interlacciamento: qui solamente ogni linea pari è passata ogni mezzo quadro e quelle dispari sono tracciate durante il secondo mezzo quadro.

Quando il fascio parte dall'angolo superiore sinistro dello schermo è detto inizio di quadro. Il quadro termina quando il fascio raggiunge l'angolo superiore sinistro di nuovo venendo dall'angolo inferiore destro dello schermo. Un quadro è composto da tutte le linee tracciate dal fascio dall'alto in basso dello schermo.

Se il fascio degli elettroni funzionasse per tutto il tempo che attraversa il quadro, tutti i punti dello schermo sarebbero accesi. Non ci sarebbero bordi neri ai lati dello schermo. Ai lati dello schermo il quadro sarebbe distorto perché è difficile controllare il fascio in quei punti. Per ridurre la distorsione, i punti vicino ai lati dello schermo non sono illuminati dal fascio (perché sono spenti) anche se questo, se fossero accesi, li puntasse. L'area visibile dello schermo è ridotta in questo modo.

Un'altra cosa importante da capire è cosa succede del fascio quando nessuna cosa è disegnata nell'area visibile. Il tempo che il fascio avrebbe dovuto usare per illuminare i lati sinistro e destro dello schermo è invece usato per portare il fascio indietro da destra a sinistra e per muoverlo giù alla linea successiva. Il tempo che il fascio avrebbe dovuto usare per illuminare i lati superiore ed inferiore dello schermo è usato per muovere il fascio dall'angolo in basso a destra all'angolo in alto a sinistra

La scheda video genera i segnali che permettono allo schermo di accendere il fascio (del colore voluto) ad ogni punto per generare un quadro. La scheda controlla anche quando lo schermo muove il fascio da destra a sinistra e giù di una riga generando un segnale chiamato sync (per sincronismo) orizzontale. C'è un impulso di sincronismo orizzontale alla fine di ogni linea. La scheda genera anche un impulso di sincronismo verticale che segnala allo schermo quando muovere il fascio dall'angolo sinistro in alto dello schermo. Un impulso di sincronismo verticale è generato quasi alla fine di ogni quadro.

Lo schermo richiede che ci sia un breve lasso di tempo sia prima che dopo gli impulsi di sincronismo orizzontale e verticale così che si possa stabilizzare la posizione del fascio di elettroni. Se il fascio non può stabilizzarsi, il quadro non sarà nitido.

Per maggiori informazioni, consultate TV and Monitor Deflection Systems (http://fribble.cie.rpi.edu/~repairfaq/REPAIR/F_deflfaq.html).

In una sezione più avanti, torneremo su queste nozioni basilari con definizioni, formule ed esempi per aiutarvi ad usarle.

6. Cose Basilari da Sapere sul Vostro Schermo e la Scheda Video

Ci sono delle cose fondamentali da sapere prima di sperimentare con i valori di Xconfig. Sono:

- le opzioni di frequenza di sincronismo orizzontale e verticale del vostro monitor
- la larghezza di banda del vostro monitor
- le frequenze della vostra scheda video, o "dot clocks"

6.1. Le frequenze di sincronismo del monitor

La frequenza di sincronismo orizzontale è semplicemente il numero di volte al secondo che il monitor può scrivere una linea di scansione orizzontale; è la più importante informazione sul vostro monitor. La frequenza di sincronismo verticale è il numero di volte al secondo alla quale il monitor può far attraversare il fascio verticalmente sullo schermo.

Le frequenze di sincronismo di solito sono elencate nelle specifiche del manuale del monitor. Il valore della frequenza di sincronismo verticale è tipicamente calibrato in Hz (cicli per secondo), quello orizzontale in KHz (kilocicli per secondo). Le comuni gamme di valori sono fra 50 e 150Hz verticali, e fra 31 e 135KHz orizzontali.

Se avete un monitor multifrequenza, queste frequenze saranno date come gamma di valori. Alcuni monitor, specialmente quelli più scadenti, hanno svariate frequenze fisse. Anche queste possono essere configurate, ma le vostre possibilità saranno molto limitate dalle caratteristiche proprie del monitor. Scegliete il paio di frequenze più alte per la migliore risoluzione. E state attenti --- provare a mandare un monitor a frequenza fissa ad una velocità di clock più alta di quella per la quale è stato costruito lo può facilmente danneggiare.

Le prime versioni di questa guida erano piuttosto permissive sull'overclocking di monitor multifrequenza, spingendoli oltre la loro più alta frequenza verticale di sincronismo per ottenere migliori risultati. Da allora abbiamo più ragioni per essere cauti sull'argomento; le vedremo nel paragrafo Sovraccaricare il Monitor più avanti.

6.2. La larghezza di banda video del monitor

La larghezza di banda video del vostro monitor dovrebbe essere inclusa nelle pagine delle specifiche del manuale. Se non ci fosse, guardate la più alta risoluzione del monitor. Indicativamente, ecco come tradurla in larghezza di banda approssimativa (e quindi nei limiti superiori approssimati per il dot clock che potete usare):

```
640x480   25
800x600   36
1024x768  65
1024x768 interlaced 45
1280x1024  110
1600x1200  185
```

Per inciso, non c'è nulla di magico in questa tabella; questi numeri sono solo i più bassi dot clock per una data risoluzione nel database standard delle Modalità XFree86 (a parte l'ultimo, che ho interpolato io). La larghezza di banda del vostro monitor potrebbe essere più alta del minimo richiesto per la sua massima risoluzione, così non abbiate paura a provare un dot clock di pochi MHz più alto.

Notate anche che la larghezza di banda è raramente un limite per i dot clock sotto i 65 MHz o giù di lì. Con una scheda SVGA e la maggior parte dei monitor ad alta risoluzione, non potete andare troppo vicino ai limiti della larghezza di banda video del vostro monitor. Ecco alcuni esempi:

```
Marca    Larghezza di Banda Video
----------   ---------------
NEC 4D    75Mhz
Nano 907a   50Mhz
Nano 9080i   60Mhz
Mitsubishi HL6615  110Mhz
Mitsubishi Diamond Scan  100Mhz
IDEK MF-5117   65Mhz
IOCOMM Thinksync-17 CM-7126 136Mhz
HP D1188A   100Mhz
Philips SC-17AS   110Mhz
Swan SW617   85Mhz
Viewsonic 21PS   185Mhz
PanaSync/Pro P21  220Mhz
```

Anche i monitor di fascia bassa di solito non sono terribilmente legati alle loro date risoluzioni. Il NEC Multisync II è un buon esempio --- non può neanche raggiungere 800x600 per le sue specifiche. Può raggiungere solo 800x560. Per tali basse risoluzioni non vi servono alti dot clock o molta larghezza di banda; probabilmente il meglio che potete fare è 32Mhz o 36Mhz, entrambi non sono troppo lontani dalla larghezza di banda video del monitor di 30Mhz.

A queste due frequenze video, l'immagine sul vostro schermo potrebbe non essere così chiara come dovrebbe, ma certamente di qualità tollerabile. Naturalmente sarebbe stato meglio se il NEC Multisync II avesse avuto una larghezza di banda video più alta, tipo 36Mhz. Ma questo non è critico per impieghi comuni come redigere testi, in quanto le differenze non sono così grandi da causare forti distorsioni dell'immagine (i vostri occhi ve lo direbbero subito se così fosse).

6.3. La frequenza di clock della scheda video

La pagina delle specifiche del manuale della vostra scheda video di solito vi darà il valore massimo del dot clock della scheda (che sarebbe il numero totale dei pixel al secondo che può scrivere sullo schermo).

Se non avete questa informazione, ve la darà il server X. Tutte le versioni recenti dei server X supportano l'opzione --probeonly che rileva questa informazione ed esce senza neanche far partire X o senza cambiare la modalità video.

Se non avete -probeonly, non disperate. Anche se il vostro X vi blocca il monitor, scriverà una riga di valori di clock ed altre informazioni sul log di errori standard. Se voi ridirigete queste ad un file, saranno salvate anche se dovete resettare per tornare alla console.

