

FBM

SPECIALISTE
PRATIQUE

LES PARAMETRES DES DISQUES IDE/AT

Décoder toutes leurs informations

49.04.79.00

Contact: M. Delhommeau

IDG. InfoPC

TELEPHONER

LE FRANCOIS DUVIVIER

Les contrôleurs de disque dur pour PC ont connu leur première «révolution culturelle» avec l'arrivée de l'IBM AT, en 1984. Cette machine comprenait une carte contrôleur de type ST506 à encodage MFM (Modified Frequency Modulation) d'origine Western Digital et un Bios pilotant le disque dur intégré en standard dans le Bios de l'IBM AT.

Pour des raisons de compatibilité, tous les modèles suivants et tous les «compatibles» ont incorporé le même type de Bios. Par conséquent, toutes les cartes contrôleur de tous les fabricants et de toutes technologies (ST506/MFM, ST506/RLL et ESDDI) fonctionnent à partir des mêmes registres et des mêmes commandes.

Le premier essai de disque IDE date de 1986. Il est l'œuvre d'Imprimis (racheté depuis par Seagate) qui, à la demande de Compaq et avec l'aide de Western Digital, incorpore une carte contrôleur compatible PC/AT sur le disque dur. Le mariage contrôleur/disque permet de réduire le nombre de composants et, surtout, d'éliminer

une carte dans la machine. L'idée va petit à petit faire son chemin, Conner se montrant un précurseur dans la commercialisation de disques 3,5 pouces IDE.

Cependant, les différences d'interfaces et d'implémentations, ainsi que les incompatibilités requises entre disques entre eux freinent la diffusion des disques IDE.

Les sociétés impliquées dans l'IDE créent alors, en octobre 1988, un groupe de travail dénommé le CAM (Common Access Method) Committee, pour «standardiser» l'IDE sous le nom ATA (AT Attachment). Un document de description de l'IDE est élaboré au cours des mois suivants.

Après plusieurs révisions, il devient même une proposition de norme auprès de l'Ansi (groupe X319.2). Les bénéfices de cette standardisation se font sentir dès 1990 et conduisent même les fabricants de disques à stopper tous leurs développements en technologies ST506, MFM, ST506/RLL et ESDI. Seul le SCSI, interface intelligente et performante permettant d'accéder à des disques de très grande capacité, subsistera face au raz-de-marée IDE.

Fig. 1 : exemple d'utilisation de HDID sur une machine équipée d'un Quantum

```

C:\>HDID C:
HDID : Affichage des paramètres d'un disque dur IDE - Version JFDac 1.00 - Utilitaire écrit
pour la revue Langages & Systèmes numéro 12 (mai 1993) - Copyright (C) 1992-93 Jean-
François Duvivier - Langages & Systèmes - IDG

Modèle du disque           : QUANTUM LP105A 910109105
Caractéristiques Cyl x Têtes x Sec  : 755 x 16 x 17
Numéro de série            : 009144281
Nombre d'octets par secteur (piète)  : 512 - 8704
Revision du firmware       : 2.2
Constante de configuration  : 025A 00000010010110101
Taille et type buffer      : 128 x 512 - type 3
Nombre d'octets ECC        :
Transfert multi-secteur    : Non
Transfert double mot       : Non
Support du DMA             : Non

```

Les commandes disponibles

Un disque IDE réagit à des commandes qui lui sont adressées. Il en existe trois types :

- Celles qui doivent obligatoirement être implémentées par le disque. Elles correspondent en fait aux commandes reconnues par la première carte contrôleur utilisée sur le premier IBM AT. Ces commandes sont et resteront, dans les Bios des machines à base de processeurs 80286 (et supérieurs) les utilisent pour réaliser toutes les fonctions de l'interruption 13h standard.

La technologie IDE/AT, grâce à sa facilité d'utilisation, sa fiabilité et ses performances, a supplanté celles employées jusqu'alors (ST506/MFM, ST506/RLL et ESDI). Les disques IDE (Integrated Drive Electronics) disposent d'un contrôleur intégré et de commandes supplémentaires. L'une d'entre elles permet d'extraire certains paramètres du disque. Nous allons l'utiliser dans un programme qui affiche ces paramètres.

