Vidéo HOWTO

Jean-Michel VANSTEENE, <vansteen@frcl.bull.fr>, <vanstee@worldnet.net> v1.1, 13 Octobre 1995

Ce document souhaite apporter une aide à la configuration de votre environnement graphique sous Linux, plus spécifiquement concernant votre sous-système vidéo. La première partie sera entièrement consacrée aux serveurs XFree86 et au fichier de configuration correspondant. La deuxième décrira quelques programmes qui mettent en oeuvre les méthodes étudiées. Toutes vos remarques concernant la forme ou le fond sont les bienvenues. Elles permettront de faire de ce document une aide précieuse pour tirer le meilleur de votre matériel. Vous pouvez les envoyer aux adresses ci-dessus. Bonne lecture.

Table des matières

1 Introduction **2**

1.1 Généralités sur X . . . 2

1.1.1 Affichages et écrans . . . 2

1.1.2 Le modèle client-serveur . . . 3

1.2 Et Linux dans tout cela . . . 3

2 Vue d'ensemble de la méthode de configuration **3**

2.1 Choisir le serveur X adéquat . . . 4

3 La partie facile du fichier XF86Config **5**

3.1 Les documents à consulter . . . 6

4 La partie difficile du fichier XF86Config **6**

4.1 La section Monitor . . . 6

4.1.1 Comment fonctionne votre écran? . . . 6

4.1.2 Les choses à connaître concernant votre carte vidéo et votre moniteur . . . 7

4.1.3 Interpréter les spécifications de base . . . 10

4.1.4 Compromis dans la configuration de votre système . . . 11

4.1.5 Taille mémoire nécessaire . . . 12

4.1.6 Calculer les tailles de trames . . . 12

4.1.7 Magie noire et tops de synchro . . . 13

4.1.8 Rassemblons nos idées . . . 16

4.1.9 Foire Aux Questions . . . 18
4.1.10 Deux exemples supplémentaires de calculs . . . 19
4.1.11 Corriger des problèmes d'image . . . 23
4.1.12 Quelques exemples réels de Modes vidéo . . . 24
4.2 La section Device . . . 28

5 Les applications d'aide à la configuration 29

5.1 Vgaset . . . 29
5.1.1 Les options de vgaset . . . 29
5.1.2 Les commandes de vgaset . . . 30
5.1.3 Les bugs de vgaset . . . 31
5.2 xclk . . . 31
5.3 SuperProbe . . . 32
5.3.1 Description . . . 32
5.3.2 Les options . . . 33
5.3.3 Les bugs . . . 34

6 Informations concernant ce document 34

1 Introduction

L'environnement X permet aux systèmes fonctionnant sous Unix d'avoir un environnement graphique de qualité. C'est un système de gestion de fenêtres basé sur un fonctionnement en réseau. Bien entendu il est utilisable sur une machine isolée, mais cela signifie qu'il prend naturellement en compte votre réseau si vous en avez un. Les programmes peuvent tourner sur des machines distantes, la visualisation se faisant sur la station de travail locale.

Sous Linux, l'environnement X est le système X-Window du MIT.

1.1 Généralités sur X

L'environnement X est relativement complexe mais nous en dégagerons les grands points. Il existe une littérature importante sur ce sujet.

1.1.1 Affichages et écrans

L'environnement X est un système de multi-fenêtrage graphique en mode point (*bitmap*). Il supporte la couleur, les niveaux de gris et le monochrome. Un système d'affichage sous X (appelé *display*) correspond à un ensemble comprenant **un clavier**, **un système de pointage** et **un ou plusieurs écrans**. Ceci est fondamental et lors de la configuration de cet environnement, notamment sous Linux, c'est cet ensemble complet qu'il faudra configurer. Nous n'étudierons en détail que la configuration du sous-système vidéo. Pour les deux autres sous-systèmes il existe des documents fort bien réalisés dont une liste est fournie au chapitre 3.1.

1.1.2 Le modèle client-serveur

X est donc un système de gestion de fenêtres orienté réseaux. La visualisation est complètement indépendante de l'exécution des applications. Alors que plusieurs applications tournent sur votre machine, plusieurs autres pourront tourner sur des machines distantes, envoyer des requêtes à travers le réseau vers votre affichage (*display*) et recevoir des événements provenant du clavier ou de la souris. Bien entendu la pile TCP/IP est utilisée.

Le programme qui contrôle chaque affichage est un *serveur*. Votre station est serveur dès que votre affichage est accessible (éventuellement via le réseau) aux clients que sont les applications.

1.2 Et Linux dans tout cela

XFree86 est le nom générique du serveur X sous Linux. Nous traiterons ici de XFree versions 3 et suivantes (X11R6).

Sa configuration est un point important et parfois difficile. Il est fortement conseillé d'y apporter un soin particulier.

Une bonne configuration permettra d'obtenir le meilleur de votre sous-système vidéo, mais une très mauvaise configuration pourra l'endommager.

Si vous suivez les conseils donnés dans la suite, vous devriez réussir assez rapidement à obtenir un système pleinement opérationnel. Bien entendu, devant la diversité du matériel existant, nous ne pourrons indiquer l'ensemble des configurations possibles, mais seulement un méthode. Il existe beaucoup de documents plus spécifiques à chaque matériel (*/usr/lib/X11/doc/README.**). A utiliser avec la connaissance nécessaire !

2 Vue d'ensemble de la méthode de configuration

La méthode décrite dans ce document s'applique à toutes les versions de XFree86. Elle s'appuiera néanmoins, pour les exemples, sur la distribution *XFree86-3.1.1*.

Il y a deux étapes majeures pour installer le système graphique et le rendre opérationnel. La première consiste à sélectionner le serveur approprié. La seconde étape consiste à le configurer, c'est-à-dire rédiger un fichier **XF86Config** correct. Ce fichier décrira les trois principaux éléments d'un système d'affichage (*display*) : le **clavier**, la **souris** et l'**écran**.

Le fichier XF86Config contient plusieurs sections ; les paragraphes suivants vous aideront à remplir chacune d'elles. Vous trouverez en général dans toute distribution de Linux un fichier d'exemple tel que */usr/lib/X11/XF86Config.eg* ou similaire.

Pour ne pas partir de rien, vous pouvez copier ce fichier dans */usr/lib/X11* et le nommer *XF86Config*. C'est l'endroit ou le serveur (quel qu'il soit) ira le chercher. Plusieurs pages de manuel en ligne sont livrées avec XFree86 concernant XF86Config(4/5). Il est vivement conseillé de les parcourir (fichier *XFree86-3.1.1/X311doc.tgz*).

Les sections du fichier XF86Config sont les suivantes :

Files

Décrit les chemins d'accès aux fontes et fichiers RGB (Rouge Vert Bleu)

Server Flags

Décrit les options générales du serveur. Reportez-vous au manuel en ligne pour connaître les options possibles.

Keyboard

Configure le clavier, et quelques paramètres optionnels.

Pointer

Décrit le périphérique de pointage, et quelques paramètres optionnels

Monitor

Décrit votre moniteur.

Graphics Device

Décrit votre carte vidéo.

Screen

Décrit comment le moniteur et la carte vidéo doivent être utilisés.

2.1 Choisir le serveur X adéquat

Plusieurs serveurs peuvent se trouver sur votre machine en tant que fichiers exécutables. Un seul doit être défini comme le serveur X de votre système. Vous devez donc choisir le bon.

Les serveurs sont généralement localisés dans */usr/X11R6/bin*. Le nom « officiel » du serveur qui sera exécuté est « X ». Il est généralement commode de créer un lien symbolique de X vers le serveur que vous aurez choisi.

Les différents serveurs sont :

XF86_Mono

(S)VGA monochrome, pour les matériels monochromes, par exemple Hercules.

XF86_VGA16

Serveur couleur VGA 16 couleurs.

XF86_SVGA

Serveur Super VGA. Il peut être utilisé pour tout matériel SVGA. Il supporte les caractéristiques accélératrices des cartes Cirrus 542{0,2,4,6,8,9}, Western Digital 90C3{1,3}, Oak Technologies Inc. OTIO87. Les autres cartes ne sont pas accélérées.

XF86_S3

Serveur accélérateur pour cartes à base de circuit S3.

XF86_Mach8

Serveur accélérateur pour cartes ATI Mach8.

XF86_Mach32

Serveur accélérateur pour cartes ATI Mach32.

XF86_Mach64

Serveur accélérateur pour cartes ATI Mach64.

XF86_8514

Serveur accélérateur pour cartes 8514/A.

XF86_P9000

Serveur accélérateur pour cartes P9000.

XF86_AGX

Serveur accélérateur pour cartes AGX

XF86_W32

Serveur accélérateur pour cartes ET4000/W32.

Il existe un manuel en ligne pour chacun de ces serveurs. Il est bon d'en prendre connaissance, notamment pour les particularités liées à celui qui vous intéresse.

Il est à noter qu'il est possible de modifier les pilotes configurés dans un serveur via le *LinkKit* ; Il existe un fichier README à ce propos. Le binaire du serveur peut ne pas contenir tous les pilotes possibles, selon la distribution que vous utilisez. Lancez `/usr/X11R6/bin/X -showconfig` pour visualiser les pilotes configurés. Si vous avez besoin de modifier votre serveur, référez vous au fichier README.

3 La partie facile du fichier XF86Config

Comme le titre l'indique, cette partie ne pose que très rarement des problèmes. Pour ne pas dépasser le cadre que nous nous sommes fixés et ne pas faire d'ombre aux documents existants, nous nous contenterons de vous aiguiller vers quelques-uns d'entre eux qui devraient vous permettre de vous en sortir.

3.1 Les documents à consulter

Vous pourrez utilement consulter :

- Les fichiers README du répertoire */usr/lib/X11/doc*.
- Les page de manuel en ligne `XF86Config(4/5)`.
- Le XFree86-HOWTO en français ou anglais.
- Le chapitre concernant *X Window* du meeeer-veil-leux ;-) livre « Le système Linux » de **Matt Welsh** et **Lar Kaufman** et traduit en français par le meeeer-veil-leux René Cougnenc.
- Les groupes *Usenet* **comp.os.linux.x** et **fr.comp.os.linux**. Avant de poser des questions, prenez bien soin de vérifier que la réponse ne se trouve pas dans un de ces documents ou dans ce qui suit. Rappelons que le site **ftp.ibp.fr** tient à jour la liste et le contenu de tous les messages émis dans **fr.comp.os.linux**. Ils se trouvent dans */pub/linux/french/fr.comp.os.linux*

4 La partie difficile du fichier XF86Config

4.1 La section Monitor

C'est de loin la plus difficile à configurer. Cette partie va être très détaillée afin que vous puissiez comprendre comment décrire tous les timings correspondant à votre matériel. Il est important de faire très attention lors de la configuration de cette section. Certaines erreurs peuvent être fatales pour le moniteur ou la carte vidéo. Même si la méthode que nous allons voir devrait permettre d'aboutir à une configuration correcte du premier coup, nous ne saurions (l'auteur et tous ceux qui lui ont permis d'écrire ce document) être tenus pour responsables de pannes résultant d'une mauvaise configuration (Rassurez-vous, ce n'est quand même pas courant ;-)).