Il risultato del tentativo o il messaggio di partenza di X dovrebbe essere qualcosa di simile ad uno dei seguenti esempi:

Se usate XFree86:

```
Xconfig: /usr/X11R6/lib/X11/Xconfig
(**) stands for supplied, (--) stands for probed/default values
(**) Mouse: type: MouseMan, device: /dev/ttyS1, baudrate: 9600
Warning: The directory "/usr/andrew/X11fonts" does not exist.
         Entry deleted from font path.
(**) FontPath set to "/usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/"
(--) S3: card type: 386/486 localbus
(--) S3: chipset:   924
                    ---
    Chipset -- questo è l'esatto tipo di chip; una delle prime maschere del 86C911

(--) S3: chipset driver: s3_generic
(--) S3: videoram:  1024k
                    -----
         Size of on-board frame-buffer RAM

(**) S3: clocks:  25.00  28.00  40.00   3.00  50.00  77.00  36.00  45.00
(**) S3: clocks:   0.00   0.00  79.00  31.00  94.00  65.00  75.00  71.00
                  ------------------------------------------------------
                              Possibili frequenze in MHz

(--) S3: Maximum allowed dot-clock: 110MHz
                                    ------
                                   Larghezza di banda
(**) S3: Mode "1024x768": mode clock =  79.000, clock used =  79.000
(--) S3: Virtual resolution set to 1024x768
(--) S3: Using a banksize of 64k, line width of 1024
(--) S3: Pixmap cache:
(--) S3: Using 2 128-pixel 4 64-pixel and 8 32-pixel slots
(--) S3: Using 8 pages of 768x255 for font caching
```

Se usate SGCS o X/Inside X:

```
WGA: 86C911 (mem: 1024k clocks: 25 28 40 3 50 77 36 45 0 0 79 31 94 65 75 71)
```

```
---  ------       -----         -------------------------------------------
 |     |            |                 Possibili frequenze in MHz
 |     |            +-- Quantità di RAM della scheda
 |     +-- Tipo di chip
 +-- Tipo di Server
```

Nota: fate questo senza processi in background sulla vostra macchina (del tutto se possibile). Poiché X è un'applicazione, le sue temporizzazioni possono collidere con l'attività del disco, rendendo i valori di cui sopra inaccurati. Fatelo svariate volte e osservate i valori che si stabilizzano; se non dovessero farlo, cominciate a fermare i processi finché non lo fanno. Il programma che gestisce il mouse, se ne avete uno, può in particolare rendere fasulla la lettura dei valori (cioè gpm per gli utenti Linux, mousemgr per gli utenti SVr4).

Per evitare la inaccuratezza delle prove sui clock, dovreste appuntarvi le temporizzazioni del clock e metterle nel vostro Xconfig come valore delle propietà dei Clocks --- questo evita il loop delle temporizzazioni e dà ad X una lista esatta dei valori di clock che può provare. Usando i dati dell'esempio precedente:

```
wga
 Clocks 25 28 40 3 50 77 36 45 0 0 79 31 94 65 75 71
```

Su sistemi con un carico molto variabile, questo vi potrebbe evitare misteriosi fallimenti di partenza di X. È possibile per X partire, ottenere le temporizzazioni sbagliate a causa del carico del sistema, e quindi non essere capace di trovare un clock uguale nel suo database di configurazione --- o trovare quello sbagliato!

6.4. Cosa controllano questi dati basilari

La gamma di frequenze di sincronismo del vostro monitor, insieme con il dot clock della scheda video, determinano la risoluzione definitiva che potete usare. Ma è compito del driver intercettare il potenziale del vostro hardware. Avere una combinazione hardware di qualità senza un device driver egualmente valido è come buttare i soldi. D'altra parte, con un device driver versatile insieme ad un hardware meno capace potete spingere un pochino le caratteristiche dell'hardware. Questa è la filosofia progettuale di XFree86.

Dovreste far combaciare il dot clock che usate con la larghezza di banda video del monitor. Ci sono molte possibilità, comunque --- alcuni monitor possono funzionare il 30% oltre la loro larghezza di banda nominale. Il rischio qui è eccedere la frequenza di sincronizzazione verticale massima del monitor; ne discuteremo in dettaglio più avanti.

Sapere la larghezza di banda vi permetterà di fare scelte più intelligenti fra le possibili configurazioni. Ciò può cambiare la qualità visiva dello schermo (specialmente la nitidezza nei dettagli).

7. Interpretare le Specifiche di Base

Questa sezione spiega cosa significano le specifiche precedenti, e alcune altre cose che dovete sapere. Prima, alcune definizioni. Accanto a ciascuna definizione la sigla che useremo per fare calcoli, fra parentesi:

frequenza di sincronismo orizzontale (HSF)

Scansioni orizzontali al secondo (vedi sopra).

frequenza di sincronismo verticale (VSF)

Scansioni verticali al secondo (vedi sopra). Principalmente importante come limite superiore della vostra frequenza di refresh.

dot clock (DCF)

Più formalmente, ’frequenza di clock’; la frequenza del quarzo o VCO della vostra scheda video --- il numero massimo dei punti-al-secondo che può emettere.

larghezza di banda video (VB)

La più alta frequenza che potete immettere nell’input del monitor e ancora aspettarvi di vedere qualcosa. Se la vostra scheda produce una trama di accensioni-spegnimenti alternati, la sua frequenza più bassa sarà metà del DCF, così in teoria la larghezza di banda comincia ad avere senso a DCF/2. Per una videata sufficientemente nitida di piccoli dettagli nell’immagine video, comunque, non la vorrete molto più bassa del vostro più alto DCF, e preferibilmente più alta.

larghezza e lunghezza di quadro (HFL, VFL)

La larghezza orizzontale del quadro (HFL) è il numero di impulsi del dot clock necessari al pennello elettronico del vostro monitor per scandire una linea orizzontale, *inclusi i bordi inattivi sinistro e destro*. La lunghezza di quadro verticale (VFL) è il numero di linee scandite nell’*intera* immagine inclusi i bordi inattivi superiore ed inferiore.

frequenza di refresh dello schermo (RR)

Il numero di volte al secondo che il vostro schermo viene ridisegnato (ciò è anche detta "frequenza di quadro"). Più sono alte le frequenze, meglio è, perché riducono lo sfarfallio. 60Hz va bene, lo standard VESA di 72Hz è meglio. Calcolatelo come

```
RR = DCF / (HFL * VFL)
```

Notate che il prodotto nel denominatore *non* è lo stesso della risoluzione visibile del monitor, ma tipicamente un po’ più grande. Andremo nel dettaglio più avanti.

Le frequenze che sono usualmente specificate per le modalità interlacciate (come 87Hz interlacciati) sono frequenze di metà quadro: un intero schermo sembra avere all’incirca quella frequenza per gli schermi standard, ma ogni singola linea è ridisegnata solo metà delle volte.

Per scopi di calcolo noi riconosciamo uno schermo interlacciato dalla sua frequenza a schermo intero (refresh), per esempio 43.5Hz. La qualità di una modalità interlacciata è migliore di quella di una modalità non-interlacciata alla stessa frequenza per il quadro intero, ma sicuramente peggio di quella non-interlacciata corrispondente alla frequenza di metà quadro.

7.1. Sulla larghezza di banda

Ai costruttori di monitor piace dichiarare ampie larghezze di banda perché questo aumenta la nitidezza dell’intensità e del cambio dei colori sullo schermo. Un’ampia larghezza di banda significa vedere dettagli più piccoli.

Il vostro monitor usa segnali elettronici per presentare un’immagine ai vostri occhi. Tali segnali arrivano sempre in forma d’onda una volta che sono convertiti in forma analogica da digitale. Possono essere considerati come combinazioni di molte forma d’onda più semplici ognuna delle quali ha una frequenza fissa, molte di esse sono nella gamma dei Mhz, per esempio, 20Mhz, 40Mhz, o anche 70Mhz. La larghezza di banda del vostro monitor è, effettivamente, la più alta frequenza di segnale analogico che può sopportare senza distorsioni.

Per i nostri scopi, la larghezza di banda video è principalmente importante come un approssimativo punto di partenza per il più alto dot clock che potete usare.

7.2. Frequenze di Sincronismo e la Frequenza di Refresh:

Ogni linea di scansione orizzontale sul display è solo la porzione visibile della scansione del quadro. Ad ogni istante c'è un solo punto acceso sullo schermo, ma con una frequenza di refresh abbastanza alta la persistenza della vista dei vostri occhi vi permette di "vedere" l'intera immagine.