- Celles qui sont spécifiques à chaque fabricant de disque dur, et qui ne sont ni documentées ni même diffusées. Il peut s'agir de commandes utilisées pour les tests lors de la fabrication du disque, mais aussi de commandes permettant une configuration «fine» des tampons internes du disque ou une modification du firmware (dont une partie peut être en PROM et une autre stockée sur le disque lui-même en un endroit invisible pour l'utilisateur, permettant ainsi une sorte de téléchargement).

- Celles qui sont optionnelles et dont l'implémentation est laissée au bon vouloir des fabricants de disques. Elles sont cependant très clairement définies, afin qu'elles présentent les mêmes codes et fonctionnalités sur tous les disques. On y retrouve principalement trois grandes familles.

La première regroupe des commandes permettant de modifier la consommation du disque (Idle, Sleep, Standby) grâce à l'extinction graduelle de certaines parties (électronique, moteur...). La majorité des disques incorporés dans les blocs-notes utilisent ces fonctionnalités pour prolonger l'autonomie des batteries.

La seconde rassemble des commandes autorisant la lecture ou l'écriture de plusieurs secteurs à la fois (opération plus performante puisqu'une seule commande est envoyée au disque pour un groupe de secteurs consécutifs au lieu d'une commande pour chaque secteur), ainsi que des commandes permettant la lecture et l'écriture d'un secteur en utilisant un canal DMA. Cependant, comme les Bios n'utilisent pas ces commandes en raison de la sacro-sainte «compatibilité», la plupart des disques ne les implémentent pas.

La commande «Identify drive»

La dernière famille ne comprend qu'une seule commande, dénommée «Identify drive», qui permet de récupérer certaines informations sur le disque. Bien qu'optionnelle, cette commande est en fait toujours

implémentée et permet de «lire» un secteur de 512 octets dont le format est donné au tableau 1. Les informations sont regroupées par mots de 16 bits (256 au total). Certaines d'entre elles seulement intéresseront l'utilisateur :

- Le modèle du disque. Cette information retourne généralement la marque et le code attribué par le fabricant au disque. Elle permet donc d'identifier très rapidement et sans ouvrir la machine le modèle de disque dur installé.

- Le numéro de série du disque. En pratique, certains disques ne présentent pas cette information car le fabricant n'a pas prévu d'inscrire une table variable. Le numéro de série est en effet la seule différence pouvant exister entre deux disques de même modèle.

- La version du firmware, c'est-à-dire du logiciel implémenté sur le disque. Bien qu'il soit beaucoup moins sujet à modification et à correction que le Bios d'une machine, par exemple, le firmware d'un disque peut être l'objet de modifications, surtout au début de sa commercialisation.

- Les caractéristiques du disque, représentées sous la forme classique des cylindres, de têtes et de secteurs par piste. En général, les paramètres rendus sont «logiques», c'est-à-dire qu'ils définissent la façon dont on peut accéder au disque et non ses caractéristiques physiques réelles. Le transcodage entre paramètres physiques et logiques était nécessaire au départ (en 1988-1989) afin de présenter 17 secteurs par pistes, valeur qui était classique pour les disques MFM, et que nécessitaient certains logiciels.

Cette raison a perdu de son importance maintenant, mais le transcodage est toujours nécessaire en raison des limitations originales du Bios qui interdisent de définir plus de 1 024 cylindres, alors que les progrès dans la densité magnétique ont permis de dépasser cette limite sur un disque de 3,5 pouces. De plus, le Bios n'intègre pas non plus la possibilité d'utiliser un disque présentant un nombre de secteurs variable selon les pistes (on peut effectivement mettre plus de secteurs sur les pistes

LES PARAMETRES DES DISQUES IDE/AT

Tableau 1 : Les paramètres de la fonction «Identify drive»