4.1.1 Comment fonctionne votre écran?

Cette connaissance est essentielle pour comprendre les calculs que vous allez devoir effectuer. Les valeurs obtenues seront utilisées par le serveur pour contrôler l'affichage au plus bas niveau.

L'écran génère une image à partir d'une succession de points. Les points sont arrangés de la gauche vers la droite pour former une ligne. Les lignes sont arrangées du haut vers le bas pour former une image. Un point est lumineux lorsque le faisceau électronique est actif et qu'il rencontre la couche phosphorescente du tube cathodique. Le faisceau balaie l'écran de façon régulière.

Les paragraphes suivants sont repris d'un original de Chin Fang <fangchin@leland.stanford.edu> et de portions d'un howto de Bob Crosson <crosson@cam.nist.gov>

Le faisceau électronique démarre sa course en haut et à gauche de l'écran. Il parcourt l'écran en lignes droites successives de gauche à droite. A l'extrême droite d'une ligne, il s'arrête momentanément pour être dévié rapidement vers la gauche et d'une ligne vers le bas, et ainsi de suite. Ce schéma est répété jusqu'à la dernière ligne. Le faisceau est alors déplacé du coin inférieur droit vers le coin supérieur gauche de l'écran, et un nouveau balayage peut être commencé.

Le démarrage du faisceau en haut et à gauche est appelé le début d'une trame. La trame se termine lorsque le faisceau atteint à nouveau l'angle supérieur gauche, provenant directement de l'angle inférieur droit.

Une trame est faite de toutes les lignes que le faisceau a tracées. Si le faisceau électronique était allumé durant tout le parcours, tous les points seraient illuminés. Il n'y aurait pas de bordures noires de chaque côté de l'écran. Aux coins, l'image serait déformée en raison de la diffculté à contrôler le faisceau en ces endroits. Pour réduire la distorsion, les points aux angles et sur les bords ne sont pas illuminés, même si le faisceau peut passer dessus. La zone visible de l'écran est donc réduite.

Il est également important de comprendre ce que devient le faisceau lorsqu'il n'est pas en train d'illuminer une partie visible de l'écran. Le temps que le faisceau utiliserait à illuminer les bords gauche et droit hors de la zone visible, sert au retour de balayage de ligne, consistant à le faire passer rapidement au début de la ligne suivante. Le temps que le faisceau utiliserait pour illuminer les bords hauts et bas hors de la zone visible, sert au retour de balayage de trame, faisant passer le faisceau du coin inférieur droit au coin supérieur gauche.

La carte vidéo génère les signaux permettant au moniteur de contrôler le faisceau électronique afin de rendre une image correcte dans la zone visible. La carte contrôle aussi le retour de balayage de ligne en générant un signal appelé synchronisation horizontale (**hsync**) et le retour de balayage de trame en générant un signal de synchronisation verticale (**vsync**). Un signal *hsync* est émis à chaque fin de ligne. Un signal *vsync* est émis à la fin de chaque trame.

La technique des signaux vidéo nécessite qu'un temps court mais non nul soit inséré avant et après les signaux de synchronisation horizontale et verticale de façon que la position du faisceau électronique puisse se stabiliser. Sans cela l'image ne serait pas très stable.

4.1.2 Les choses à connaître concernant votre carte vidéo et votre moniteur

Avant de bidouiller votre fichier de configuration, plusieurs choses sont à connaître :

1. Les plages de fréquences de synchronisation horizontale et verticale de votre moniteur. Ces valeurs seront indiquées dans les parties **HorizSync** et **VertRefresh**.
2. La plage de fréquences d'horloge de votre carte, ou « dot clock » (horloge point ou pixel).
3. La largeur de bande passante de votre moniteur. Cette valeur alimentera la partie **Bandwidth**.

Les fréquences de synchronisation de votre moniteur :

La fréquence de synchronisation horizontale correspond au nombre de lignes horizontales que le moniteur peut écrire en une seconde. C'est la valeur la plus importante concernant votre moniteur. La fréquence de synchronisation verticale est le nombre de balayages verticaux que le moniteur peut effectuer en une seconde.

Les fréquences de synchronisation sont généralement données dans les pages des spécifications techniques de votre moniteur. La fréquence de synchro verticale est un nombre exprimé en Hz (cycles par secondes), la fréquence horizontale étant donnée en kHz (milliers de cycles par seconde). Les valeurs courantes sont de l'ordre de 50 à 80 Hz pour les premières et 31 à 135 kHz pour les secondes.

Si vous possédez un moniteur multi-synchro, ces fréquences seront données sous forme d'intervalle(s). Certains moniteurs possèdent plusieurs fréquences fixes. Elles peuvent être utilisées, mais vos options seront limitées en raisons des caractéristiques intrinsèques de votre moniteur. Choisissez le couple de valeurs les plus élevées possibles pour une meilleure résolution. Attention, piloter un moniteur à fréquences fixes à des fréquences trop élevées peut l'endommager.

Les fréquences d'horloge de votre carte :

Votre manuel de spécifications de votre carte vidéo décrit généralement la ou les fréquences d'horloge auxquelles elle peut travailler, autrement dit, le nombre total de points qu'elle peut écrire par seconde. Si vous n'avez pas cette information, le serveur X les obtiendra pour vous. Même si ce dernier ne fonctionne pas bien, il inscrira sur la sortie standard d'une console une ou plusieurs lignes de valeurs d'horloges ainsi que d'autres informations. Si vous redirigez le tout dans un fichier, ce sera sauvegardé, même si vous avez à relancer votre système pour retrouver votre affichage.

Voici un exemple des valeurs obtenues pour un adaptateur S3 (le serveur SGCS est utilisé) :

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

WGA :

type de serveur

86C911 :

processeur vidéo

1024k :

taille de la mémoire vidéo

Les valeurs indiquées sont des fréquences en Mhz.

Les fréquences dont nous disposons maintenant serviront dans la section **Device** et dans la section **Monitor**. Nous nous intéresserons à cette dernière dans ce chapitre. Le suivant décrira rapidement la configuration de ces valeurs dans la section Device.

La bande passante de votre moniteur :

Enfin, il est important et utile de connaître la bande passante vidéo de votre moniteur. Vous déduirez ainsi approximativement la plus grande valeur possible pour l'*horloge point*. Il y a pas mal de données possibles, sachant que certains moniteurs sont capables de fonctionner 30 % au dessus de leur bande passante nominale !

La connaissance de la bande passante va vous permettre d'effectuer des choix plus intelligents parmi plusieurs configurations possibles. Cela peut affecter la qualité visuelle de l'affichage (finesse des détails, scintillement, ...).

La bande passante de votre moniteur doit être indiquée dans le manuel qui l'accompagne. Si ce n'est pas le cas, regardez la plus grande résolution qu'il supporte. Voici des correspondances très courantes entre résolution et bande passante (et donc les valeur maximales pour l'*horloge point* utilisable) :

```
640 x 480               25
800 x 600               36
1024 x 768              65
1024 x 768 entrelace    45
1280 x 1024             110
```

De toute façon, il n'y a rien de magique dans tout cela ; ces valeurs sont les plus petites fréquences d'horloge pour chaque résolution dans la base de données des modes vidéo standard de XFree86. La bande passante de votre moniteur peut être plus élevée que le minimum nécessaire pour sa résolution maximale, ne soyez dont pas affolé de pouvoir utiliser une valeur légèrement supérieure.

Notez également que la bande passante est rarement un problème pour les fréquences d'horloge en dessous de 65 Mhz. Avec un moniteur SVGA ou haute résolution, vous ne frôlerez pas la limite de la bande passante de votre moniteur. Voici quelques exemples :

```
Marque                  Bande passante video
------                  --------------------


NEC 4D                          75 Mhz
Nano 907a                       50 Mhz
Nano 9080i                      60 Mhz
Mitsubishi HL6615               110 Mhz
Mitsubishi Diamond Scan         100 Mhz
IDEK MF-5117                    65 Mhz
IOCOMM Thinksync-17 CM-7126     136 Mhz
HP D1188A                       100 Mhz
Philips SC-17AS                 110 Mhz
Swan SW617                      85 Mhz
```

Même les moniteurs bas de gamme ne sont généralement pas terriblement limités par la largeur de bande pour leur résolution. L'écran NEC Multisync II en est un bon exemple. D'après ses spécifications, il ne peut afficher 800 x 600 points, mais seulement 800 x 560. Pour de telles résolutions, vous n'avez pas besoin d'horloge très rapide ni d'une bande passante très large ; il est probable que le meilleur que vous puissiez obtenir corresponde à des fréquences de l'ordre de 32 à 36 Mhz, les deux n'étant pas très loin de la bande passante du moniteur qui est de 30 Mhz.

A ces fréquences, votre image ne sera peut être pas aussi fine qu'il le faudrait mais certainement d'une qualité tolérable. Bien sûr, il serait préférable d'avoir une bande passante supérieure, disons au moins 36 Mhz. Ce n'est toutefois pas critique pour la plupart des applications comme l'édition de texte, du moment qu'il n'y a pas de distorsion sévère dans l'image (vos yeux vous le diraient).

Les contrôles :

La gamme de fréquences de synchronisation de votre moniteur, associée à la fréquence de l'horloge de votre adaptateur vidéo, détermine la résolution maximale que vous pourrez obtenir. Mais il appartient au pilote de pouvoir gérer au mieux votre matériel. Un matériel de plus haut niveau sans un pilote de périphérique apte à en tirer partie est un gâchis ! D'un autre côté, avec un très bon pilote et un matériel peu performant, vous pourrez pousser votre matériel à son maximum. C'est la philosophie de XFree86.

4.1.3 Interpréter les spécifications de base

Cette section explique en détail les spécifications dont nous venons de parler, ainsi que d'autres choses dont nous aurons besoin. D'abord, quelques définitions. Entre parenthèses sont indiqués les noms symboliques que nous utiliserons pour effectuer les calculs.