Ecco alcuni disegni che possono aiutare:

```
 _________________________
|                         |     La frequenza di sincronismo orizzontale
|->->->->->->->->->->->-> |     è il numero di volte al secondo
|                        )|     al quale il pennello elettronico
|<-----<-----<-----<---   |     e del monitor può tracciare
|                         |     un disegno tipo questo
|                         |
|                         |
|                         |
|_________________________|
 _________________________
|        ^                |     La frequenza di sincronismo verticale
|       ^ |               |     è il numero di volte al secondo
|       | v               |     al quale il pennello elettronico
|       ^ |               |     del monitor può tracciare
|       | |               |     un disegno tipo questo
|       ^ |               |
|       | v               |
|       ^ |               |
|_______|_v_______________|
```

Ricordate che la scansione è un disegno a zigzag molto stretto; cioè, il pennello va da sinistra a destra e contemporaneamente su e giù.

Ora possiamo vedere come il dot clock e la grandezza dello schermo si mettono in relazione con la frequenza di refresh. Per definizione, un hertz (Hz) è un ciclo al secondo. Così, se la vostra larghezza di quadro orizzontale è HFL e la vostra lunghezza di quadro verticale è VFL, allora per coprire l'intero schermo servono (HFL * VFL) impulsi. Dal momento che la vostra scheda emette DCF impulsi al secondo per definizione, allora ovviamente il pennello elettronico del vostro monitor può andare sullo schermo da sinistra a destra e dall'alto in basso e indietro DCF / (HFL * VFL) volte al secondo. Questa è la frequenza di refresh del vostro schermo, perché è quante volte il vostro schermo può essere aggiornato (ovvero *"rinfrescato"*) ogni secondo!

Dovete capire questo concetto per ottenere una configurazione che medi la risoluzione con lo sfarfallio in un modo qualsiasi che rispetti i vostri bisogni.

Per coloro che capiscono meglio i disegni del testo, eccone uno:

```
        RR                                      VB
         |   min HSF                      max HSF |
```

```
         |    |             R1        R2  |      |
max VSF -+----|------------/----------/---|------+----- max VSF
         |    |:::::::::::/::::::::::/:::::\     |
         |    \::::::::::/::::::::::/:::::::\    |
         |     |::::::::/::::::::::/:::::::::|   |
         |     |:::::::/::::::::::/::::::::::\   |
         |     \::::::/::::::::::/::::::::::::\  |
         |      \::::/::::::::::/::::::::::::::| |
         |       |::/::::::::::/:::::::::::::::| |
         |        \/::::::::::/:::::::::::::::::\|
         |        /\:::::::::/:::::::::::::::::::|
         |       /  \:::::::/::::::::::::::::::::|\
         |      /    |:::::/:::::::::::::::::::::| |
         |     /     \::::/::::::::::::::::::::::| \
min VSF -+----/-------\--/-----------------------|--\--- min VSF
         |   /         \/                        |   \
         +--/----------/\------------------------+----\- DCF
           R1        R2  \                       |     \
                          min HSF                |    max HSF
                                                VB
```

Questo è un diagramma di modalità di un monitor generico. L'asse x del diagramma mostra la frequenza del clock (DCF), l'asse y rappresenta la frequenza di refresh (RR). La regione riempita del diagramma descrive le capacità del monitor: ogni punto entro questa regione è una possibile modalità video.

Le linee etichettate 'R1' e 'R2' rappresentano una risoluzione fissa (tipo 640x480); vogliono mostrare come una risoluzione può essere realizzata da svariate differenti combinazioni di dot clock e frequenza di refresh. La linea R2 rappresenterebbe una risoluzione più alta di R1.

Gli spazi in alto e in basso della regione valida sono semplici linee orizzontali che rappresentano i valori limite per la frequenza di sincronismo verticale. La larghezza di banda video è un limite superiore della frequenza di clock e quindi è rappresentato da una linea verticale che affianca la regione delle possibilità a destra.

In Tracciare le Capacità del Monitor troverete un programma che vi aiuterà a disegnare un diagramma tipo questo (ma molto più carino, con grafica X) per il vostro monitor. Quella sezione tratterà anche la parte interessante; la derivazione degli spazi attorno all'immagine risultante dai limiti sulla frequenza di sincronismo orizzontale.

8. Compromessi nel Configurare il Sistema

Un altro modo di guardare alla formula che abbiamo derivato prima è

```
DCF = RR * HFL * VFL
```

Cioè, il vostro dot clock è fisso. Potete usare questi punti al secondo per guadagnare sia frequenza di refresh, sia risoluzione orizzontale, sia risoluzione verticale. Se uno di questi aumenta, uno o entrambi gli altri devono diminuire.

Notate, comunque, che la vostra frequenza di refresh non può essere più grande della massima frequenza verticale di sincronismo del monitor. Così, per ogni monitor ad un dato dot clock, c'è un valore minimo di lunghezze di quadro sotto il quale non potete andare.

Nello scegliere i vostri parametri, ricordate: se mettete RR troppo basso, farete una smorfia per lo sfarfallio dello schermo. Tenetelo sopra i 60Hz. 72Hz è lo standard ergonomico VESA. 120Hz è la frequenza dei tubi fluorescenti negli U.S.A. (100MHz in Europa e altri posti con la corrente elettrica a 50 Hz); se siete sensibili a questi, ve ne dovete tenere al di sopra.

Lo sfarfallio è molto affaticante per gli occhi, anche se gli occhi umani sono adattabili e la tolleranza della gente ad esso è assai varia. Se siete di fronte al vostro monitor ad un angolo di 90°, su sfondo scuro e un buon colore di contrasto, e un'intensità da bassa a media, voi *potete* stare bene anche a 45Hz.

Il test è questo: aprite un xterminal con puro sfondo bianco e testo nero usando xterm -bg white -fg black e allargatelo così da coprire tutta l'area visibile. Ora regolate la luminosità del monitor a 3/4 del suo massimo, e girate la faccia lontano dal monitor. Provate a sbirciare il monitor (usando le più sensibili cellule della visione periferica). Se non avvertite alcuno sfarfallio o pensate che quello che c'è è tollerabile, allora la frequenza di refresh va bene per voi. Altrimenti è meglio che settiate una più alta frequenza di refresh, perché quel semi-invisibile sfarfallio vi affaticherà gli occhi da morire e vi farà venire il mal di testa, anche se lo schermo sembra OK a prima vista.

Per le modalità interlacciate, l'ammontare dello sfarfallio dipende su più fattori tipo la attuale risoluzione verticale e il contenuto della videata. Così sperimentate. Anche se non dovreste andare molto sotto agli 85Hz per metà quadro, comunque.

Così diciamo che avete scelto una frequenza di refresh minimamente accettabile. Nello scegliere il vostro HFL e VFL, avrete un po' di spazio di manovra.

9. Requisiti di Memoria

La RAM disponibile per il buffer di quadro può limitare la risoluzione che potete ottenere su schermi a colori o in scala di grigi. Probabilmente non è così importante su schermi che hanno solo due colori, bianco e nero senza gradazioni di grigio.

Per gli schermi a 256 colori, un byte di memoria video è richiesto per ogni punto visibile. Questo byte contiene l'informazione che determina quale mistura di rosso, verde e blu deve essere generata per quel punto. Per ottenere la quantità di memoria richiesta, moltiplicate il numero di punti visibili per linea per il numero di linee visibili. Per uno schermo con una risoluzione di 1024x768, questa sarebbe 1024 x 768 = 786432, che è il numero di punti visibili sullo schermo. Questo è anche, a un byte per punto, il numero di byte di memoria video che sono necessari sulla vostra scheda video.

Così, la memoria video richiesta sarà tipicamente (HR * VR)/1024 Kbytes di VRAM, arrotondati (sarebbero esattamente 768K nell'esempio precedente). Se avete più memoria di quella strettamente richiesta, ne avrete extra per il panning di schermo virtuale.