Mot	Signification	Mot	Signification
0	General configuration bit-significant information :	6	Number of sectors per track
15	0 reserved	7-9	Vendor unique
14	1=format speed tolerance gap required	10-19	Serial number (20 ASCII characters)
13	1=track offset option available	20	Tampon type
12	1=data strobe offset option available	21	Tampon size in 512 byte increments
11	1=rotational speed tolerance is > 0,5%	22	Number of ECC bytes for read/write long commands
10	1=disk transfer rate > 10 Mbps	23-26	Firmware revision (8 ASCII characters)
9	1=disk transfer rate > 5 Mbps but <= 10 Mbps	27-46	Model number (40 ASCII characters)
8	1=disk transfer rate <= 5 Mbps	47	0000h=Read/write multiple commands not implemented x =Number of sectors that can be transferred
7	0 reserved	48	0000h=cannot perform doubleword I/O 0001h=can perform doubleword I/O
6	1=fixed drive	49	Capabilities
5	1=spindle motor control option implemented	15-9 0=Reserved 8 1=DMA supported 7-0 Vendor unique	
4	1=head switch time > 15 µsec	50	Reserved
3	1=not MFM encoded	51	PIO data transfer cycle timing mode
2	1=soft sectored	52	DMA data transfer cycle timing mode
1	1=hard sectored	53-127	Reserved
0	0 reserved	128-159	Vendor unique
1	Number of fixed cylinders	160-255	Reserved
2	Reserved		
3	Number of heads		
4	Number of unformatted bytes per track		
5	Number of unformatted bytes per sector		

extérieures du plateau que sur celles situées plus au centre). Cette technique, appelée ZBR (Zone Bit Recording), est très largement pratiquée pour augmenter les capacités d'un disque. Certains disques, rares il est vrai, peuvent indiquer leurs vrais paramètres «physiques» avant transcodage. C'était notamment le cas de quelques disques Conner. On s'en aperçoit généralement par un nombre de cylindres supérieur à 1 024. Dans le cas du Quantum LPS105 qui nous a servi de test (voir figure 1), on peut affirmer que les paramètres rendus sont «logiques» pour deux raisons. D'une part, le faible nombre de 17 secteurs par piste ne reflète pas les densités longitudinales que les fabricants sont capables d'atteindre aujourd'hui. A titre d'exemple, la technologie RLL datant de 1987 permettait déjà d'obtenir 26 secteurs par piste. D'autre part, le nombre de têtes (16) supposerait 8 plateaux qu'il est assurément impossible de faire tenir dans un disque haut de 2,5 cm. En fait, le LPS105 dispose de 2 plateaux, donc de 4 têtes, avec 1219 cylindres et un nombre de secteurs par piste égal à 35 (à l'intérieur), 42 (au milieu) ou 49 (à l'extérieur).

Le taux de transfert interne au disque. Les valeurs proposées sont en fait des seuils (5 et 10 mégabits par seconde) et trois bits du mot de configuration sont utilisés pour indiquer les performances du disque par rapport à ces seuils. Bien entendu, ces taux de transfert ne sont pas ceux que l'on va retrouver sous DOS avec un utilitaire tel que Coretest, mais il existe une relation directe : un disque ayant un meilleur taux de transfert

interne sera aussi plus performant sous DOS.

La taille et le type du tampon interne au disque. La plupart des disques modernes sont équipés d'un tampon mémoire permettant des opérations de «look-ahead» (lecture anticipée des secteurs qui suivent celui qu'on vient de lire) ainsi que du cache classique (la lecture répétée d'un même secteur ne provoquera plus d'accès «physique» au disque à partir de la deuxième opération, puisque les données sont déjà présentes dans le tampon du disque). La taille est importante car elle permet d'effectuer un cache plus important ou de lire en avance une piste entière. Le type de cache peut prendre quatre valeurs différentes : 0 (tampon non spécifié), 1 (tampon monosecteur et monoaccès), 2 (tampon multisecteur, double accès, capable de transferts simultanés vers le disque et vers l'ordinateur) et 3 (identique à 2, avec en plus des possibilités de cache en lecture). Plus le chiffre représentant le type est élevé, plus le tampon est performant.

D'autres informations sont également disponibles, tels le nombre d'octets ECC (codes de correction d'erreur) ou les possibilités de transfert multisecteurs ou DMA.