Fréquence de synchronisation horizontale (HSF)

Nombre de parcours horizontaux par seconde (voir ci-dessus)

Fréquence de synchronisation verticale (VSF)

Nombre de parcours verticaux par seconde (voir ci-dessus). Importante car elle correspond au taux de rafraîchissement maximal.

Fréquence d'horloge point (DCF)

Plus formellement « fréquence d'horloge de pilotage » ; quelquefois appelée à tort bande passante. La fréquence de l'oscillateur (VCO) de votre adaptateur, correspondant au nombre maximum de points par seconde qu'il puisse émettre.

Largeur de bande passante vidéo (VB)

La plus haute fréquence que le signal de votre moniteur puisse atteindre. Cela limite l'horloge point que vous pouvez utiliser et donc la finesse de votre image.

Longueurs de trame (HFL, VFL)

La longueur de trame horizontale (HFL) correspond au nombre de tops d'horloge nécessaire pour que le faisceau électronique de votre moniteur parcoure une ligne (incluant les bords gauche et droit inactifs). La longueur de trame verticale (VFL) correspond au nombre de lignes balayées pour une image entière (y compris les bords supérieur et inférieur inactifs).

Taux de rafraîchissement (RR)

C'est le nombre d'images par seconde. Les plus hautes fréquences sont les meilleures, puisqu'elles réduisent le scintillement. 60 Hz est correct, mais la valeur standard VESA est de 72 Hz. Le calcul donne :

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

A noter que le produit au dénominateur ne correspond pas à la résolution visible du moniteur, mais bien à quelque chose de plus grand. Nous verrons les détails après.

A propos de la largeur de bande

Les fabricants de moniteurs aiment annoncer de larges bandes passantes, parce que cela détermine directement la précision des changements de l'intensité et des couleurs de votre écran. Une plus grande bande passante signifie souvent une plus grande finesse des détails.

Votre moniteur utilise des signaux électroniques pour présenter une image à vos yeux. De tels signaux sont convertis sous forme analogique à partir d'un signal digital. Ils peuvent être considérés comme une combinaison d'ondes simples, chacune ayant une fréquence fixe, la plupart d'entre elles étant dans la gamme du mégahertz, disons de l'ordre de 20 à 70 MHz. La largeur de bande de votre moniteur correspond effectivement à la plus haute fréquence de signal analogique qu'il puisse supporter sans distorsion.

Dans notre cas, la largeur de bande est surtout importante comme point de coupure approximatif pour la plus grande valeur d'horloge point que nous pourrons utiliser.

Fréquences de synchronisation et taux de rafraîchissement

Le tracé d'une ligne horizontale à l'écran ne correspond qu'à la portion visible de la trame horizontale entière. Il n'y a à un moment donné qu'un seul point réellement actif. Le taux de rafraîchissement est suffisant pour que, grâce à la persistence rétinienne, vous ayez une image complète.

Souvenez-vous que le parcours du faisceau est un zig-zag ; il va de gauche à droite et revient rapidement à gauche en descendant.

Maintenant, nous pouvons évoquer les relations entre taux de rafraîchissement, horloge point et taille de trame. Par définition, un hertz (Hz) est une unité de fréquence correspondant à un cycle par seconde. Donc si la longueur de trame horizontale est HFL, et la longueur de trame verticale est VFL, l'écran est entièrement couvert en (HFL * VFL) tops d'horloge. Puisque votre carte émet DCF tops par seconde, manifestement le faisceau électronique de votre moniteur peut balayer l'écran de gauche à droite et de haut en bas DCF / (HFL * VFL) fois par seconde. C'est le taux de rafraîchissement (RR) de votre écran, c'est-à-dire le nombre de fois par seconde que votre image sera redessinée.

Vous devez bien assimiler ce concept pour effectuer une configuration correcte avec le meilleur compromis entre résolution et scintillement.

4.1.4 Compromis dans la configuration de votre système

La formule suivante est obtenue à partir de la précédente :

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

Votre *horloge point* est fixée par les caractéristiques de votre carte vidéo. Vous pouvez utiliser ce capital de points par seconde pour acheter soit du taux de rafraîchissement, soit de la résolution horizontale ou verticale. Si l'un augmente, un ou les deux autres doivent diminuer.

Notez cependant que le taux de rafraîchissement ne peut être supérieur à la fréquence maximum de synchro verticale de votre moniteur. Pour un moniteur donné et une horloge point donnée, il y a un minimum dans les longueurs de trames en dessous duquel vous ne pouvez pas descendre.

En choisissant vos paramètres, n'oubliez pas : si RR est trop petit, vous serez gêné par le scintillement.

Vous ne souhaiterez donc pas descendre votre taux de rafraîchissement en dessous de 60 Hz. C'est le minimum acceptable en dessous duquel l'oeil fatigue vite ; si vous êtes très sensible, choisissez 72 Hz, valeur standard VESA.

La sensibilité au scintillement est très personnelle mais dépend aussi de l'environnement de travail. Si vous travaillez sur des fonds noirs et des couleurs très contrastées, vous pourrez descendre à 45 Hz.

Faites un essai : ouvrez une fenêtre `xterm -bg white -fg black` et agrandissez-la au maximum. Réglez votre moniteur aux trois quarts de son intensité et ne regardez pas l'écran. Jetez ensuite des coups d'oeil sur votre écran, sans le regarder droit devant mais en utilisant la vision périphérique ; ce sont les cellules visuelles qui sont le plus sensibles au scintillement. Si vous estimez que la vision est confortable ou tolérable, le taux de rafraîchissement vous convient, sinon augmentez-le de façon à ne pas vous exposer à des maux de tête !

Disons qu'à ce stade vous disposez d'une valeur de taux de rafraîchissement. Avec le choix de HFL et VFL, nous avons encore pas mal de marge de manoeuvre.

4.1.5 Taille mémoire nécessaire

La quantité de mémoire vidéo dont vous disposez limite la résolution que vous pouvez atteindre pour un affichage couleur ou à niveaux de gris. Ce n'est probablement pas un critère pour les écrans noir et blanc.

Pour les affichages à 256 couleurs, un octet de mémoire vidéo permet de coder un point. Cet octet contient l'information déterminant le mélange de Rouge, Vert et Bleu nécessaire pour l'obtention de la couleur. Pour une résolution de 800 x 600 points avec un octet par point, il faut donc 800 * 600 = 480.000 octets de mémoire sur votre carte vidéo.

Plus généralement, les besoins en mémoire seront de (HR * VR) / 1024 kilo-octets de RAM (DRAM ou VRAM). Dans le cas d'une résolution de 936 x 702 points, nous aurons besoin de (936 * 702) / 1024, soit 642 ko. Si vous disposez d'un mégaoctet, c'est parfait.

Si vous n'avez que 512 ko, il vous faudra choisir une résolution inférieure. Même avec un moniteur très performant, une taille mémoire insuffisante ne permet pas de tirer pleinement avantage de celui-ci. D'autre part, si vous disposez d'un mégaoctet et que votre écran ne supporte que 800 x 600, vous ne pourrez dépasser cette résolution.

La mémoire en supplément n'est jamais perdue. XFree86 saura l'utiliser pour gérer un écran virtuel plus grand que la résolution utilisée (voir la configuration de la taille d'un écran virtuel dans XF86Config). Souvenez-vous qu'une taille mémoire de 512 ko correspond à 512 * 1024 = 524288 octets.

Si vous vous contentez de 16 couleurs, vous pouvez indiquer au serveur que vous n'utiliserez que 4 bits par point et ainsi doubler la résolution que votre carte peut supporter. Les cartes S3, par exemple, savent gérer 1024 x 768 x 256. Vous pouvez passer à 1280 x 1024 x 16 avec 4 bits par point (Depth 4, voir ce paramètre dans XF86Config).

4.1.6 Calculer les tailles de trames

Attention : cette méthode a été développée pour les moniteurs multi-synchro. Elle fonctionnera sans doute pour les moniteurs à fréquence fixe, sachant que vous ne devrez pas vous permettre de dépasser les fréquences imposées par celui-ci.

Commençons par diviser DCF par la plus grande valeur de HSF pour obtenir le nombre minimum de points par ligne disponibles.

Par exemple, supposons que vous ayez une carte SVGA Sigma Legend avec une horloge à 65 MHz et un moniteur avec une fréquence horizontale de 55 kHz. La valeur `(DCF / HSF)` est donc 1181.

Maintenant, un peu de magie ;-). Arrondissez cette valeur à la plus proche valeur multiple de 8. La plupart des cartes SVGA et S3 utilisent en effet des registres 8 bits décalés de 3 bits vers la gauche, ce qui fait réellement 11 bits. De toute façon, ça ne fait pas de mal. Donc, nous obtenons 1176.

Ceci correspond à la valeur de HFL minimum utilisable. Une valeur supérieure de HFL (donc une meilleure résolution horizontale) peut être obtenue, en diminuant la valeur de HSF. Mais le scintillement sera supérieur. Au début une petit gymnastique intellectuelle est nécessaire. Dites-vous bien que la fréquence d'horloge de votre carte vidéo donne un nombre maximal de points par seconde. Evidemment, plus vous mettez de points par ligne, plus vous mettez de temps à la parcourir, et moins vous en parcourerez.

Continuons ! De manière un peu brutale (cela se démontre, mais l'on tomberait dans des calculs certes plus précis, mais rébarbatifs), disons que **80 %** de la longueur de la trame horizontale est disponible pour la partie visible de la ligne balayée. Cela laisse la marge nécessaire pour les bords et le retour de balayage. Nous obtenons donc la valeur de 944.

Maintenant, pour obtenir une image au format 4:3, définissons la résolution verticale aux **3/4** de la résolution horizontale. Ce qui nous donne 708. Enfin, la valeur réelle de VFL est obtenue en **multipliant** cette valeur par 1.05 (toujours pour les bords et le retour de balayage). Nous obtenons 743.

Là encore, cela semble très empirique, mais nous considérons pour cette méthode que vous connaissez le minimum de choses sur votre matériel. Nous verrons plus loin dans les exemples réels que si les données de départ sont nombreuses, il est possible d'effectuer des calculs rigoureux.

A propos de 4:3, ce rapport nous rapproche du célèbre nombre d'or, (1 + sqrt(5))/2. Il semble assez plaisant d'avoir un tel format ; les résolutions classiques (640 x 480, 800 x 600 et 1024 x 768) l'approchent toutes. C'est plus psychologique que technique ; rien ne vous empêche de modifier ce rapport.