Comunque, se avete solo 512K sulla vostra scheda video, allora non potete usare questa risoluzione. Anche se avete un buon monitor, ma non abbastanza video RAM, non potete avvantaggiarvi della potenzialità del vostro monitor. D'altra parte, se la vostra SVGA ha un mega, ma il vostro monitor può visualizzare al massimo 800x600, allora l'alta risoluzione è per voi irraggiungibile (vedi Usare le Modalità Interlacciate per un possibile rimedio).

Non preoccupatevi se avete più memoria di quella richiesta; XFree86 la userà per permettervi di 'scrollare' l'area visibile (vedi il file di documentazione di Xconfig sul parametro dell'ampiezza dello schermo virtuale). Ricordate anche che una scheda con 512K byte di memoria non ha realmente 512,000 byte installati, ne ha 512 x 1024 = 524,288 byte.

Se usate X/Inside con una scheda video S3, e volete campare con 16 colori (4 bit per pixel), potete settare la profondità a 4 in Xconfig ed effettivamente raddoppiare la risoluzione che può gestire la scheda video. Le schede S3, ad esempio, normalmente fanno 1024x768x256. Potete farle fare 1280x1024x16 con profondità 4.

10. Calcolare le Ampiezze di Quadro

Attenzione: questo metodo è stato sviluppato per monitor multisync. Probabilmente funzionerà lo stesso con monitor a frequenza fissa, ma non ci sono garanzie!

Iniziate dividendo DCF per il vostro massimo HSF per ottenere la larghezza di quadro orizzontale.

Per esempio: supponiamo che avete un Sigma Legend SVGA con un dot clock a 65MHz, ed il vostro monitor ha una frequenza di scansione orizzontale di 55KHz. La quantità (DCF / HSF) è allora 1181 (65MHz = 65000KHz; 65000/55 = 1181).

Ecco il nostro primo assaggio di magia. Dovete arrotondare al più vicino multiplo di 8. Questo ha a che fare con il controller hardware VGA usato dalla SVGA e dalle schede S3; usa un registro a 8-bit, spostato a sinistra di 3 bit, per cui è in realtà una quantità di 11-bit. Altri tipi di schede tipo la ATI 8514/A può non aver bisogno di questo, ma non lo sappiamo e correggere non farà male. Così arrotondiamo la configurazione della larghezza orizzontale di quadro a 1176.

Questa configurazione (DCF / HSF arrotondato ad un multiplo di 8) è il minimo HFL che potete usare. Potete ottenere HFL più lunghi (e così, possibilmente, più punti orizzontali sullo schermo) settando l'impulso di sincronismo in modo da produrre un HSF più basso. Ma lo pagherete con una frequenza più bassa e sfarfallio più visibile.

In genere, è disponibile l'80% della larghezza orizzontale di quadro per la risoluzione orizzontale, la parte visibile della linea scandita orizzontale (questo permette, all'incirca, di avere i bordi e il tempo per il pennello di tornare indietro -- cioè, il tempo necessario al pennello di muoversi dal lato destro dello schermo al lato sinistro della prossima linea). In questo esempio, sarebbero 944 impulsi.

Ora, per ottenere il normale rapporto della visualizzazione dello schermo di 4:3, settate la risoluzione verticale a 3/4 della risoluzione orizzontale che avete appena calcolato. Per il nostro esempio, sarebbero 708 impulsi. Per ottenere il vostro attuale VFL, moltiplicatelo per 1.05 per ottenere 743 impulsi.

Il 4:3 non è una meraviglia della tecnica; nulla vi vieta di usare un rapporto a Sezione Aurea se questo vi permette di usare al meglio il vostro monitor. Questo rende conveniente calcolare l'altezza e la larghezza del quadro dalla diagonale, dovete solo moltiplicare la diagonale per 0.8 per ottenere la larghezza e per 0.6 per avere l'altezza.

Così, HFL=1176 e VFL=743. Dividendo 65MHz per il prodotto dei due ci dà una bella e salutare frequenza di refresh di 74.4Hz. Eccellente! Meglio dello standard VESA! E avrete 944x708 all'accensione, più dell'800 per 600 che probabilmente vi aspettavate. Proprio niente male!

Potreste anche aumentare la frequenza di refresh fino a circa 76 Hz, sfruttando il fatto che i monitor spesso possono sincronizzarsi orizzontalmente a circa 2khz più delle specifiche, e in qualche modo abbassando lo VFL (che sarebbe, prendere meno del 75% di 944 nell'esempio precedente). Ma prima di provare questa manovra di "sovraccarico", se lo fate, siate *sicuri* che i pennelli elettronici del vostro monitor possano sincronizzarsi verticalmente fino a 76 Hz. (Il popolare NEC 4D, ad esempio, non può. Arriva solo fino a 75 Hz VSF). (Vedi Sovraccaricare il Monitor per una discussione più generale su quest'argomento).

Fino a qui, il più è semplice aritmetica e fatti basilari sugli schermi raster. Proprio poca magia nera!

11. Magia Nera e Impulsi di Sincronismo

OK, ora avete calcolato i valori di HFL/VFL per il dot clock scelto, trovato una frequenza di refresh accettabile, e controllato di avere abbastanza VRAM. Adesso facciamo la vera magia nera -- vi serve di sapere quando e dove piazzare gli impulsi di sincronismo.

Gli impulsi di sincronismo effettivamente controllano le frequenze di scansione orizzontale e verticale del monitor. Gli HSF e VSF che avete tirato fuori dalla pagina delle specifiche del manuale sono nominali e approssimative frequenze massime di sincronismo. L'impulso di sincronismo nel segnale della scheda video dice al monitor quanto veloce deve andare.

Ricordate le due figure precedenti? Solamente parte del tempo richiesto per tracciare (raster-scanning) un quadro è usata per mostrare l'immagine visibile (per esempio la vostra risoluzione).

11.1. Sincronismo Orizzontale:

Secondo la precedente definizione, ci vogliono HFL impulsi per tracciare una linea di scansione orizzontale. Chiamiamo il numero di impulsi visibili (la risoluzione orizzontale dello schermo) HR. Quindi, ovviamente, HR < HFL per definizione. Per praticità, assumiamo che entrambi partano allo stesso istante come mostrato qui sotto:

```
|___ __ __ __ __ __ __ __ __ __ __ __ __
|_ _ _ _ _ _ _ _ _ _ _ _                |
|_______________________|_______________|_____
0                       ^               ^     unità: impulsi
                        |   ^       ^   |
                        HR  |       |  HFL
                        |   |<----->|   |
                        |<->|  HSP  |<->|
                        HGT1         HGT2
```

Ora, noi vorremmo piazzare un impulso di sincronismo di lunghezza HSP come mostrato sopra, ad esempio, fra la fine degli impulsi di clock per mostrare dati e la fine degli impulsi di clock per l'intero quadro. Perché così? Perché se possiamo ottenere ciò, allora l'immagine sullo schermo non si sposterà a destra o a sinistra. L'immagine sarà dove dovrebbe essere sullo schermo, coprendo all'incirca tutta l'area visibile del monitor.

In più, vogliamo circa 30 impulsi di "tempo di guardia" ad entrambi i lati dell'impulso di sincronismo. Questo è rappresentato da HGT1 e HGT2. In una configurazione tipica HGT1 != HGT2, ma se state facendo una configurazione da zero, vorrete far partire le vostre sperimentazioni con loro uguali (cioè, con l'impulso di sincronismo centrato).

Il sintomo di un impulso di sincronismo non centrato è che l'immagine appare a schermo con un bordo troppo largo e l'altro lato dell'immagine che si ripiega su se stessa, producendo un spesso bordo bianco e una striscia di "immagine fantasma" da quella parte. Un impulso verticale fuori misura può far girare il quadro come sui televisori quando non è regolato bene il sincronismo verticale (infatti, è lo stesso fenomeno in azione).

Se siete fortunati, le larghezze degli impulsi di sincronismo del vostro monitor saranno documentate sulle pagine delle specifiche del manuale. Altrimenti, ecco dove inizia la vera magia nera...

Dovete un po' provare, sbagliare e riprovare in questa parte. Ma il più delle volte, possiamo sicuramente supporre che un impulso di sincronismo è lungo dai 3.5 ai 4.0 microsecondi.

Ancora per praticità, diciamo che HSP è 3.8 microsecondi (che poi, detto fra noi, non è un cattivo valore di partenza quando si sperimenta).