L'écriture de l'utilitaire HDID qui affiche certains de ces paramètres a été réalisée en langage mixte (MSC et assembleur). La partie en C permet l'interprétation et l'affichage des données, tandis que l'assembleur est plus adapté pour gérer le dialogue avec le disque. L'interface entre les deux langages se fait au moyen d'une fonction

GetID qui admet en paramètre un pointeur vers le tampon de 512 octets préparé pour recevoir les données. GetID rend un entier de type Int servant de code d'erreur.

Le programme C est donc très simple. La définition du tampon se fait au moyen d'une structure qui doit impérativement occuper au moins 512 octets, et dont la composition est calquée sur le tableau 1. L'utilisation d'une structure permet un accès simplifié à chacun de ses éléments. L'appel à la fonction GetID doit transmettre l'adresse de cette structure au moyen de l'opérateur &. A noter que les trois chaînes de caractères ASCII rendues par le disque doivent être retraitées car les octets sont inversés deux à deux (octets bas et octet haut de chaque mot). C'est le but de la fonction swab. Il est aussi nécessaire d'y ajouter le 0 qui sert de terminateur de chaîne en langage C, car il ne figure pas sur les données rendues par le disque. La dernière partie du programme est consacrée uniquement à l'affichage.

Du côté assembleur, la fonction GetID est définie comme PUBLIC et précédée du caractère _ (underscore), ce qui est la convention pour les fonctions C. Afin d'accéder facilement aux paramètres d'une fonction C, il est d'usage de définir une structure (Sgetid ici) qui comprendra bien sûr le registre bp «pushé» dès le début du programme pour servir de pointeur de pile, mais aussi l'adresse de retour au C (un seul mot puisqu'on travaille en mode small), puis tous les paramètres de cette fonction dans l'ordre de leur apparition. Les registres BP/SI/DI ainsi que les registres de segments doivent être sauvegardés s'ils sont modifiés lors de l'exécution de la routine assembleur.

Tableau 2 : les registres d'un disque IDE

Adresse	Lecture	Ecriture
1F0h	Données	Données
1F1h	Erreur	Particularités
1F2h	Nombre de secteurs	Nombre de secteurs
1F3h	Numéro de secteur	Numéro de secteur
1F4h	Cylindre (octet bas)	Cylindre (octet bas)
1F5h	Cylindre (octet haut)	Cylindre (octet haut)
1F6h	Disque / Tête	Disque / Tête
1F7h	Statut	Commande

Vu du PC, le disque dur présente huit registres (voir tableau 2) situés de 1F0h à 1F7h. Certains présentent des fonctions différentes en lecture ou en écriture. HDID n'en utilise que trois : le registre de données (pour lire les paramètres rendus par le disque), le registre Disque/Tête pour définir le disque accédé (00h pour le premier disque, 10h pour le second) et le registre Statut/Commande pour envoyer la commande. En lec-

ture, le Statut permet de gérer l'état du disque et de contrôler les erreurs. La signification des différents bits est donnée dans le tableau 3.

Tableau 3 : le registre de statusa

Bit	Signification
7	Busy : le disque est occupé.
6	Drive ready : le disque est prêt à accepter une commande.
5	Drive write fault : erreur d'écriture.
4	Drive seek complete : indique que les têtes sont correctement positionnées au-dessus d'une piste.
3	Data request : le disque est prêt à transférer des données vers le PC.
2	Corrected data : malgré une erreur, les données ont pu être corrigées.
1	Index : positionné à chaque révolution du disque.
0	Error : une erreur s'est produite lors de l'opération. Le registre 1F1h permet d'obtenir plus de précisions sur cette erreur.

L'envoi d'une commande requiert d'abord d'autoriser l'IRQ 14 utilisée par le disque pour signifier qu'il est prêt à transférer les paramètres. Il faut ensuite s'assurer que le disque est prêt à accepter une commande en testant le statut, puis charger les registres correspondants. Lorsque le disque a exécuté la commande, il génère une interruption IRQ 14 automatiquement traitée par le Bios, qui met à jour certaines variables dans sa zone de données située dans le segment 40h. Il suffit de tester une de ces valeurs pour savoir si l'IRQ 14 a été générée. Dans l'affirmative, c'est le signal de la disponibilité des paramètres qui peuvent alors être lus à partir du registre de données.