Nous avons donc : HFL=1176 et VFL=743. Divisons 65 Mhz par le produit des deux, ce qui nous donne un taux de rafraîchissement de 74.4 Hz. Excellent ! Mieux que le standard VESA ! Et vous avez une résolution de 944 x 708, ce qui est meilleur que 800 x 600. Pas mal !

Vous pouvez même améliorer le taux de rafraîchissement aux alentours de 76 Hz, en sachant que les moniteurs ont souvent une synchro horizontale de 2 kHz supérieure à la valeur nominale, et en diminuant VFL. Mais laissons ces manipulations à plus tard. Assurez-vous que votre moniteur supportera les 76 Hz (Le NEC 4D, par exemple ne peut aller qu'à 75 Hz).

Avouez que jusque-là cela n'a pas été complètement magique !

4.1.7 Magie noire et tops de synchro

Nous avons calculé les valeurs HFL et VFL pour la fréquence de l'*horloge point* que nous avons choisie. Nous avons trouvé un taux de rafraîchissement acceptable et vérifié que nous disposions d'assez de mémoire. Maintenant il est nécessaire de savoir quand et où placer les tops de synchro.

Les tops de synchro contrôlent les fréquences réelles de balayage horizontale et verticale de votre moniteur. Les valeurs HSF et VSF que vous avez extraites de votre manuel sont des fréquences nominales, et correspondent à des maxima approximatifs. Les tops de synchro injectés dans le signal vidéo par votre adaptateur permettent au moniteur de fonctionner correctement.

Vous souvenez-vous que seule une partie du temps nécessaire au balayage d'une trame est utilisée pour l'image à afficher (i.e. la résolution) ?

Synchro horizontale :

Selon les définitions précédentes, il faut HFL tops d'horloge de votre carte pour tracer une ligne complète. Appelons HR le nombre de tops correspondant seulement à la partie visible (la résolution horizontale). Donc HR < HFL par définition. Concrètement, considérons que le signal vidéo démarre à l'instant 0, et schématisons les deux valeurs comme figuré ci-dessous :

```
+----------------------------------------------------------------+
|_____________________________________                           |
|                                    |                          | unite : top
+-----------------------------------+---------------------------+
0                                  HR                           HFL
                                    |        |         |        |
                                    |<-HGT1->| <-HSP-> |<-HGT2->|
```

Maintenant, nous devons placer un top de synchro de longueur HSP entre la fin du signal vidéo correspondant à la partie visible et la fin de la trame entière. Pourquoi cela ? Parce que nous souhaitons centrer l'image au mieux et nous avons déjà vu qu'il fallait laisser un temps non négligeable, d'une part pour les bords non visibles, d'autre part pour le retour du faisceau. Si nous réussissons cela, nous obtiendrons une image correcte.

De chaque côté de HSP doivent être prévus des temps « de garde », et de stabilisation. Environ 30 tops en première approximation. Ce sont HGT1 et HGT2. En réalité HGT1 est différent de HGT2, mais vous pourrez les prendre égaux si vous partez de rien. Vous ajusterez après.

Si vous avez de la chance, ou un bon manuel accompagnant votre moniteur (ou les deux), les valeurs ci-dessus sont indiquées, le plus souvent en durée, mais il est très facile d'en déduire le nombre de points (ou de tops d'horloge) correspondant. Par exemple, mon manuel (moniteur HYUNDAI HL7682P) indique les valeurs suivantes :

```
                      HGT1          HSP           HGT2          Frequences
                      ----          ---           ----          ----------
640x480:  H(u)        0.636         3.813         1.907         31,5 kHz
VGA       V(ms)       0.318         0.064         1.048         60.0 Hz

800x600   H(us)       1.119         2.399         1.279         48,1 kHz
VESA      V(ms)       0.772         0.124         0.479         72,0 Hz

1024x768  H(us)       0.615         1.723         2.338         48,9 kHz
SVGA/NI   V(ms)       0.204         0.306         0.327         60,5 Hz

1024x768  H(us)       0.320         1.813         1.920         56,48 kHz
VESA      V(ms)       0.053         0.106         0.513         70,1 Hz

1280x1024 H(us)       0.296         0.593         2.815         64,3 kHz
Standard  V(ms)       0.124         0.078         0.498         60,1 Hz
```

Ce sont des valeurs assez courantes, certaines étant au standard VESA. Il y a fort à parier que beaucoup de moniteurs ont des valeurs semblables. Vous pouvez utiliser ces valeurs comme référence, selon votre résolution.

Pour nos calculs, prenons par exemple une valeur de HSP = 3,8 μs (VGA). Utilisant l'horloge à 65 MHz décrite, nous savons que HSP est équivalent à : 65 * (10 ^ 6) * 3.8 * (10 ^ (-6)) soit 247 tops d'horloge (^ signifie puissance).

Retour sur le schéma précédent. Comment devons nous placer nos 247 tops d'horloge ?

En utilisant notre exemple, nous avions HR=944 et HFL=1176. La différence est donc de 1176 - 944 = 232 qui est inférieur à 247 ! Nous devons donc effectuer quelques ajustements. Que pouvons-nous faire ?

En augmentant HFL de 8 et en diminuant VFL d'autant, nous obtenons une différence de 1184 - 936 = 248. On y est presque.

Ensuite, au lieu d'utiliser la valeur de 3,8 μs, nous utiliserons 3,5 μs pour HSP. Nous obtenons alors 65 * 3,5 = 227. Cela semble mieux, bien que 248 ne soit pas beaucoup plus grand que 227. Rappelons qu'il faut réserver du temps pout HGT1 et HGT2. Or il faut au moins 30 tops pour chacun. De plus ils doivent être multiples de 8, donc disons 32. Sommes-nous coincés ?

Non, heureusement. Nous allons en fait calculer la longueur de trame qu'il va nous falloir, en nous rappelant que la valeur de HFL = 1176 est un minimum. On peut l'augmenter, en diminuant la valeur HSF, donc le taux de rafraîchissement.

Continuons à chercher les bonnes valeurs. 936 est bien multiple de 8. Ajoutons HGT1, 936 + 32 = 968. Ajoutons ensuite HSP, 968 + 227 = 1195. Ajoutons enfin HGT2, 1195 + 32 = 1227. Cela ne semble pas trop mal, mais 1232 est la plus proche valeur multiple de 8.

Utilisons encore la calculatrice pour trouver la longueur du top de synchro réellement utilisé. 1232 - 32 = 1200 qui est multiple de 8. 1232 - 32 - 968 = 232, correspondant à une synchro de 3,57 μs qui est raisonnable.

De plus 936 / 1232 = 0.76 qui est assez proche des 80% que nous donnions. Nous solliciterons donc le moniteur à 52,7 kHz (65 MHz / 1232) qui est dans ses possibilités.

Synchro verticale :

En utilisant la règle des trois-quarts, nous donnerons à la résolution verticale la valeur 702. La longueur de trame verticale sera de 702 * 1,05 = 737.

Le taux de rafraîchissement de l'écran est donc égal à 65 Mhz / (737 * 1232) = 71,6 Hz. C'est excellent !

Le schéma de synchro verticale est similaire :

```
+------------------------------------------------------------------+
|_____________________________________                             |
|                                    |                            | unite : top
+-----------------------------------+----------------------------+
0                                  VR                          VFL
                                    |         |         |
                                    |<--VGT-->|<--VSP-->|
```

La synchro démarre après l'affichage de la dernière ligne. VGT correspond au temps « de garde » vertical (compté en nombre de lignes) nécessaire avant le top de synchro. Certains moniteurs se satisfont d'une valeur VGT = 0, ce que nous utiliserons dans l'exemple. Certains nécessitent 2 ou 3 lignes de stabilisation, et si un doute existe, il vaut mieux les insérer. Aucun problème ne pourra venir de là.

Revenons à notre exemple : à partir de la définition de la longueur de trame, l'unité de temps vertical correspond au temps nécessaire pour tracer une ligne horizontale complète, ce qui donne ici (1232 / 65 Mhz) = 18,95 μs.

L'expérience montre que la valeur de top de synchro verticale devrait se trouver dans l'intervalle 50 à 300 μs. Prenons par exemple 150 μs, qui correspond à 8 temps de trame horizontale (150 / 18.95).

4.1.8 Rassemblons nos idées

Nous pouvons déjà construire le début de la section **Monitor**. Nous ne nous étendrons pas sur les rubriques *Identifier*, *VendorName* et *ModelName*. Très facile.

La rubrique *Bandwidth* doit indiquer la largeur de bande passante de votre moniteur. Exemple :

```
# Bandwidth est en MHz sauf si unites specifiees

    Bandwidth   75.0
```

La rubrique *HorizSync* indique la ou les fréquences horizontales supportées par votre moniteur. On peut y indiquer une seule fréquence, plusieurs valeurs discrètes ou bien un ou plusieurs intervalles. Exemple :

```
#    HorizSync   31.5              # typique pour un moniteur a frequence fixe unique
     HorizSync   30-82             # multisync
#    HorizSync   31.5, 35.2, 37.5  # frequences de synchro multiples
#    HorizSync  15-25, 30-50       # plusieurs intervalles de frequences
```

On agira de même pour la rubrique *VertRefresh* avec les fréquences verticales.

Dans la suite de cette section se trouvent des tables des modes vidéo. Pour chaque mode, cinq rubriques sont définies : le **nom du mode**, la **fréquence d'horloge**, les **timings horizontaux** et **verticaux**, enfin les **options**. Deux présentations sont possibles : l'ancienne (celle qui était en vigueur pour XFree-2.1) se présente sous forme de lignes de valeurs numériques, chaque ligne spécifiant au serveur un mode :

```
ModeLine        < nom >   <horloge>    HR  SH1  SH2  HFL      VR  SV1  SV2  VFL
```

La nouvelle présentation, plus claire, fait précéder chaque rubrique d'un mot-clé :

```
Mode "nom"
        DotClock  <horloge>
        HTimings  HR  SH1  SH2  HFL
        VTimings  VR  SV1  SV2  VFL
        Flags     <options>
EndMode
```

Pour des raisons de simplicité, nous utiliserons la première présentation.

Le nom du mode vidéo est généralement donné entre guillemets et est utilisé comme référence dans la section **Screen** après le mot-clé **Modes**. Le nom peut être omis si la ligne décrite appartient au même mode vidéo que la précédente.

La valeur <*horloge*> représente la fréquence d'horloge (*dot clock*, que nous avons appelée DCF) qui est utilisée pour générer les timings des deux rubriques suivantes.