Ora, usando la temporizzazione di 65Mhz come in precedenza, sappiamo che HSP è equivalente a 247 impulsi di clock (= 65 * 10**6 * 3.8 * 10^-6) [ricordate che M=10^6, micro=10^-6]

Ad alcuni costruttori piace specificare i loro parametri di quadro orizzontale come tempi piuttosto che come larghezza di punti. Potreste vedere i seguenti valori:

tempo attivo (HAT)

  Corrisponde a HR, ma in unità di tempo (di solito microsecondi). HAT * DCF = HR.

tempo di oscuramento (HBT)

  Corrisponde a (HFL - HR), ma in unità di tempo (di solito microsecondi). HBT * DCF = (HFL - HR).

piedistallo frontale (HFP)

  Questo è proprio HGT1.

sync time

  Questo è proprio HSP.

back porch (HBP)

  Questo è proprio HGT2.

11.2. Sincronismo Verticale:

Tornando alla figura precedente, come mettiamo i 247 impulsi di clock secondo quanto mostrato nel disegno?

Usando il nostro esempio, HR è 944 e HFL è 1176. La differenza fra i due è 1176 - 944=232 < 247! Ovviamente dobbiamo mettere un po' le cose a posto qui. Cosa possiamo fare?

La prima cosa è aumentare 1176 a 1184, e diminuire 944 a 936. Adesso la differenza = 1184-936= 248. Hmm, più vicino.

Poi, invece di usare 3.8, usiamo 3.5 per calcolare HSP; quindi, abbiamo 65*3.5=227. Sembra meglio. Ma 248 non è molto più grande di 227. Normalmente servono 30 impulsi o giù di lì fra HR e l'inizio di SP, e lo stesso fra la fine di SP e HFL. E DEVONO essere multipli di otto! Siamo bloccati?

No. Facciamo così, 936 % 8 = 0, (936 + 32) % 8 = 0 anche. Ma 936 + 32 = 968, 968 + 227 = 1195, 1195 + 32 = 1227. Hmm... questo non è male. Ma non è un multiplo di 8, così lo arrotondiamo a 1232.

Ma ora abbiamo un problema potenziale, l'impulso di sincronismo non è più messo in mezzo fra h e H. Tranquillamente, usando la nostra calcolatrice troviamo che 1232 - 32 = 1200 è anche lui un multiplo di 8 e (1232 - 32) - 968 = 232 che corrisponde ad usare un impulso di sincronismo di 3.57 microsecondi, ancora ragionevole.

In più, 936/1232 ~ 0.76 o 76%, che non è lontano dall'80%, così dovrebbe essere tutto a posto.

Inoltre, usando l'attuale lunghezza di quadro orizzontale, noi in pratica chiediamo al nostro monitor di sincronizzarsi a 52.7khz (= 65Mhz/1232) che è nelle sue capacità. Nessun problema.

Usando le regole che abbiamo summenzionato, 936*75%=702, questa è la nostra nuova risoluzione verticale. 702 * 1.05 = 737, la nostra nuova altezza di quadro.

La frequenza di refresh dello schermo = 65Mhz/(737*1232)=71.6 Hz. Questo è ancora eccellente.

Disegnare il layout dell'impulso di sincronismo verticale è simile:

```
|___ __ __ __ __ __ __ __ __ __ __ __ __ __
|_ _ _ _ _ _ _ _ _ _ _ _ _ _              |
|_________________________|________________|______
0                        VR               VFL      unità: impulsi
                          ^    ^        ^
                          |    |        |
                          |<->|<----->|
                           VGT    VSP
```

Facciamo partire l'impulso di sincronismo appena passata la fine degli impulsi di dati video verticali. VGT è il tempo di guardia verticale richiesto per l'impulso. La maggior parte dei monitor funzionano bene con un VGT di 0 (nessun tempo di guardia) e noi useremo questo valore nel nostro esempio. Pochi monitor abbisognano di due o tre impulsi di tempo di guardia, e di solito non fa male aggiungerli.

Tornando all'esempio: dal momento che per definizione della lunghezza di quadro, un impulso verticale è il tempo per tracciare un intera riga ORIZZONTALE, nel nostro esempio, questa sarà 1232/65Mhz=18.95us.

L'esperienza insegna che un impulso di sincronismo verticale dovrebbe rientrare nella gamma fra 50us e 300us. Come esempio usiamo 150us, che si traduce in 8 impulsi di clock verticale (150us/18.95us~8).

Alcuni fabbricanti preferiscono dare i loro parametri di quadro verticale come temporizzazioni piuttosto che ampiezza di punti. Potrete vedere i seguenti valori:

tempo attivo (VAT)

Corrisponde a VR, ma in millisecondi. VAT * VSF = VR.

tempo di oscuramento (VBT)

Corrisponde a (VFL - VR), ma in millisecondi. VBT * VSF = (VFL - VR).

piedistallo frontale (VFP)

Questo è proprio VGT.

sincronismo

Questo è proprio VSP.

piedistallo secondario (VBP)

Questo è come un secondo tempo di guardia dopo l'impulso di sincronismo verticale. Spesso è zero.

12. Riassumendo

Il file Xconfig "Table of Video Modes" contiene righe di numeri, essendo ogni riga una completa specificazione di una modalità di operazione X-server. I campi sono raggruppati in quattro sezioni, la sezione del nome, la sezione della frequenza di clock, la sezione orizzontale, e la sezione verticale.

La sezione del nome contiene un campo, il nome della modalità video specificata nel resto della riga. Questo nome è riferito nella riga "Modes" della sezione Graphics Driver Setup nel file Xconfig. Il campo nome può essere omesso se il nome di una riga precedente è lo stesso.

La sezione della frequenza di clock contiene solo il campo della frequenza (che noi abbiamo chiamato DCF) della riga modalità video. Il valore di questo campo specifica quale dot clock è stato usato per generare i numeri delle sezioni seguenti.

La sezione orizzontale consiste di quattro campi che specificano come ciascuna linea orizzontale deve essere generata. Il primo campo della sezione contiene il numero di punti per linea che saranno accesi per formare l'immagine (da noi chiamati HR). Il secondo campo (SH1) indica in quale punto partirà l'impulso di sincronismo orizzontale. Il terzo campo (SH2) indica a quale punto finirà l'impulso di sincronismo orizzontale. Il quarto campo specifica la larghezza totale del quadro (HFL).

Anche la sezione verticale contiene quattro campi. Il primo campo contiene il numero di linee visibili che appariranno sullo schermo (VR). Il secondo campo (SV1) indica a quale linea partirà l'impulso di sincronismo verticale. Il terzo campo (SV2) specifica a quale linea finirà l'impulso verticale di sincronismo. Il quarto campo contiene il totale dell'altezza del quadro (VFL).

Esempio:

```
#Nome Mod.    clock  temporiz. oriz.   temporiz. vert.

"752x564"      40    752 784  944 1088  564 567 569 611
       44.5   752 792  976 1240  564 567 570 600
```

(Nota: X11R5 non supporta i dot clocks frazionari.)

Per Xconfig, tutti i numeri che abbiamo appena menzionato - il numero dei punti accesi sulla riga, il numero dei punti che separano i punti accesi dall'inizio dell'impulso di sincronismo, il numero dei punti che rappresentano la durata dell'impulso, e il numero di punti dopo la fine dell'impulso di sincronismo - sono sommati per produrre il numero di punti per riga. Il numero di punti orizzontali deve sempre essere divisibile per otto.

Esempio di numeri orizzontali: 800 864 1024 1088

Questa linea d'esempio ha il numero di punti accesi (800) seguito dal numero del punto da dove parte l'impulso di sincronismo (864), seguito dal numero del punto quando lo stesso impulso termina (1024), seguito dal numero dell'ultimo punto sulla riga (1088).

Notate di nuovo che tutti i valori dei numeri orizzontali (800, 864, 1024, e 1088) sono divisibili per otto! Ciò non è richiesto per i valori verticali.