Le programme doit prendre bien soin, lors de toutes ses opérations, de maintenir à jour les variables du Bios (en particulier la copie du status en 40:74h) afin de ne pas perturber le fonctionnement ultérieur de la machine. Le programme C est compilé en mode small avec un MSC 6.0. Le programme assembleur est assemblé avec un MASM 6.0. Les syntaxes suivantes ont été utilisées :

```
masm /MX hdida,hdida;  
cl hdid.c hdida.obj
```

Parmi les améliorations qui pourraient être apportées à la version 1.00 de HDID, figurent le paramétrage du disque (0 ou 1), la sortie complète des 512 octets sous forme hexadécimale ou l'interprétation plus fine des paramètres codés sous forme binaire.

Bibliographie :

- ✓ Common Access Method AT Attachment (draft proposal).
- ✓ The programmer's PC Sourcebook, Thom Hogan (Microsoft Press).

LES PARAMETRES DES DISQUES IDE/AT

```

/*****
  HDID : Affichage des paramètres
  d'un disque IDE
  *
  COPYRIGHT (C) 1993 Jean-François
  DUVIVIER
  COPYRIGHT (C) 1993 Langages et
  Systèmes / IDG
  *****/
#include <stdio.h>
#include <stdlib.h>
int GetID (char *);

main ()
{
  struct ideparm {
    unsigned int  config,nbcyl,reserved1;
    unsigned int  nbheads,nbtr,nbse,
    nbse,reserved2[3];
    char          iserial[20];
    unsigned int  btype,bsize,necc;
    char          ifirmware[8],imodel[40];
    unsigned      intnbsecint,doublew,
    capa,reserved[255-49];
  } ide;
  char          serial[21],firmware[9],
  model[41];
  unsigned int  i,j;
  printf("HDID : Affichage des paramètres
  d'un disque dur IDE - "
  "Version JFDac 1.00\n"
  "Utilitaire écrit pour la revue
  Langages & Systèmes "
  "numéro 12 (mai 1993)\n"
  "Copyright (C) 1992-93 Jean-
  François Duvivier - "
  "Langages & Systèmes - IDG\n");
  if (GetID((char *)&ide)!=0)
  {printf("Erreur d'accès au disque
  !\n");exit(1);}
  swab(ide.iserial,serial,20);serial[20]=0;
  swab(ide.ifirmware,firmware,8);firmware[8]
  =0;
  swab(ide.imodel,model,40);model[40]=0;
  printf("Modèle du disque
  : %s\n",model);
  printf("Caractéristiques Cyl x Têtes x Sec
  : %d x %d x %d\n",
  ide.nbcyl,ide.nbheads,ide.nbse);
  printf("Numéro de série
  : %s\n",serial);
  printf("Nombre d'octets par secteur et
  piste : %d - %d\n",
  ide.nbbse,ide.nbbtr);
  printf("Révision du firmware
  : %s\n",firmware);
  printf("Constante de configuration
  : %04X (%d,ide.config);
  for (i=0,j=ide.config;i<16;i++)
  {printf((j>32767)?"1":"0");j=j<1;}
  printf("\n");
  printf("Taille et type buffer
  : %d x 512 - type %d\n",
  ide.bsize,ide.btype);
  printf("Nombre d'octets ECC
  : %d\n",ide.necc);
  printf("Transfert multi-secteur
  : %s ",ide.nbsecint?"Oui":"Non");
  if (ide.nbsecint) printf("(%dsecteurs
  par interruption)",ide.nbsecint);