La rubrique *HTimings* contient quatre champs décrivant comment doit être générée chaque ligne à l'écran. Ces champs sont exprimés en nombre de pixels. La première valeur correspond à **la résolution horizontale visible**, c'est-à-dire le nombre de points illuminés (que nous avons appelé HR). La deuxième valeur indique à quel pixel le top de synchro horizontal doit commencer (HR + HGT1). Le troisième champ indique à quel pixel le top de synchro doit finir (HR + HGT1+ HSP). Le quatrième champ donne la longueur totale de la trame (HFL).

La rubrique *VTimings* contient également quatre champs. La première valeur correspond à **la résolution verticale visible**, c'est-à-dire le nombre de lignes illuminées (que nous avons appelée VR). La deuxième valeur indique à quelle ligne le top de synchro vertical doit commencer (VR + VGT). Le troisième champ indique à quelle ligne le top de synchro doit finir (VR + VGT+ VSP). Le quatrième champ donne la longueur totale de la trame verticale (VFL).

Exemple :

```
#Nom_de_mode    Horloge Timings horizontaux     Timings verticaux

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

Note : X11R5 ne supporte que les valeurs entières pour l'horloge.

Rappel : Toutes les valeurs horizontales HR, HGT1, HGT2, HFL (et donc les cumuls) sont **divisibles par 8**. Cela n'est pas requis pour les timings verticaux.

Autre exemple :

```
valeurs de timings horizontaux : 800 864 1024 1088
```

Cette ligne définit le nombre de points illuminés (800), le numéro du point à partir duquel le top de synchro démarre (864), le numéro du point à partir duquel le top de synchro s'arrête (1024), suivi du numéro du dernier point de la ligne horizontale (1088).

Le nombre de lignes depuis le haut de l'écran jusqu'en bas correspond à une trame. Le signal de base d'une trame est une ligne. Un certain nombre de lignes correspond à l'image affichée. Après la dernière ligne affichée, un délai d'un certain nombre de lignes est généré, précédant le top de synchro verticale. Celui-ci dure l'équivalent de quelques « temps lignes ». Enfin un nouveau délai de quelques lignes est nécessaire après la synchro. On retrouve de la même manière quatre valeurs :

```
valeurs de timings verticaux : 600 603 609 630
```

Cet exemple indique qu'il y a 600 lignes visibles à l'écran, que la synchro verticale démarre à la 603e ligne, s'arrête à la 609e ligne et qu'il y a au total 630 lignes.

Ces valeurs ne sont **pas forcément divisibles par 8**.

Revenons à notre exemple. Il ne nous reste plus maintenant qu'à écrire toutes ces valeurs correctement dans la section appropriée, comme suit :

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

où SH1 et SH2 correspondent respectivement au début et à la fin de la synchro horizontale, SV1 et SV2 respectivement au début et à la fin de la synchro verticale.

```
                #nom       horloge  timings horizontaux    timings verticaux   options
ModeLine        936x702    65       936 968 1200 1232      702 702  710  737
```

Aucune option n'est nécessaire ici. Nous avons fini.

Selon le matériel dont vous disposez, certaines options pourront être nécessaires. Décrivons-les rapidement :

Interlace

indique que le mode est entrelacé

DoubleScan

indique un mode ou chaque ligne est parcourue deux fois.

+HSync et -HSync

permettent de sélectionner la polarité du signal de synchronisation horizontale

+VSync et -VSync

permettent de sélectionner la polarité du signal de synchronisation verticale

Composite

permet de spécifier l'utilisation d'une synchro composite sur les matériels qui le supportent.

+CSync et -CSync

permettent de sélectionner la polarité du signal de synchronisation composite

4.1.9 Foire Aux Questions

1. **L'exemple que vous avez donné ne correspond pas à une résolution standard. Puis-je l'utiliser ?**

   Pourquoi pas ? Il n'y a aucune raison pour que vous soyez limité aux résolutions de 640 x 480, 800 x 600 ou 1024 x 768. Le serveur X vous autorise à configurer votre système avec une très grande liberté. Cela prend relativement peu de temps d'obtenir une configuration correcte. La chose importante à prendre en compte est le taux de rafraîchissement. Ne choisissez pas une résolution trop grande au prix d'un scintillement désagréable.

2. **Est-ce la seule résolution possible avec les valeurs d'horloge de 65 MHz et une HSF de 55 kHz ?**

   Absolument pas ! Nous vous invitons à suivre la procédure pour faire des essais différents jusqu'à obtenir une configuration qui vous convienne. Cette expérience peut vous amuser. La plupart des configurations vous donneront des effets étranges, mais rien ne peut vraiment abîmer votre moniteur multi-synchro (sauf si vous demandez à votre carte vidéo une cadence hors de ses limites ; de même, si vous restez relativement près des résolutions conseillées pour votre moniteur, tout ira bien). Pour les moniteurs à fréquences fixes, attention ! Ce genre d'essais par tâtonnement peut les endommager.

3. **Vous avez mentionné deux résolutions standard. Dans le fichier XF86Config, il y a plein de résolutions disponibles, pouvez-vous me dire s'il est possible d'en prendre et de les bricoler ?**

   Absolument ! Prenez par exemple la résolution standard de 640 x 480. Elle est décrite avec une fréquence de pilotage de 25 MHz, les longueurs de trames sont respectivement de 800 et 525. Le taux de rafraîchissement est donc de 59,5 Hz. Pas trop mauvais. Mais la fréquence de 28 MHz est une valeur assez courante sur les cartes vidéo SVGA. Si nous l'utilisons pour piloter cette résolution, en suivant la procédure décrite, nous obtenons des longueurs de trames de 812 et 505. Le taux de rafraîchissement passe alors à 68 Hz. Nette amélioration par rapport à la valeur standard !

4. **Comment faire avec les modes entrelacé / non-entrelacé ?**

   A une fréquence donnée, un affichage entrelacé scintillera plus qu'un non-entrelacé, ce qui explique que ce mode soit de plus en plus abandonné. Avec un scintillement accru, vous gagnez en résolution avec une horloge point plus faible. Si la valeur DCF était assez grande (90 Mhz ou plus), le mode entrelacé n'engendrerait plus de scintillement, mais aux fréquences courantes, les moniteurs fonctionnant en mode entrelacé sont assez mauvais sous X.

5. **Pouvez-vous résumer rapidement ce dont vous venez de parler ?**

   - Pour toute fréquence de pilotage donnée, l'augmentation de la résolution implique une diminution du taux de rafraîchissement, et donc introduit plus de scintillement ;
   - Si une haute résolution est souhaitable et que votre moniteur la supporte, essayez de vous procurer une carte vidéo SVGA fournissant une horloge (DCF) suffisante. Plus grande sera cette fréquence, meilleure sera votre image.

4.1.10 Deux exemples supplémentaires de calculs

Prenons une autre hypothèse :

Soit une carte vidéo ayant une horloge à 40 MHz, et un afficheur ayant une fréquence horizontale pouvant varier de 30 à 37 kHz. Nous souhaitons faire fonctionner l'ensemble à une résolution d'environ 800 x 600. Le nombre minimum de points par ligne est donc de 40000000 / 37000 = 1081,081, soit 1081 points par ligne.

Nous choisirons cette valeur dans les calculs qui vont suivre. Chaque ligne de notre affichage aura au moins 1081 points. Arrondissons-le à 1088 pour le rendre divisible par 8. Maintenant, considérons que le top de synchro horizontale doit durer 3,8 μs. Calculons à combien de points cela correspond, sachant que la "durée" d'un point est de 1 / 40000000 de seconde, soit 0.025 μs par point.

Le nombre de points correspondant à 3,8 μs est :

```
3,8 us = P points * (0,025 us / point)
```

soit

```
P points = 3,8 us / (0,025 us / point) = 152 points
```

152 est divisible par 8. S'il ne l'était pas, nous devrions choisir le multiple de 8 le plus proche. Nous avons 1088 points par ligne au total. La résolution visible sera de 800 points. Il nous reste donc à déterminer les temps qui se trouveront avant et après la synchronisation nécessaire pour l'affichage.

La règle veut que l'on mette environ 30 points de chaque côté. Dans ce cas, choisissons la valeur de 32, divisible par 8.

Or il nous reste :

```
(1088 - 800 - 152) = 136 points
```

que nous devons répartir de chaque côté de la synchro. Divisons le nombre en deux parts égales, soit 68 points avant la synchro et 68 points après.

Les quatre valeurs du fichier XF86Config seront donc

```
800     (800 + 68)      (800 + 68 + 152)        (800 + 68 + 152 + 68)
```

ou

```
800 868 1020 1088
```

Maintenant, calculons les nombres verticaux. Tout d'abord, souvenez-vous que les nombres verticaux ne sont pas exprimés en termes de points mais de lignes. Donc, nous avons à calculer le temps que prend l'affichage d'une ligne. Ce calcul est simple à faire, car nous savons que chaque ligne fait 1088 points et que chaque point dure 0,025 μs, ce qui donne :

```
(1088 points / ligne) * (0,25 us / point) = 27,2 us / ligne
```

Puisque nous avons choisi une définition de 800 points par ligne, choisissons le nombre de lignes selon le rapport de 4/3. 800 valant 4 x 200, nous aurons 600 lignes visibles. La résolution sera bien de 800 x 600.

Nous savons qu'un top de synchro vertical doit se trouver dans l'intervalle 50 à 300 μs. Si nous choisissons 150 μs comme valeur courante, nous trouverons à combien de lignes cela correspond en divisant :

```
(150 us / top) / (27,2 us / ligne) = 5,51 lignes par top
```

En arrondissant à la valeur entière immédiatement supérieure, nous obtenons 6 lignes par top de synchro.

Pour obtenir le nombre total de lignes par trame (lignes illuminées plus lignes non-illuminées sur les bords), nous considérons que le nombre total de lignes est de 5% supérieur au nombre de lignes visibles. Le nombre total de lignes est de :

```
(600 lignes) * 1,05 = 630 lignes par trame
```

Nous devons placer le top de synchro dans les lignes non visibles, entre la fin des lignes visibles et la fin de la trame. Puisque nous disposons de 630 lignes, dont 600 illuminées, 6 correspondant à la durée du top, il nous reste :

```
(630 - 600) - 6 = 24 lignes
```

Certains moniteurs acceptent que le top démarre immédiatement après les lignes visibles, d'autres en revanche nécessitent une ou deux lignes d'attente avant le top de synchro. De façon à être sûrs de notre coup, insérons trois lignes entre la fin des lignes illuminées et le top de synchro. Le reste des lignes sera ajouté après la fin de la synchro. Les nombres des timings verticaux sont donc :