Il numero di righe dall'alto in basso formano il quadro. Il segnale di temporizzazione base per un quadro è la riga. Un certo numero di righe formeranno l'immagine. Dopo che l'ultima riga accesa è stata disegnata, ci sarà un ritardo di un certo numero di righe prima che sia generato l'impulso di sincronismo verticale. Quindi l'impulso di sincronismo durerà per un po' di righe, e finalmente saranno generate le ultime righe del quadro, cioè il ritardo necessario dopo l'impulso. I valori che specificano questa modalità d'operazione sono immessi in una maniera simile al seguente esempio.

Esempio di valori verticali: 600 603 609 630

Questo esempio mostra che ci sono 600 righe visibili sullo schermo, che l'impulso di sincronismo verticale parte alla 603esima riga e termina alla 609esima, e che sono usate un totale di 630 righe.

Notate che i valori verticali non devono essere divisibili per otto!

Torniamo all'esempio dove stavamo lavorando. Secondo quanto detto prima, tutto quello che dobbiamo fare da adesso in poi è di scrivere i nostri risultati in Xconfig in questo modo:

```
<name>    DCF      HR   SH1  SH2   HFL   VR   SV1  SV2 VFL
```

dove SH1 è il punto di partenza dell'impulso di sincronismo orizzontale e SH2 è l'ultimo; similmente, SV1 è il punto di partenza dell'impulso di sincronismo verticale e SV2 è l'ultimo.

Per inserire questi, ricordate il discorso sulla magia nera e gli impulsi di sincronismo che è stato fatto precedentemente. SH1 è il punto dove inizia l'impulso di sincronismo orizzontale; quindi, SH1 = HR + HGT1. SH2 è l'estremo opposto; quindi, SH2 = SH1 + HSP. Similarmente, SV1 = VR + VGT (ma VGT di solito è zero) e SV2 = SV1 + VSP.

```
#nome     clock    temporiz.  oriz.     temporiz. vert.     flag
936x702   65       936 968 1200 1232    702 702 710 737
```

Non è necessario alcun flag; questa non è una modalità interlacciata. Ora siamo proprio a posto.

13. Sovraccaricare il Vostro Monitor

Non dovreste assolutamente superare le specifiche di scansione del monitor se questo è un tipo a frequenza fissa. Potreste far fumare il vostro hardware! Superare le specifiche di un monitor multisincrono crea dei subdoli problemi potenziali che dovreste evitare.

Al contrario, avere un pixel-clock più alto della massima larghezza di banda del monitor è abbastanza innocuo. È problematico eccedere la frequenza massima di sincronismo. Alcuni monitor moderni possono avere dei circuiti di protezione che spengono il monitor a frequenze di scansione pericolose, ma non fateci affidamento. In particolare ci sono monitor multisinc più vecchi (come il Multisync II) che usano solo un trasformatore orizzontale. Questi monitor non hanno molta protezione contro il sovraccarico. Mentre avrete sicuramente un circuito di regolazione dell'alto voltaggio (che può essere assente nei monitor a frequenza fissa), questo non necessariamente coprirà ogni gamma di frequenza possibile, specie nei modelli più economici. Questo non solo implica un maggior carico sul circuito, ma causa anche un invecchiamento precoce dei fosfori dello schermo, ed ulteriori emissioni di radiazioni dallo schermo (inclusi i raggi X).

Comunque, la problematica fondamentale in questione è lo slew rate (la pendenza dei segnali video) dei driver per l'output del video, e questo è di solito indipendente dalla frequenza dei pixel, ma (se il costruttore della vostra scheda video ci tiene a questi problemi) in relazione alla massima frequenza dei pixel della scheda.

Quindi state attenti là fuori...

14. Usare le Modalità Interlacciate

(Questa sezione è largamente dovuta a David Kastrup <`dak@pool.informatik.rwth-aachen.de`>)

Ad un dato clock fisso, uno schermo interlacciato avrà molto meno sfarfallio di uno non-interlacciato, se il circuito verticale del vostro monitor lo supporta stabilmente. È principalmente a causa di ciò che furono inventate le modalità interlacciate.

Le modalità interlacciate hanno una cattiva reputazione perché sono inferiori alle loro compagne non-interlacciate alla stessa frequenza di scansione verticale, VSF (quello che di solito viene specificato nelle pubblicità). Ma sono assolutamente superiori alla stessa frequenza di scansione orizzontale, ed è qui che di solito le caratteristiche decisive del vostro monitor/scheda video dicono le bugie.

Ad una *frequenza di refresh* fissa (o mezzo quadro, o VSF) il display interlacciato sfarfallerà di più: un display interlacciato a 90Hz sarà inferiore ad un display non-interlacciato a 90Hz. Però avrà bisogno di metà larghezza di banda video e metà frequenza di scansione orizzontale. Se lo comparate ad una modalità non-interlacciata con lo stesso dot clock e le stesse frequenze di scansione, sarà enormemente superiore: 45Hz non-interlacciati sono intollerabili. Con 90Hz interlacciati, io ho lavorato per anni con il mio Multisync 3D (a 1024x768) ed in modo molto soddisfacente. Penso che vi servirà almeno uno schermo a 70Hz non-interlacciati per lo stesso comfort.

Dovete stare attenti ad alcuni punti, comunque: usate modalità interlacciate solo ad alte risoluzioni, così che le righe accese alternate siano più vicine. Vorrete provare varie larghezze e posizioni dell'impulso di sincronismo per ottenere le posizioni di riga più stabili. Se le linee alternate sono chiare e scure, l'interlacciamento vi *salterà* agli occhi. Ho una applicazione che sceglie tali valori per uno sfondo (XCept, nessun'altra applicazione lo fa che io sappia, fortunatamente). Io passo a 800x600 usando XCept perché altrimenti mi fanno proprio male gli occhi.

Per lo stesso motivo, usate i font di minimo 100dpi, o altri font che abbiano le stanghette orizzontali spesse minimo due righe (per le alte risoluzioni, nient'altro avrebbe comunque senso).

E naturalmente, non usate mai la modalità interlacciata quando il vostro hardware ne supporterebbe una non-interlacciata con la stessa frequenza di refresh.

Se, comunque, pensate che per qualche risoluzione state spingendo o il monitor o la scheda video al massimo delle loro capacità, e ottenete sfarfallio o perdita di nitidezza (eccesso di larghezza di banda), potete provare la stessa risoluzione usando una modalità interlacciata. Chiaramente ciò non serve a nulla se il VSF del monitor è già vicino ai suoi limiti.

Scrivere modalità interlacciate è facile: fatelo come se fossero non-interlacciate. Sono necessarie solo due considerazioni in più: vi servono un numero totale pari di righe verticali (l'ultima cifra nel mode line), e quando specificate il flag "Interlace", la frequenza verticale di quadro raddoppia. Il vostro monitor deve supportare una frequenza di 90Hz se la modalità che avete specificato risulta di 45Hz senza il flag "Interlace".

Come esempio, ecco la mia modeline per 1024x768 interlacciati: il mio Multisync 3D supporta fino a 90Hz verticali e 38kHz orizzontali.

```
ModeLine "1024x768" 45 1024 1048 1208 1248 768 768 776 807 Interlace
```

Entrambi i limiti sono abbastanza sfruttati con questa modalità. Specificando la stessa modalità, ma senza il flag "Interlace", si è ancora quasi al limite della capacità orizzontale del monitor (e per dirla tutta, un po' sotto al minimo della frequenza di scansione verticale), ma così si crea uno sfarfallio intollerabile dello schermo.

Regole basilari di programmazione: se avete scritto una modalità a meno della metà della capacità verticale del monitor, rendete pari il numero totale delle linee ed aggiungete il flag "Interlace". La qualità dell'immagine dovrebbe aumentare di molto nella maggioranza dei casi.

Se avete una modalità non-interlacciata che in qualche modo sfrutta al limite le specifiche del monitor dove la frequenza di scansione verticale giace a circa il 30% o più sotto al massimo del monitor, scrivere a mano una modalità interlacciata (probabilmente con una più alta risoluzione) potrebbe portare a risultati superiori, ma non ci giurerei.

15. Domande e Risposte

D: L'esempio che avete fatto è per un formato non standard dello schermo, lo posso usare?

R: Perché no? NON ci sono ragioni di usare 640x480, 800x600, o anche 1024x768. I server XFree86 vi lasciano configurare il vostro hardware molto liberamente. Di solito ci vogliono due o tre prove prima di arrivare alla scelta giusta. La cosa importante da raggiungere è un alta frequenza di refresh con un'area visibile ragionevole. Non alta risoluzione al prezzo di sfarfallio scassa-occhi!