```

```

  printf("\n");
  printf("Transfert double mot
  : %s\n",ide.doublew?"Oui":"Non");
  printf("Support du DMA
  : %s\n",(ide.capa&&256)?"Oui":"Non");
}
/*****
  HDIDA : Sous-programme assembleur
  pour l'utilitaire HDID *
  COPYRIGHT (C) 1993 Jean-François
  DUVIVIER
  COPYRIGHT (C) 1993 Langages et
  Systèmes / IDG
  *****/
PUBLIC _GetID

_text segment byte public 'CODE'
assume cs:_text,ds:_text

.286

;**** Définition des données et des
; équivalences
hd_status1 equ byte ptr ds:74h
hd_status2 equ byte ptr ds:8Ch
hd_flags equ byte ptr ds:8Eh
base equ 1F0h

sgetid struc ;structure définie pour
bp0 dw ? ;permettre un accès
adr0 dw ? ;plus facile des
adrbuf dw ? ;paramètres dans la
sgetid ends ;pile
getid equ [bp-bp0]

;-----
_GetID proc

  push bp ;récupération pointeur
  mov bp,sp ;de pile
  push es ;sauvegarde en entrée
  push ds ;des registres
  ;sensibles
  push si ;pour le MSC (segments
  push di ;et si/di

  push ds ;préparation de ES:DI
  pop es ;es=segment data du
  ;MSC
  mov di,getid.adrbuf

  mov ax,40h
  mov ds,ax
  call sblock
  jnz fin
  call wint ;attente des
  ;paramètres et
  jnz fin ;contrôle du timeout
  mov cx,512/2 ;transfert d'un
  ;secteur
  mov dx,base ;(512 octets soit
  ;256 mots)
  cld ;à partir du registre
  cli ;data vers le buffer
  rep insw ;défini par ES:DI.
  sti

fin: mov al,hd_status1 ;récupération
;de l'erreur
xor ah,ah ;éventuelle.
pop di ;restauration des

```

```

;registres
pop si ;sauvegardés en entrée
pop ds
pop es
pop bp
ret ;retour au MSC
;-----
sblock:
mov hd_flags,0 ;RAZ des flags
irq14: cli ;arrêt des interruptions
in al,0A1h ;lecture du masque
jmp short $+2 ;sur le second PIC
;8259 (esclave)
jmp short $+2
and al,10111111b ;autorisation
;IRQ 14
out 0A1h,al ;(IRQ6 du PIC2
;esclave)
jmp short $+2
jmp short $+2
in al,21h ;lecture du
;masque sur le
;premier PIC
jmp short $+2 ;8259 (maître)
jmp short $+2
and al,11111011b ;autorisation
;IRQ 2
out 21h,al ;(IRQ de renvoi
;entre le PIC
;esclave et le
;PIC maître)
jmp short $+2
jmp short $+2
sti

mov dx,base+7 ;test registre
;de status
xor cx,cx ;on fait 65536
;essais
@@: in al,dx ;lecture du
;status
mov hd_status2,al ;et sauvegarde
;de sa valeur.
xor al,01011000b ;inversion de
;certains bits
test al,11110000b ;test des quatre
;premiers
loopnz @B ;et boucle si
;jz ready ;nécessaire.
;après tous ces
;essais,
mov hd_status1,20h ;"General
;controller
;failure" et
;retour direct.
ret

ready: mov hd_status1,0 ;pas d'erreur !
mov dx,base+6 ;adresse registre
;Drive/Head
mov al,00h ;10h pour le
;deuxième disque
out dx,al
inc dx
mov al,0ECh ;commande
out dx,al ;"Identify drive"

xor al,al ;force le flag Z
ret ;et retour.
;-----
wint: mov ax,300 ;grande boucle
;de 300 fois

```

```

xor cx,cx ;la petite
;boucle de 65536
;fois !
@@: test hd_flags,80h ;permet de
;tester si l'IRQ
;14 du disque dur
;a été détectée
;par le BIOS.

loopz @B
jnz wok
dec ax
jnz @B
mov al,80h ;erreur "Timeout"
;après tous
;ces essais
wok: jmp short wend ;Remise à zéro
;de hd_flags
mov hd_flags,al
wend: mov hd_status1,al ; Mise à jour du
;status
or al,al ; Mise à jour du
;flag Z
ret ;et retour.

_GetID ENDP
_text ends ;fin du programme
end

```

NOUVEAU

TESTEZ VOS

LOGICIELS

AVANT DE LES

ACHETER

3616

INFOPC