```
600   (600 + 3)   (600 + 3 + 6)   (600 + 3 + 6 + 21)
```

ou

```
600 603 609 630
```

Avant de faire quoi que ce soit d'autre, nous devons nous assurer que le moniteur pourra supporter 630 lignes par trame à raison de 27,2 μs par ligne. Calculons le nombre de trames par seconde que génèrera notre configuration, et comparons la valeur à celle donnée dans le manuel. Pour 630 lignes par trame à raison de 27,2 μs par ligne, nous avons 630 * 27,2 = 17136 μs par trame, soit 0,017136 seconde par trame, ou encore 1 / 0,017136 trame par seconde :

```
1 / (0,017136 seconde / trame) = 58,4 trames / seconde
```

Si la valeur 58,4 Hz est dans l'intervalle des valeurs acceptées par votre moniteur, tout va bien. Si celui-ci ne supporte pas une telle vitesse de synchro verticale (taux de rafraîchissement), nous devons modifier le nombre de lignes par trame en ajustant tous les timings proportionnellement.

Combinons maintenant les valeurs horizontales et verticales pour obtenir une ligne que l'on peut mettre dans le fichier XF86Config :

```
ModeLine "800x600"      40      800 868 1020 1088       600 603 609 630
```

Nous pouvons maintenant tester la configuration. Il se peut qu'il y ait des problèmes, en raison des suppositions que nous avons faites, mais dans la grande majorité des cas, cela devrait au moins vous donner un affichage stable. Il faut encore quelques expérimentations pour obtenir quelque chose de vraiment satisfaisant.

Des calculs réels

Ma carte vidéo a un oscillateur battant à 40 MHz, je pars donc avec cette valeur d'horloge. La fréquence de synchro horizontale de mon moniteur est de 37 kHz, donc le nombre de points minimum par ligne est de 40000000 / 37000 = 1081. La fréquence de synchronisation verticale peut varier de 50 à 90 Hz.

Le manuel de mon moniteur explique que la durée du top de synchro horizontal ne peut dépasser 3,92 μs. Avezc une durée de 0,025 μs par point, le top dure :

```
(3,92 us) / (0,025 us / point) = 156,8 points
```

Arrondissons cette valeur à l'entier le plus proche divisible par 8, soit 160. Le manuel dit également que le temps séparant le dernier point illuminé du début de la synchro doit être d'au moins 0,67 μs, ce qui, à une fréquence d'horloge de 40 MHz - souvenez-vous que 40 MHz = 1 / 40000000 = 0,025 μs par point - correspond à :

```
P points = (0,67 us) / (0,025 us / point) = 26,8 points.
```

que nous arrondissons à 32 (divisible par 8).

Mon manuel dit encore que le temps suivant le top de synchro doit être d'au moins 3,56 μs. En nombre de points cela correspond à :

```
P points = (3,56 us) / 0,025 us / point) = 142,4 points.
```

que nous arrondissons à 144.

Maintenant, pour une ligne horizontale nous avons 800 points illuminés, 32 points avant la synchro, 160 points de durée de synchro et 144 points après la synchro.

```
800 + 32 + 160 + 144 = 1136
```

Une ligne fait maintenant 1136 points au lieu des 1088 (1081 arrondis) précédemment calculés, mais souvenez vous que cette valeur était le minimum de points que nous pouvions avoir. Donc 1136 points sont corrects pour démarrer.

Nous avons déjà une partie de notre configuration :

```
800  (800+32)  (800+32+160)  (800+32+160+144)
```

Une ligne de 1136 points représente 1136 * 0,025 = 28,4 μs.

La règle des 4 / 3 nous mène à une résolution verticale de 600 points visibles.

Le manuel de mon moniteur indique que la synchro verticale doit durer au moins 64 μs. En nombre de lignes, cela correspond à :

```
(64 us / synchro) / (28,4 us / ligne) = 2,25 lignes / synchro.
```

arrondi à 3 lignes.

Le manuel indique également que le temps après la dernière ligne affichée et le début de la synchro doit être d'au moins 318 μs et que le délai après la synchro doit être d'au moins 630 μs. Calculons le nombre de lignes que cela représente :

```
(318 us) / (28,4 us / ligne) = 11,20 lignes

(630 us) / (28,4 us / ligne) = 22,18 lignes
```

que nous arrondissons respectivement à 12 et 23 lignes.

La deuxième partie de la configuration correspond aux valeurs :

```
600 (600+12) (600+12+3) (600+12+3+23)
```

Nous voilà prêts. La ligne du fichier XF86Config est donc la suivante :

```
ModeLine "800x600"      40      800 832 992 1136        600 612 615 638
```

C'est le premier mode vidéo que j'ai essayé. Il m'a semblé à première vue qu'il y avait un peu de scintillement, j'ai essayé de modifier les timings et j'ai obtenu une qualité correcte pour une résolution de 784 x 614.

4.1.11 Corriger des problèmes d'image

Votre fichier de configuration est prêt. Vous lancez le serveur X et l'image ne semble pas correcte. Que faire ? Voici quelques solutions.

Vous **déplacez** l'image en changeant les timings de synchro. Vous **dimensionnez** l'image en changeant la longueur de trame (vous devez modifier aussi les tops de synchro pour garder une image centrée, sinon, le dimensionnement déplace l'image).

Encore quelques petites choses : les positions horizontale et verticale sont indépendantes. Le déplacement horizontal de l'image n'affecte pas sa position verticalement, et réciproquement. Cependant, ce n'est pas tout à fait vrai pour le dimensionnement. Alors que la modification de la largeur de l'image n'affecte pas sa hauteur et vice-versa, la modification des deux peut être limitée. En particulier, si votre image est trop large dans les deux dimensions, vous résoudrez le problème avec une fréquence d'horloge supérieure, cela augmentant la résolution disponible.

L'image est déplacée à gauche ou à droite

Pour corriger cela, déplacez la synchro horizontale. C'est-à-dire

augmentez ou diminuez (par saut de 8) les deux valeurs centrales des timings horizontaux qui définissent les temps avant et après la synchro horizontale.

Si l'image est déplacée vers la gauche (le bord droit est trop large), et que vous souhaitez la déplacer à droite, diminuez ces deux valeurs. Si l'image est déplacée vers la droite (le bord gauche est trop large), et que vous souhaitez la déplacer à gauche, augmentez ces deux valeurs.

L'image est déplacée vers le haut ou vers le bas

Pour corriger ce problème, vous devez déplacer la synchro verticale, c'est-à-dire augmenter ou diminuer les deux valeurs centrales des timings verticaux, dans le fichier XF86Config, définissant les temps avant et après la synchro.

Si l'image est déplacée vers le haut (bord inférieur trop large), et que vous souhaitez la descendre, diminuez les deux valeurs. Si l'image est déplacée vers le bas (bord supérieur trop large), et que vous souhaitez la monter, augmentez les deux valeurs.

L'image est trop large (resp. trop étroite)

Diminuez (resp. augmentez) la longueur de trame horizontale. C'est-à-dire, modifiez les quatre nombres de la rubrique horizontale. Pour éviter de déplacer l'image, déplacez aussi la synchro (2ème et 3ème valeurs) de moitié moins.

L'image est trop grande (resp. trop réduite) verticalement

Diminuez (resp. augmentez) la longueur de trame verticale. C'est-à-dire, modifiez les quatre nombres de la rubrique verticale. Pour éviter de déplacer l'image, déplacez aussi la synchro (2ème et 3ème valeurs) de moitié moins.

Toute distorsion qui ne puisse être maîtrisée par ces méthodes correspond certainement à une erreur plus fondamentale, comme une erreur de calcul ou une horloge trop rapide pour le moniteur.

Finalement, souvenez-vous que le fait d'augmenter la longueur de trame, diminuera la fréquence de rafraîchissement, et vice-versa.

4.1.12 Quelques exemples réels de Modes vidéo

```
#
# 640x480@60Hz Mode non-entrelace
# Synchronisation horizontale = 31.5kHz
# Timing: H=(0.95us, 3.81us, 1.59us), V=(0.35ms, 0.064ms, 1.02ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     25.175  640  664  760  800    480  491  493  525

#
# Alternate 640x480@60Hz Mode non-entrelace
# Synchronisation horizontale = 31.5kHz
# Timing: H=(1.27us, 3.81us, 1.27us) V=(0.32ms, 0.06ms, 1.05ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     25.175  640  672  768  800    480  490  492  525

#
# 640x480@63Hz Mode non-entrelace (non-standard)
# Synchronisation horizontale = 32.8kHz
# Timing: H=(1.41us, 1.41us, 5.08us) V=(0.24ms, 0.092ms, 0.92ms)
```

```
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     28.322  640  680  720  864    480  488  491  521

#
# 640x480@70Hz Mode non-entrelace (non-standard)
# Synchronisation horizontale = 36.5kHz
# Timing: H=(1.27us, 1.27us, 4.57us) V=(0.22ms, 0.082ms, 0.82ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     31.5    640  680  720  864    480  488  491  521

#
# VESA 640x480@72Hz Mode non-entrelace
# Synchronisation horizontale = 37.9kHz
# Timing: H=(0.76us, 1.27us, 4.06us) V=(0.24ms, 0.079ms, 0.74ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     31.5    640  664  704  832    480  489  492  520

#
# VESA 800x600@56Hz Mode non-entrelace
# Synchronisation horizontale = 35.1kHz
# Timing: H=(0.67us, 2.00us, 3.56us) V=(0.03ms, 0.063ms, 0.70ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "800x600"     36      800  824  896 1024    600  601  603  625

#
# Alternate 800x600@56Hz Mode non-entrelace
# Synchronisation horizontale = 35.4kHz
# Timing: H=(0.89us, 4.00us, 1.11us) V=(0.11ms, 0.057ms, 0.79ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "800x600"     36      800  832  976 1016    600  604  606  634

#
# VESA 800x600@60Hz Mode non-entrelace
# Synchronisation horizontale = 37.9kHz
# Timing: H=(1.00us, 3.20us, 2.20us) V=(0.03ms, 0.106ms, 0.61ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "800x600"     40      800  840  968 1056    600  601  605  628 +hsync +vsync

#
# Alternate 800x600@60Hz Mode non-entrelace
# Synchronisation horizontale = 37.9kHz
# Timing: H=(1.20us, 3.80us, 1.40us) V=(0.13ms, 0.053ms, 0.69ms)
```

```
#
# name        clock   horizontal timing     vertical timing      flags
 "800x600"     40      800 848 1000 1056     600  605  607  633