D: È la sola risoluzione possibile quella data di 65Mhz dot clock e 55Khz HSF?

R: Assolutamente no! Siete incoraggiati a seguire la procedura generica e fare un po' di prova-e-sbaglia fino ad arrivare ad una configurazione che vi soddisfi. Sperimentare così può essere molto divertente. Molte configurazioni vi possono dare solo schermi sgangherati, ma in pratica i moderni monitor multi-sync non sono facilmente danneggiabili. Assicuratevi comunque che il vostro monitor possa supportare le frequenze di quadro che avete scelto prima di usarlo per periodi più lunghi.

State attenti ai monitor a frequenza fissa! Questo genere di sperimentazioni li può rapidamente danneggiare. Assicuratevi di usare frequenze di refresh valide per *qualsiasi* esperimento su di essi.

D: Tu hai appena nominato due risoluzioni standard. In Xconfig, sono disponibili molte risoluzioni standard, mi puoi dire a che serve sperimentare con le temporizzazioni?

R: Certamente! Prendi, ad esempio, lo "standard" 640x480 presente nell'attuale Xconfig. Usa 25Mhz di frequenza pilota, lunghezze di quadro di 800 e 525 => frequenza di refresh ~ 59.5Hz. Non male. Ma 28Mhz è una frequenza pilota comune a molte schede SVGA. Se noi la usiamo per pilotare 640x480, seguendo le procedure prima discusse, otterremmo lunghezze di quadro tipo 812 e 505. Ora la frequenza di refresh è aumentata a 68Hz, un miglioramento abbastanza significativo rispetto allo standard.

D: Puoi riassumere quello che si è detto fino ad ora?

R: In breve:

- per ogni frequenza pilota fissa, aumentare la risoluzione massima ci penalizza nell'abbassare la frequenza di refresh e producendo così più sfarfallio.
- se serve un'alta risoluzione e il tuo monitor la supporta, cerca di procurarti una scheda SVGA che abbia un uguale dot clock o DCF. Più sono elevati, meglio è!

16. Risolvere Problemi con l'Immagine.

OK, così avete i vostri valori di configurazione X. Li mettete in Xconfig con una etichetta di modalità test. Fate partire X, con la combinazione di tasti andate nella nuova modalità, ... e l'immagine non va bene. Che fate? Ecco una lista di comuni distorsioni dell'immagine e come risolverle.

(Eliminare questi difetti minori è la specialità di xvidtune(1).)

Spostate l'immagine cambiando la temporizzazione dell'impulso di sincronismo. La *scalate* cambiando la lunghezza del quadro (dovete cambiare l'impulso di sincronismo per mantenere la stessa posizione relativa, altrimenti cambiarne la grandezza sposterà l'immagine). Ecco alcune ricette più specifiche:

Le posizioni orizzontali e verticali sono indipendenti. Cioè, spostare l'immagine orizzontalmente non incide sulla sua posizione verticale, o viceversa. Però lo stesso non vale sempre per l'ingrandimento o rimpicciolimento. Mentre cambiare la larghezza non ha nulla a che vedere con l'altezza o viceversa, cambiare entrambe può avere dei limiti. In particolare, se l'immagine è troppo larga in entrambe le dimensioni voi probabilmente dovete andare a più alti dot clock per aggiustare le cose. Dal momento che questo alza la risoluzione, è spesso un problema!

16.1. L'immagine è troppo a destra o troppo a sinistra

Per aggiustare questo, cambiate l'impulso di sincronismo orizzontale. Ovvero, aumentate o diminuite (a multipli di 8) i due numeri in mezzo nella sezione della temporizzazione orizzontale che definiscono i limiti di partenza e fine dell'impulso di sincronismo orizzontale.

Se l'immagine è spostata a sinistra (bordo destro troppo largo, voi avete bisogno di muovere l'immagine a destra) diminuite i valori. Se l'immagine è spostata a destra (bordo sinistro troppo largo, avete bisogno di spostare l'immagine a sinistra) incrementate l'impulso di sincronismo.

16.2. L'immagine è troppo su o troppo giù

Per aggiustare questo, cambiate l'impulso di sincronismo verticale. Cioè, incrementate o diminuite i due valori mediani della sezione di temporizzazione verticale che definiscono i limiti di partenza e fine dell'impulso di sincronismo verticale.

Se l'immagine è spostata su (bordo in basso troppo largo, avete bisogno di spostare l'immagine in giù) diminuite i valori. Se l'immagine è spostata in giù (bordo in alto troppo largo, avete bisogno di spostare l'immagine in su) incrementate i valori.

16.3. L'immagine è troppo larga sia orizzontalmente che verticalmente

Passate ad una più alta velocità di clock della scheda. Se avete più modalità nel file di clock, forse è stata attivata per errore una modalità a velocità inferiore.

16.4. L'immagine è troppo larga (troppo stretta) orizzontalmente

Per rimediare a questo, aumentate (diminuite) la lunghezza di quadro orizzontale. Ovvero, cambiate il quarto valore nella prima sezione di temporizzazione. Per evitare di muovere l'immagine, cambiate anche l'impulso di sincronismo (secondo e terzo numero) della metà, per mantenerla nella stessa posizione relativa.

16.5. L’immagine è troppo alta (troppo bassa) verticalmente

Per rimediare, aumentate (diminuite) la lunghezza del quadro. Cioè cambiate il quarto numero nella seconda sezione di temporizzazione. Per evitare di spostare l’immagine, cambiate anche l’impulso di sincronismo (il secondo e terzo numero) della metà, per mantenerla nella stessa posizione relativa.

Qualsiasi altra distorsione che non può essere risolta dalla combinazione di queste tecniche è probabilmente sintomo di qualcosa di profondamente sbagliato, tipo un errore di calcolo o un dot clock più veloce di quello che può reggere il monitor

In ultimo, ricordate che aumentare la lunghezza del quadro diminuirà la frequenza di refresh, e viceversa.

Occasionalmente potete aggiustare piccole distorsioni con i controlli del vostro monitor. Lo svantaggio è che se portate i controlli troppo lontano dallo zero (il settaggio di fabbrica) per risolvere i problemi di grafica, potreste ottenere un’immagine distorta in modo testo. È meglio avere una giusta "modeline".

17. Tracciare le Capacità del Monitor

Per tracciare un diagramma della modalità del monitor, vi serve il pacchetto gnuplot (un linguaggio di disegno freeware per sistemi operativi UNIX-compatibili) e il tool modeplot, uno script di shell/gnuplot per tracciare il diagramma delle caratteristiche del vostro monitor, digitate come opzioni a linea di comando.

