



HAL
open science

Manuel d'utilisation FLIP (User's manual)

Gilles Kahn

► **To cite this version:**

Gilles Kahn. Manuel d'utilisation FLIP (User's manual). RT-0002, INRIA. 1981, pp.11. inria-00070152

HAL Id: inria-00070152

<https://inria.hal.science/inria-00070152>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (3) 954 90 20

Rapports Techniques

N° 2

MANUEL D'UTILISATION

FLIP

USER'S MANUAL

Gilles KAHN

Juin 1981

FLIP

Manuel d'utilisation

FLIP

Manuel d' utilisation

Résumé

Ce document est le manuel d'utilisation du système FLIP. Ce système a été conçu pour permettre la réalisation très rapide de transparents de couleur, tels qu'on en utilise pour les présentations techniques, les cours ou les exposés de présentation générale.

FLIP

User's Manual

Abstract

This document is the User's Manual for the FLIP system. This system has been devised to allow a very quick production of color transparencies such as those used in technical presentations, lectures or talks directed to a non specialist audience.

FLIP

Gilles Kahn
INRIA, Rocquencourt

Introduction

FLIP est un système de production de transparents pour rétroprojecteurs. A partir d'une description très sommaire comportant des indications de couleurs et de proportions, il est possible de produire automatiquement un transparent sur un traceur couleur.

Les objectifs du système sont très simples:

1. Permettre une construction de transparents standards *plus rapide* qu'à la main.
2. Permettre l'inclusion de n'importe quel dessin productible sur le traceur dans un transparent.

Le nom de FLIP vient du jargon technique utilisé dans certains milieux pour parler des transparents. FLIP est un programme pour produire des flips. Aller faire un transparent, c'est aller faire un flip. Si l'on veut, FLIP est un flippeur. Si quelqu'un, comme l'auteur, a déjà produit des transparents avec FLIP, on dira qu'il a flippé. Cette terminologie se traduit assez bien en anglais.

Principes Généraux

FLIP produit un transparent à partir d'une description textuelle. Dans cette description, aucune indication absolue de taille physique n'est nécessaire. Les proportions des sous-parties d'un flip sont indiquées par des entiers qui n'ont de signification que par rapport aux sous-parties collatérales. La taille des caractères est toujours calculée automatiquement en fonction de la quantité de texte à écrire. En l'absence d'indication de couleur, le noir est utilisé.

Il est indispensable, à mon avis, de se rendre compte que produire un flip n'est pas un travail de composition de même nature que produire un texte justifié, avec des coupures, des indentations très précises, etc... Ce n'est même pas de même nature que de produire la page de garde d'un livre. Un flip va rester sur l'écran 10 minutes au maximum, et peut-être une fois seulement. Il va comporter de grosses lettres pour les gens du fond et souvent une figure pour réveiller les dormeurs.

FLIP a été conçu avec cette idéologie. Si vous vous servez de FLIP pour faire autre chose que des flips, cette idéologie aura probablement pour conséquence qu'il vous manquera une primitive. Vous aller passer de la catégorie utilisateur de FLIP à la catégorie «hacker» de FLIP.

Opération

1. Il vous faut d'abord préparer un texte ascii avec votre éditeur préféré. Ce texte est la description d'un flip en FLIP. Il doit être conservé dans un segment *suffixé* par `flip`, disons `exemple.flip`.
2. Après vous être connecté sur une console branchée derrière un traceur HP appeler
`flip exemple`
3. Si vous souhaitez une version propre de votre segment d'entrée, appelez le formatteur de flips:

`fliprint exemple`

Pour `flip` ou `fliprint`, le temps CPU moyen consommé est de l'ordre de 0.5 secondes par flip.

Si votre flip est assez gros il peut contenir des erreurs de syntaxe. En utilisant d'abord `fliprint` puis en cas de succès `flip`, vous économiserez votre tension nerveuse, déjà mise à rude épreuve par le peu de temps qui vous reste pour produire des flips avant la date fatidique de votre exposé.

Conventions lexicales

Un texte de flip est fabriqué avec les entités lexicales suivantes:

Mots Clefs: Entièrement en caractères minuscules.

Voici la liste de tous les mots clefs:

`horiz vertic en encadrer ecrire fin fincadre
couvrir avec pivoter`

Adjectifs: Entièrement en majuscules.

Voici tous les adjectifs acceptés, pour le moment.

`BLANC NOIR BLEU VERT ROUGE`

Nombres: Entiers sans signe.

Exemples: `0 70 555`

Remarque: Soyez raisonnable, n'utilisez pas de nombres de plus de 3 chiffres.

Ligne: Depuis un tiret jusqu'à la fin de la ligne.

Exemple: `-Texte d'une ligne`

Chaînes: Suite de caractères entre simples apostrophes.

Exemples: `'Salut' 'Ici FLIP'`

Symboles spéciaux: Le seul symbole spécial utilisé par FLIP est ':'

Flips Elémentaires

1. Cadres

<flip> ::= Adjectif

Demande le tracé d'un cadre de couleur. Si BLANC est la couleur, le cadre est transparent.