#
# VESA 800x600@72Hz Mode non-entrelace
# Synchronisation horizontale = 48kHz
# Timing: H=(1.12us, 2.40us, 1.28us) V=(0.77ms, 0.13ms, 0.48ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "800x600"     50      800  856  976 1040    600  637  643  666  +hsync +vsync

#
# 1024x768@43.5Hz, mode entrelace (standard 8514/A)
# Synchronisation horizontale = 35.5kHz
# Timing: H=(0.54us, 1.34us, 1.25us) V=(0.23ms, 0.23ms, 0.93ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768i"   44.9   1024 1048 1208 1264    768  776  784  817  Interlace

#
# VESA 1024x768@60Hz Mode non-entrelace
# Synchronisation horizontale = 48.4kHz
# Timing: H=(0.12us, 2.22us, 2.58us) V=(0.06ms, 0.12ms, 0.60ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768"    65     1024 1032 1176 1344    768  771  777  806 -hsync -vsync

#
# 1024x768@60Hz Mode non-entrelace (horloge point non-standard)
# Synchronisation horizontale = 48.4kHz
# Timing: H=(0.65us, 2.84us, 0.65us) V=(0.12ms, 0.041ms, 0.66ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768"    62     1024 1064 1240 1280   768  774  776  808

#
# VESA 1024x768@70Hz Mode non-entrelace
# Synchronisation horizontale = 56.5kHz
# Timing: H=(0.32us, 1.81us, 1.92us) V=(0.05ms, 0.14ms, 0.51ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768"    75     1024 1048 1184 1328    768  771  777  806 -hsync -vsync

#
# 1024x768@70Hz Mode non-entrelace (horloge point non-standard)
# Synchronisation horizontale = 56.25kHz
# Timing: H=(0.44us, 1.89us, 1.22us) V=(0.036ms, 0.11ms, 0.53ms)
```

```
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768"    72     1024 1056 1192 1280    768  770  776 806   -hsync -vsync

#
# 1024x768@76Hz Mode non-entrelace
# Synchronisation horizontale = 62.5kHz
# Timing: H=(0.09us, 1.41us, 2.45us) V=(0.09ms, 0.048ms, 0.62ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1024x768"    85     1024 1032 1152 1360    768  784  787  823

#
# 1280x1024@44Hz, Mode entrelace
# Synchronisation horizontale = 51kHz
# Timing: H=(0.02us, 2.7us, 0.70us) V=(0.02ms, 0.24ms, 2.51ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1280x1024i"  80     1280 1296 1512 1568   1024 1025 1037 1165  Interlace

#
# Alternate 1280x1024@44Hz, Mode entrelace (horloge point non-standard)
# Synchronisation horizontale = 47.6kHz
# Timing: H=(0.42us, 2.88us, 0.64us) V=(0.08ms, 0.12ms, 0.96ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1280x1024i"  75     1280 1312 1528 1576   1024 1028 1034 1080  Interlace

#
# 1280x1024@59Hz Mode non-entrelace (non-standard)
# Synchronisation horizontale = 63.6kHz
# Timing: H=(0.36us, 1.45us, 2.25us) V=(0.08ms, 0.11ms, 0.65ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1280x1024"  110     1280 1320 1480 1728   1024 1029 1036 1077

#
# 1280x1024@61Hz, Mode non-entrelace
# Synchronisation horizontale = 64.25kHz
# Timing: H=(0.44us, 1.67us, 1.82us) V=(0.02ms, 0.05ms, 0.41ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "1280x1024"  110     1280 1328 1512 1712   1024 1025 1028 1054

#
# 1280x1024@74Hz, Mode non-entrelace
# Synchronisation horizontale = 78.85kHz
# Timing: H=(0.24us, 1.07us, 1.90us) V=(0.04ms, 0.04ms, 0.43ms)
```

```
#
# name        clock   horizontal timing     vertical timing      flags
 "1280x1024"  135     1280 1312 1456 1712   1024 1027 1030 1064
```

4.2 La section Device

Cette section n'est pas particulièrement compliquée mais il convient d'y prêter attention. Pour la raison déjà indiquée au chapitre 3, nous ne nous étendrons pas trop sur cette section. Nous passerons bien évidemment sur les rubriques *Identifier*, *VendorName* et *BoardName*.

Les rubriques *Chipset*, *Ramdac* et *Videoram* peuvent être obtenues grâce à l'utilitaire `SuperProbe`. Lancez-le sur une console texte. Voir le chapitre 5.3

La rubrique *Clocks* correspond à l'ensemble des fréquences utilisables par votre carte vidéo. Pout obtenir ces valeurs vous pouvez au choix :

- Consulter la documentation et notamment le fichier */usr/lib/X11/doc/modeDB.txt*. Vous y trouverez peut-être les valeurs pour votre carte vidéo.
- Lancer la commande : `X -probeonly`. Les détails sont fournis dans le livre « Le système Linux » décrit au chapitre 3.1, ou

  dans le XFree86-HOWTO.
- Lancer X normalement. Si aucune rubrique **Clocks** n'est définie, le serveur calculera les valeurs. Avec un peu de chance et surtout le bon serveur, vous obtiendrez un affichage correct. Appuyez sur les touches Ctrl+Alt+F<n> (où n correspond à la console à partir de laquelle vous avez lancé X). Notez les valeurs que X a calculées et mettez-les dans le fichier **XF86Config**.

Note : Si vous faites un test (avec ou sans option `-probeonly`) sur votre machine, faites-le avec le minimum d'applications en cours. Le serveur X étant une application également, les boucles de timings peuvent être perturbées par des activitiés du disque et rendre les valeurs imprécises. Effectuez le test plusieurs fois afin de vérifier que les valeurs sont stables ; sinon, tuez des processus en activité jusqu'à ce qu'elles le soient. Utilisateurs SVR4 : le processus *mousemgr* est connu pour être perturbant.

Afin d'éviter que la recherche des timings soit effectuée à chaque lancement de X et risquer que ceux-ci soient imprécis, il est préférable de les figer en les indiquant dans le fichier XF86Config. Cela supprime la recherche et donne au serveur une liste exacte des valeurs qu'il doit utiliser. En utilisant les données du paragraphe 4.1.2 on obtiendrait :

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

Sur des systèmes déjà chargés, vous éviterez ainsi des démarrages variables de X, ou même des échecs. Il est tout à fait possible que X échoue parce que les timings qu'il a trouvés sont faux et qu'ils ne correspondent à aucune valeur connue d'*horloge point* (dot clock).

5 Les applications d'aide à la configuration

Pour vous aider dans vos calculs nombreux et fastidieux, un certain nombre de petits programmes ont été écrits. Ils ne prennent que peu de place et il est conseillé de les installer sur votre disque.

5.1 Vgaset

Un petit programme pour vous aider à configurer votre serveur X. Son rôle est de faciliter la recherche de la meilleure résolution de votre écran. Il tourne dans une fenêtre *xterm* (X doit donc déjà tourner), et affiche la géométrie actuelle de votre écran, sous une forme similaire à celle décrite dans le fichier XF86Config. Vous pouvez modifier cette géométrie en temps réel, à l'aide d'un certain nombre de commandes décrites ci-après.

Vous pouvez ainsi déterminer la configuration qui correspond à votre souhait, ou revenir à tout moment à la configuration initiale.

Attention

L'auteur et moi-même[1] vous mettons en garde sur l'utilisation de ce programme. Vgaset possède les caractéristiques nécessaires pour vous éviter d'endommager votre moniteur. Cependant, si vous spécifiez une plage de fréquences que votre moniteur ne supporte pas, ou bien une mauvaise horloge, *vgaset* peut causer des dommages irréversibles à votre matériel. Si vous ne connaissez pas toutes les valeurs, lancez *vgaset* sans option. Ne laissez en aucun cas votre moniteur fonctionner longtemps avec de mauvaises valeurs. Si quelque chose ne va pas, éteignez-le ! Utilisez *vgaset* à vos risques et périls.

Remarque

Je vous sens fiévreux tout à coup ! N'ayez pas peur, ceci n'est pas fréquent. Allez-y en faisant attention.

5.1.1 Les options de vgaset

Les options de la ligne de commande sont les suivantes :

-V

Rend *vgaset* verbeux. En particulier, il affiche les registres CRTC.

-d *horloge*

Définit la valeur de l'horloge point. Ceci permet à *vgaset* de donner des informations sur les timings et de les utiliser pour déterminer des intervalles de fréquences. *horloge* peut etre spécifé en **hertz**, **kilohertz** ou **megahertz**. Faites bien attention à donner la bonne valeur.

-x *fréquence*

spécifie la fréquence horizontale minimale (en **hertz** ou **kilohertz**) supportée par votre moniteur. Par défaut la valeur prise est la valeur courante. Cette option est ignorée si l'option `-d` *horloge* est absente, sauf pour `-x0`.

1. Le traducteur

-X *fréquence*

spécifie la fréquence horizontale maximale (en **hertz** ou **kilohertz**) supportée par votre moniteur. Par défaut la valeur prise est la valeur courante. Cette option est ignorée si l'option `-d` *horloge* est absente.

-y *fréquence*

spécifie la fréquence verticale minimale supportée par votre moniteur. Par défaut la valeur prise est la valeur courante. Cette option est ignorée si l'option `-d` *horloge* est absente, sauf pour `-y0`.

-Y *fréquence*

spécifie la fréquence verticale maximale supportée par votre moniteur. Par défaut la valeur prise est la valeur courante. Cette option est ignorée si l'option `-d` *horloge* est absente.

-s *config*

donne les valeurs horizontales de début de synchro, de fin de synchro et du nombre total de points, et les valeurs verticales de début de synchro, de fin de synchro et du nombre total de lignes. Ces valeurs correspondent aux trois dernières valeurs de chaque rubrique d'un mode vidéo :

```
"640x480"      28     640 680 728 776         480 480 482 494
                          --- --- ---             --- --- ---
```

-v *résolution*

spécifie la valeur de la résolution virtuelle. Ceci vous permet d'augmenter la résolution affichée, au mieux jusqu'à ces valeurs. Celles-ci ne doivent pas dépasser les valeurs données à la ligne *Virtual* de la section *Screen*.

Exemple :

```
vgaset -d80 -x30 -X64 -y50 -Y100 -v 1152 910

        --> correspond a un ecran Crystal Scan 1572 FS a 80 MHz
```

5.1.2 Les commandes de vgaset

Elles sont activées en appuyant simplement sur une des touches décrites ci-après, colonne de gauche :