Ecco una copia di modeplot:

```
#!/bin/sh
#
# modeplot -- genera un grafico X mode delle modalità disponibili con il
#             proprio monitor
#
# Digitare ‘modeplot -?’ per vedere le opzioni di controllo.
#

# Descrizione del monitor. Larghezza di banda in MHz, frequenze orizzontali
# in kHz e frequenze verticali in Hz.
TITLE="Viewsonic 21PS"
BANDWIDTH=185
MINHSF=31
MAXHSF=85
MINVSF=50
MAXVSF=160
ASPECT="4/3"
vesa=72.5 # minima frequenza di refresh raccomandata da VESA

while [ "$1" != "" ]
do
 case $1 in
 -t) TITLE="$2"; shift;;
 -b) BANDWIDTH="$2"; shift;;
 -h) MINHSF="$2" MAXHSF="$3"; shift; shift;;
 -v) MINVSF="$2" MAXVSF="$3"; shift; shift;;
 -a) ASPECT="$2"; shift;;
```

```
 -g) GNUOPTS="$2"; shift;;
 -?) cat <<EOF
modeplot control switches:

-t "<description>" nome del monitor           default: "Viewsonic21PS"
-b <nn>            largh.di banda in MHz      default: 185
-h <min> <max>   min & max HSF (kHz)        default: 31 85
-v <min> <max>   min & max VSF (Hz)         default: 50 160
-a <aspect ratio>  rapporto visivo            default: 4/3
-g "<options>"     opzioni passate a gnuplot

Le opzioni -b, -h e -v sono obbligatorie, -a, -t, -g opzionali. Potete usare
-g per passare un tipo di device a gnuplot così che (per esempio) l'output di
modeplot può essere rediretto ad una stampante. Vedi gnuplot(1) per
dettagli.

Il tool modeplot è stato creato da Eric S. Raymond <esr@thyrsus.com>
basato su analisi e codice di Martin Lottermoser Martin Lottermoser
<Martin.Lottermoser@mch.sni.de>

This is modeplot $Revision: 1.25 $
EOF
  exit;;
 esac
 shift
done

gnuplot $GNUOPTS <<EOF
set title "$TITLE Mode Plot"

# Numeri magici. Sfortunatamente, il grafico è abbastanza sensibile a
# modifiche, e potrebbe fallire nel rappresentare la realtà di alcuni
# monitor. Dobbiamo correggere questi valori per ottenere una
# approssimazione del diagramma dei modi. Questi sono ricavati analizzando
# molti valori nel database ModeDB.
F1 = 1.30 # moltiplicatore per convertire la risoluzione orizzontale
                # in larghezza di quadro
F2 = 1.05 # moltiplicatore per convertire la risoluzione verticale
                # in altezza di quadro

# Definizione delle funzioni (moltiplicando per 1.0 si forza
# l'aritmetica a numeri reali)
ac = (1.0*$ASPECT)*F1/F2
refresh(hsync, dcf) = ac * (hsync**2)/(1.0*dcf)
dotclock(hsync, rr) = ac * (hsync**2)/(1.0*rr)
resolution(hv, dcf) = dcf * (10**6)/(hv * F1 * F2)

# Mette delle etichette sugli assi
set xlabel 'DCF (MHz)'
set ylabel 'RR (Hz)' 6 # Put it right over the Y axis

# Genera il diagramma
set grid
```

```
set label "VB" at $BANDWIDTH+1, ($MAXVSF + $MINVSF) / 2 left
set arrow from $BANDWIDTH, $MINVSF to $BANDWIDTH, $MAXVSF nohead
set label "max VSF" at 1, $MAXVSF-1.5
set arrow from 0, $MAXVSF to $BANDWIDTH, $MAXVSF nohead
set label "min VSF" at 1, $MINVSF-1.5
set arrow from 0, $MINVSF to $BANDWIDTH, $MINVSF nohead
set label "min HSF" at dotclock($MINHSF, $MAXVSF+17), $MAXVSF + 17 right
set label "max HSF" at dotclock($MAXHSF, $MAXVSF+17), $MAXVSF + 17 right
set label "VESA $vesa" at 1, $vesa-1.5
set arrow from 0, $vesa to $BANDWIDTH, $vesa nohead # style -1
plot [dcf=0:1.1*$BANDWIDTH] [$MINVSF-10:$MAXVSF+20] \
  refresh($MINHSF, dcf) notitle with lines 1, \
  refresh($MAXHSF, dcf) notitle with lines 1, \
  resolution(640*480,   dcf) title "640x480  " with points 2, \
  resolution(800*600,   dcf) title "800x600  " with points 3, \
  resolution(1024*768,  dcf) title "1024x768 " with points 4, \
  resolution(1280*1024, dcf) title "1280x1024" with points 5, \
  resolution(1600*1280, dcf) title "1600x1200" with points 6

pause 9999
EOF
```

Una volta che avete a posto modeplot e il package gnuplot, vi servono le seguenti caratteristiche del monitor:

- larghezza di banda video (VB)
- gamma delle frequenze di sincronismo orizzontale (HSF)
- gamma delle frequenze di sincronismo verticale (VSF)

Il programma plot deve fare delle semplificazioni che non sono necessariamente corrette. Questo è il motivo per cui il diagramma che ne risulta ne è solo una rozza descrizione. Queste semplificazioni sono:

- Tutte le risoluzioni hanno una singola frequenza fissa di aspetto AR = HR/VR. Risoluzioni standard hanno AR = 4/3 o AR = 5/4. I programmi modeplot assumono 4/3 di default, ma si può non tenerne conto.
- Per le modalità considerate, le larghezze di quadro orizzontali e verticali sono multipli fissi di risoluzioni orizzontali e verticali, rispettivamente:

  ```
  HFL = F1 * HR
  VFL = F2 * VR
  ```

Come linea guida, prendete F1 = 1.30 and F2 = 1.05 (vedi Calcolare le grandezze di quadro).

Ora prendete una particolare frequenza di sincronismo, HSF. Dato l'assunto prima presentato, ogni valore di clock DCF già determina la frequenza di refresh RR, per esempio per ogni valore di HSF c'è una funzione RR (DCF). Questo può essere derivato come segue.

La frequenza di refresh è uguale al clock diviso per il prodotto della grandezza del quadro:

```
RR = DCF / (HFL * VFL)  (*)
```

D'altra parte, la larghezza di quadro è uguale al clock diviso per la frequenza di sincronismo orizzontale:

```
HFL = DCF / HSF   (**)
```

VFL può essere ridotto a HFL secondo i due assunti precedenti:

```
VFL = F2 * VR
     = F2 * (HR / AR)
     = (F2/F1) * HFL / AR (***)
```

Inserendo (**) e (***) in (*) otteniamo:

```
RR = DCF / ((F2/F1) * HFL**2 / AR)
   = (F1/F2) * AR * DCF * (HSF/DCF)**2
   = (F1/F2) * AR * HSF**2 / DCF
```

Per un HSF fisso, F1, F2 e AR, questa è un iperbole nel nostro diagramma. Disegnando due tali curve per le frequenze di sincronismo orizzontale minime e massime noi otteniamo i bordi rimanenti dell'area permessa.

Le righe dritte che attraversano l'area possibile rappresentano risoluzioni particolari. Questo è basato su (*) e il secondo assunto:

```
RR = DCF / (HFL * VFL) = DCF / (F1 * HR * F2 * VR)
```

Disegnando tali linee per tutte le risoluzioni che ci interessano, si possono immediatamente leggere le possibili relazioni fra risoluzione, frequenza di clock e frequenza di refresh delle quali il monitor è capace. Notate che queste linee non dipendono dalle proprietà del monitor, ma dipendono dal secondo assunto.

Il tool modeplot vi offre un modo facile per farlo. Digitate modeplot -? per vedere le sue opzioni di controllo. Una chiamata tipica è questa:

```
modeplot -t "Swan SW617" -b 85 -v 50 90 -h 31 58
```

L'opzione -b specifica la larghezza di banda video; -v e -h settano le gamme di frequenza di sincronismo orizzontale e verticale.

Quando leggete l'output di modeplot, tenete sempre a mente che vi dà solo una descrizione approssimativa. Per esempio, non tiene conto dei limiti di un HFL risultante da una minima ampiezza richiesta dell'impulso di sincronismo, e può essere solo accurato quanto lo sono gli assunti. Non c'è niente di meglio quindi di un calcolo dettagliato (compresa un po' di magia nera) come viene presentato in Mettere Tutto Assieme. Comunque, vi dovreste sentire più a vostro agio per quanto possibile e per quanti compromessi si debbano fare.

18. Ringraziamenti

Il padre originale di questo documento è stato Chin Fang <`fangchin@leland.stanford.edu`>.

Eric S. Raymond <`esr@snark.thyrsus.com`>; ha rielaborato, riorganizzato, e massicciamente riscritto l'originale di Chin Fang col proposito di capirlo. In questo processo, ha immesso la maggior parte di un altro how-to di Bob Crosson <`crosson@cam.nist.gov`>.

Il materiale sulle modalità interlacciate è largamente di David Kastrup <`dak@pool.informatik.rwth-aachen.de`>.

Nicholas Bodley <`nbodley@alumni.princeton.edu`> ha corretto e chiarito la sezione su come funziona lo schermo.

Payne Freret <`payne@freret.org`> ha corretto alcuni minori errori tecnici sulla progettazione dei monitor.

Martin Lottermoser <`Martin.Lottermoser@mch.sni.de`> ha contribuito all'idea di usare gnuplot per fare diagrammi di modalità e ha fatto l'analisi matematica che c'è dietro modeplot. Il modeplot in distribuzione fu ridisegnato e generalizzato da ESR dal codice originale di Martin per un caso.