Exemple:

ROUGE

Tracé un cadre rectangulaire rouge.

Notes:

-Si la couleur demandée est inconnue de FLIP (INDIGO par exemple), vous obtiendrez du noir.

-Pour savoir à quoi sert un cadre blanc, reportez vous à la section effets spéciaux.

2. Texte

<flip> ::= Ligne +

Le texte demandé est écrit en respectant absolument le découpage en lignes. La ligne la plus longue est centrée. Les autres lignes sont cadrées à gauche par rapport à cette ligne. La *hauteur* des caractères est déterminée en fonction de la place verticale disponible et du nombre de lignes à écrire. Ensuite, FLIP essaie de fabriquer des caractères avec des proportions normales. Si il est impossible de faire tenir la ligne la plus longue dans la dimension horizontale, les caractères sont écrasés en largeur, donnant un aspect étiré vers le haut. Toutes les lignes sont écrites avec les mêmes caractères.

Exemple:

-Transparent
-Normal

Flips Colorés

1. Cadres

<flip> ::= en Adjectif encadrer <flip> fincadre

Le sous-flip est encadré d'un cadre de couleur. Pour le choix de la couleur, voir les règles ci-dessus pour la construction des cadres vides.

Exemple:

```
en ROUGE encadrer
-FLIP
-FLIP
-FLIP
fincadre
```

2. Couleur du texte

<flip> ::= en Adjectif écrire <flip>

La couleur par défaut du texte à écrire dans le sous-flip est celle spécifiée par l'adjectif.

Exemples:

```
en ROUGE écrire
-This
-is
-a flip
```

ou encore

```
en ROUGE encadrer
en VERT écrire -INRIA
fincadre
```

Flips Composés

<flip> ::= horiz (Nombre:<flip>)+ fin
<flip> ::= vertic (Nombre:<flip>)+ fin

Nota: Les parenthèses utilisées ici sont des méta-parenthèses.

Un flip est décomposé en bandes horizontales (resp. verticales). Dans chaque bande s'inscrit un sous-flip. La taille de la bande (hauteur dans le premier cas, largeur dans le second) est à proportion du nombre indiqué, par rapport à la somme des nombres de la liste de bandes.

Exemple:

```
horiz
50:ROUGE
80:-FLIP
-a des possibilites
-tres reduites
70: en ROUGE écrire -INRIA
fin
```

La page est découpée en 3 bandes horizontales dans les proportions de 25%, 40% et 35%. Dans chacune de ces bandes s'inscrit un sous-flip.

Remarque: la couleur spécifiée dans un ordre

en Adjectif écrire <flip>

s'applique à tout le <flip>. Les couleurs s'empilent donc comme les déclarations de variables en Algol ou autres. Par exemple:

```

en ROUGE encadrer
  en BLEU écrire
    horiz
      50: -Salut
      50: en VERT écrire -Voilà du vert
      50: -retour au bleu
    fin
  fincadre

```

Flips Multicouches

<flip> ::= couvrir <flip1> avec <flip2> ... fin

Les flips sont superposés dans le même cadre.

Exemple:

```

couvrir
  en ROUGE écrire -ROUGE      VIOLET
avec
  en BLEU  écrire -      BLEU VIOLET
fin

```

Effets Spéciaux

1. Transparents à plusieurs niveaux. Utiliser des boîtes blanches.
2. Faire deux cadres rectangulaires concentriques. Utiliser des boîtes blanches.
3. Centrer 3 lignes consécutives. Réfléchir à la différence entre:

```

-Court
-Un peu long
-Extremement long

```

et

```

horiz
  50: -Court
  50: -Un peu long
  50: -Extremement long
fin

```

4. Cadrage à droite.

Après le tiret de début de texte, tous les caractères comptent jusqu'à la fin de la ligne, y compris les blancs. Le système utilise des caractères de taille constante. C'est à dire que si votre entrée est cadrée à droite, elle le restera.

5. Tirer des traits horizontaux ou verticaux. Ce sont des boites de hauteur ou de largeur nulle

6. Accents, soulignement. Utiliser le caractère de retour arrière du chariot.

Flips Pivotés

`<flip> ::= pivoter <flip>`

Le sous-flip est pivoté d'un quart de tour dans le sens positif, c'est à dire dans le sens contraire des aiguilles d'une montre, par rapport à son contexte. Cela permet, par exemple d'écrire verticalement ou à l'envers.

Où FLIP s'éclate

`<flip> ::= Chaîne`

La chaîne de caractères, sans ses apostrophes, suivie de 5 nombres est passée au niveau commande. Les quatre premiers nombres dénotent respectivement l'abscisse et l'ordonnée du coin en haut à gauche, la largeur et la hauteur disponible. Ces nombres sont exprimés en unités par défaut du traceur, c'est à dire en huitièmes de millimètres. Le cinquième nombre, compris entre 0 et 3 indique l'orientation à donner au dessin dans ce cadre, avec la convention que 0 signifie le Nord vers le haut, 1 signifie le Nord vers la gauche et ainsi de suite.