```
l       diminue la marge gauche
L       augmente la marge gauche
r       diminue la marge droite
R       augmente la marge droite
h       diminue la duree de synchro horizontale
H       augmente la duree de synchro horizontale
t       diminue la marge du haut
T       augmente la marge du haut
b       diminue la marge du bas
B       augmente la marge du bas
v       diminue la duree de synchro verticale
V       augmente la duree de synchro verticale
```

```
x       diminue la resolution horizontale
X       augmente la resolution horizontale
y       diminue la resolution verticale
Y       augmente la resolution verticale
?       reprend les valeurs initiales
RC      retour-chariot : reaffiche la configuration
q       quitte
EOT     quitte
```

Après chaque commande, *vgaset* réaffiche la configuration selon une présentation compatible avec la rubrique *ModeLine* du fichier de configuration.

5.1.3 Les bugs de vgaset

Un certain nombre ! Non, pas trop quand même, mais ce programme a été testé seulement sur certaines cartes vidéo (*ET4000*, *Diamond SpeedStar*, *ATI Graphics UltraPro*). Suivez l'actualité pour découvrir d'éventuelles mises à jour ou savoir si des problèmes ont été détectés.

Le serveur est perturbé lorsque vous modifiez le nombre de pixels. Si vous diminuez la valeur, le serveur ne déroulera plus l'écran virtuel entier (scrolling). Si vous l'augmentez, il n'utilisera pas les nouveaux pixels (seulement avec *vgaset*). Cependant c'est utile pour déterminer la meilleure résolution.

Il n'est pour l'instant pas possible de modifier la valeur de l'horloge point (*dot clock*) avec *vgaset*.

5.2 xclk

xclk est un autre petit utilitaire qui vous aide à calculer les valeurs pour votre configuration. Il est basé sur les heuristiques et règles que nous avons fixés dans ce document.

Pour utiliser *xclk*, vous devez connaître les fréquences supportées par votre carte vidéo et votre moniteur (Un moniteur multi-synchro est d'ailleurs conseillé). Les valeurs d'horloge peuvent être déterminées notamment avec le programme **clock.exe** disponible dans les archives Linux, ou encore en utilisant les utilitaires fournis avec votre carte vidéo (Ex : les cartes ET4000 sont livrées avec les utilitaires DMODE et VMODE sous DOS). Il y a trois façons d'utiliser *xclk* :

```
xclk       <Frequence_horloge>  <frequences_horiz>
xclk    -h <Frequence_horloge>  <Pixels_horiz>
xclk    -r <Frequence_horloge>  <taux_rafraichissement>
```

Les fréquences horloge sont en **MHz**, les fréquences horizontales en **kHz** (ne donnez que les valeurs).

La première syntaxe utilise la fréquence d'horloge de votre carte et la fréquence horizontale de votre moniteur.

Exemple : Ma carte a une horloge de 64 MHz et mon moniteur supporte une fréquence horizontale de 56 kHz. J'utilise la commande :

```
xclk 64 56              # premiere syntaxe
```

Ceci génère deux lignes compatibles avec la rubrique **ModeLine** du fichier de configuration (ajoutez éventuellement le mot-clé *ModeLine* si vous souhaitez utiliser la ligne dans le fichier) :

```
"896x672"       64      896  928  1160  1192    672  672  683  706
# Tx de rafraichissement = 76.05Hz ; Frequence horizontale = 53.69 kHz
```

Il se peut que *xclk* donne des valeurs ridicules. Exemple `xclk 65 40` génère :

```
"1296x972"      65      1296  1328  1600  1632  972  972  980  1021
# Tx de rafraichissement = 39.01Hz ; Frequence horizontale = 39.83 kHz
```

Avec un tel taux de rafraîchissement, on croirait regarder un film du début du siècle. N'utilisez pas cette résolution. Les meilleurs taux de rafraîchissement sont autour de 70 Hz. Plus la valeur est élevée, mieux c'est (selon ce que peut supporter votre moniteur). Un MAG 15H peut supporter une valeur allant jusqu'à 120 Hz. Beaucoup vont jusqu'à 75 Hz.

Utilisons alors la troisième syntaxe, qui permet de spécifier le taux de rafraîchissement, par exemple `xclk -r 64 70` génère :

```
"944x708"       64      944  976  1208  1240    708  708  718  743
# Tx de rafraichissement = 69.47Hz ; Frequence horizontale = 51.61Khz
```

Une troisième façon d'utiliser *xclk* permet d'obtenir une résolution particulière. Exemple `xclk -h 64 944` génère :

```
"944x708"       64      944  976  1216  1248    708  708  718  743
# Refresh rate = 69.47Hz ; Horizontal Frequency = 51.61Khz
```

Comme vous pouvez le noter, certaines valeurs ont légèrement changé, malgré des données équivalentes. En fait rien de bien important puisque la différence n'excède pas 16 unités, ce qui est supporté par les moniteurs.

Si les valeurs données par *xclk* ne fonctionnent pas, recalculez les valeurs. Surtout n'essayez pas des valeurs farfelues avec votre serveur X.

5.3 SuperProbe

5.3.1 Description

SuperProbe est un utilitaire dont le rôle est de déterminer la configuration matérielle vidéo dont vous disposez, que ce soit sur un bus ISA/EISA/VLB ou PCI (sous réserve). Il recherche de différentes manières et parmi les registres connus (les machines MicroChannel et PCI peuvent poser problème ; *SuperProbe* fonctionne avec la plupart en utilisant l'option `-no_bios`). *SuperProbe* est un programme particulièrement exposé aux erreurs, notamment sous Unix pour lequel le matériel est plus ésotérique que sous DOS. Il aura donc probablement besoin d'aide de la part de l'utilisateur.

SuperProbe tourne sous SVR3, SVR4, Linux, 386BSD/FreeBSD/NetBSD, Minix-386 et Mach. Il pourrait être très facile de l'étendre aux autres systèmes Unix, voire aux systèmes non-Unix.

Actuellement, *SuperProbe* sait détecter les cartes MDA, Hercules, CGA, MCGA, EGA, VGA et tous les *chipsets* SVGA (voir l'option *-info*, ci-dessous). Il sait aussi identifier de nombreux RAMDAC[2] HiColor/TrueColor utilisés sur les cartes SVGA, ainsi que la quantité de mémoire installée (pour beaucoup de *chipsets*). Il peut détecter les cartes 8514/A et dérivées, mais pas les XGA ni PGC (l'auteur a l'intention de le faire). Il n'est pas non plus capable actuellement de reconnaître les cartes Targa, TIGA ni MicroField.

5.3.2 Les options

Les options supportées par *SuperProbe* sont les suivantes :

-verbose

Rend *SuperProbe* verbeux. Il fournit bon nombre d'informations pendant son exécution.

-no16

SuperProbe ne tentera pas d'utiliser les ports nécessitant un décodage d'adresse d'E/S[3] sur 16 bits. Les premiers bus ISA spécifiaient un décodage sur 10 bits. Par conséquent certaines anciennes cartes (dont pas mal de cartes 8 bits) décoderont mal les références aux ports utilisant les 6 bits supplémentaires et peuvent se mettre dans des états bizarres, pensant avoir été adressées alors qu'elles ne l'étaient pas. Il est recommandé de mettre cette option si votre système comporte une ou plusieurs cartes 8 bits.

-excl *liste*

SuperProbe ne tentera pas d'accéder aux ports d'E/S donnés dans la liste. Certaines cartes vidéo utilisent des ports d'E/S non standard pouvant entrer en conflit avec d'autres cartes installées. En donnant à *SuperProbe* une liste de ports déjà utilisés, il saura qu'aucune carte vidéo ne les utilise et par conséquent ne les testera pas (ce qui pourrait perturber votre système). La liste d'exclusion est donnée sous forme de liste de numéros (ou d'intervalles de numéros) de ports d'E/S séparés par des virgules. Un intervalle est donné sous la forme « inf-sup », bornes incluses. Les valeurs peuvent être données en décimal, en octal (nombres commençant par '0'), ou en hexadécimal (nombres commençant par '0x').

-mask10

Cette option est utilisée conjointement avec la liste d'exclusion (*-excl*). Elle indique à *SuperProbe* que lors du test d'un port d'E/S hors de la liste d'exclusion, l'adresse du port doit être masquée sur 10 bits. C'est important avec les anciennes cartes 8 bits ne pouvant décoder que 10 bits, ainsi que pour certaines cartes 16 bits bon marché. Cette option est moins restrictive que *-no16*.

-order *liste*

Cette option donne à *SuperProbe* une liste de *Chips* à tester, et lui indique l'ordre dans lequel il doit le faire. La liste est formée de noms séparés par des virgules. Celle-ci a priorité sur la liste prédéfinie dans *SuperProbe*. Pour connaître la liste des noms supportés, utilisez l'option *-info* décrite ci-après. Notez que les valeurs affichées comme « Standard Video hardware » ne sont pas utilisables avec l'option *-order*.

2. DAC=Digital-Analog Converter (Convertisseur Digital-Analogique)
3. Entrée/Sortie

-noprobe *liste*

Cette option définit les Chips que *SuperProbe* ne doit **pas** tester. L'ordre de tests est soit celui par défaut, soit celui imposé par l'option *-order*. La liste contient des noms de chips séparés par des virgules. Pour connaître la liste des noms supportés, utilisez l'option *-info* décrite ci-après. Notez que les valeurs affichées comme « Standard Video hardware » ne sont pas utilisables avec l'option *-noprobe*.

-bios *base*

Cette option indique l'adresse de base du BIOS de la carte graphique. Par défaut, *SuperProbe* essaiera de localiser celle-ci de lui-même (l'adresse normale étant 0xC0000). S'il échoue (un message d'erreur sera affiché si cela arrive), utilisez l'option *-bios* pour lui indiquer la base.

-no_bios

Invalide la lecture du BIOS Vidéo et suppose qu'une carte EGA ou plus récente (VGA, SVGA) est présente comme première carte vidéo.

-no_dac

Supprime la recherche du type de RAMDAC lorsqu'une carte (S)VGA est identifiée.

-no_mem

Supprime la recherche de la taille mémoire de la carte vidéo.

-info

SuperProbe imprimera une liste de tous les matériels vidéo qu'il sait identifier.

5.3.3 Les bugs

Il y en a sûrement ! Si vous en trouvez, veuillez, svp, les rapporter à l'auteur.

Il est possible que SuperProbe bloque votre machine. Dans ce cas, effectuez l'essai avec les options *-no16, -excl* et *-mask10* de façon à éviter les conflits avec d'autres cartes installées.

6 Informations concernant ce document