Exemple (hypothétique):

```
horiz
  50: -Pla Genere automatiquement
  80: 'tracer mafigure'
  50: 'monlogo'
fin
```

Les processeurs tracer et monlogo sont probablement totalement distincts.

Nota

Si l'on veut un flip avec des dimensions inhabituelles, il suffit de spécifier le coin en haut à gauche et les dimensions en x et en y à *l'appel* de FLIP. Dans ce cas il faut aussi spécifier l'orientation à donner au FLIP.

`flip exemple 0 700 700 700 0`

Cette possibilité permet en outre d'appeler FLIP récursivement

```
horiz
  50: en ROUGE encadrer
    'flip exemple'
  fincadre
  50: en BLEU encadrer
    'flip exemple5'
  fincadre
fin
```

Programmes Associés à FLIP

fliprint

Ce programme formate un texte en FLIP et détecte les erreurs de syntaxe. L'effet de la récupération d'erreurs est mis en évidence. Sur option, on peut obtenir une version anglaise de son flip d'entrée. (Enfin, seuls les mots-clefs changent, pas le texte...).

Exemples:

```
fliprint mon_flip
```

```
fliprint mon_flip -english
```

flip_ascii_file,faf

Ce programme permet de sortir un segment ascii quelconque sur le traceur. Le texte sera écrit en noir. Le nom du segment est à donner en entier car faf ne fait pas d'hypothèse sur le suffixe. Avec l'option -couleur faf interprète le caractère slash inversé dans le segment comme spécifiant un changement de couleur. Le caractère qui suit slash inversé doit être un chiffre entre 1 et 4, avec 1,2,3,4, signifiant respectivement noir, bleu, rouge, vert. Un slash inversé seul revient à la couleur par défaut, c'est à dire noir. Ce système a été mis en place pour permettre l'écriture de préprocesseurs, qui génèrent ainsi un texte multicolore. Si vous trouvez qu'il est difficile de préparer à la main un texte sous cette forme, soyez rassurés, c'est exprès, pour vous dissuader de le faire.

Exemple d'appels de faf:

```
faf TRUC.pascal
faf TRUC.pascal -couleur
```

Remarque: Mentor permet de sortir les programmes Pascal en trichromie selon ces conventions. Les mots clefs sont en bleu, les commentaires en vert et le reste du texte est en noir.

Notes diverses

1. Il existe une interface de FLIP avec la série Tektronix 4014, 4015. Pour tous renseignements, s'adresser à l'auteur.

2. Attention, FLIP ne tolère pas les marques de tabulation dans les textes, en particulier dans les segments passés à faf. Il est nécessaire d'éliminer ces marques de tabulation avec votre éditeur de textes favori. Cependant, il n'y a aucune objection à utiliser les marques de tabulation pour bien présenter un texte en FLIP.

Remerciements

Le travail fait antérieurement par J.M. Hullot pour interfacer le traceur HP a facilité considérablement l'implémentation de FLIP. Par ailleurs J.M Hullot et Jean Vuillemin ont bien voulu servir de cobayes comme utilisateurs de FLIP.

Transparent

Normal

FLIP

FLIP

FLIP

Salut

FLIP

a des possibilites
tres reduites

INRIA

Voila du vert

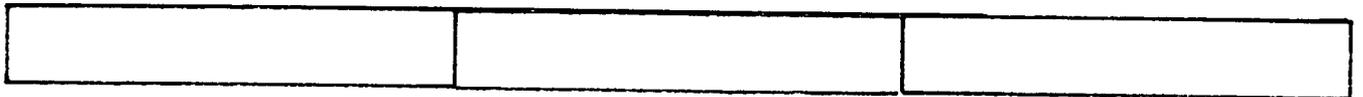
retour au bleu

Les entrées du MICRO-PLA

Conditions

Opcode

Etat



Les sorties du MICRO-PLA

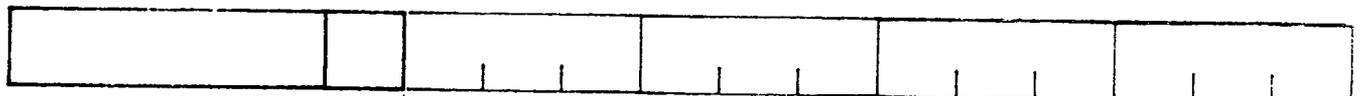
Etat

Opcode

source1

source2

destination



MICRO-instruction pour le NANO-PLA

Opcode pour la micro-instruction de sortie est:
car, cdr, rplaca, rplacd, cons ou move.

Un bit est réservé en sortie pour décider si le
nouvel état du PLA est codé dans Etat ou s'il faut
charger EVAL-STATE pour le déterminer.

(Oeuvre d'art due à Jean-Marie Hullot)