

## EXPERIMENTER SUR APPLE II

Il est peu de laboratoires de psychologie qui ne disposent d'au moins un Apple II. Cet état de fait trouve une part de ses raisons dans la prolifération des logiciels adaptés à ce matériel et utiles à tout chercheur, qu'ils soient destinés au traitement de texte, à la gestion de fichiers, ou à la réalisation de graphiques. La circulation libre (et en toute légalité !) au sein de la communauté psychologique de programmes d'analyses statistiques relativement polyvalents, concernant par exemple l'analyse des comparaisons (e.g. Perruchet, 1982) et ses prolongements fiduciaires (Kerguelen, 1983), ou différentes formes d'analyses corrélationnelles et factorielles (e.g. De Lagarde, 1983) peut également se révéler comme un avantage décisif.

Mais l'Apple II est également choisi, par certains, pour ses qualités propres en tant qu'outil d'investigation expérimentale. Nous esquisserons en conclusion une évaluation de ce choix, mais notre objectif essentiel n'est pas là. Cet article est né d'un double constat. D'une part, les ingénieurs et techniciens tendent souvent à négliger, pour ne pas dire mépriser, cet instrument dont ils confinent volontiers l'usage à des fins ludiques, et préfèrent exercer leurs compétences sur du matériel informatique plus performant, ou bénéficiant au moins d'une connotation plus "professionnelle". Or, d'autre part, l'exploitation de l'Apple dans un cadre expérimental n'a pas toujours la simplicité désirable. Ainsi le recours au langage machine est presque toujours nécessaire, et là où un langage évolué suffit, les opérations requises ne sont pas toujours présentées dans les manuels de base. Cette situation conduit parfois le chercheur, par manque de temps ou de formation, à ne pouvoir exploiter au mieux son matériel. L'objectif de cet article est de contribuer à pallier cette difficulté, par le partage d'éléments de connaissances conférés par 5 années d'expérimentation sur l'Apple. Il convient d'ajouter que les indications fournies n'ont pas de prétention à l'originalité, encore moins à l'exhaustivité.

Les deux premières sections seront consacrées à l'usage expérimental des dispositifs d'entrée/sortie primaires de l'Apple : l'écran vidéo et le clavier. La 3e section est relative aux autres dispositifs d'entrée/sortie généralement nécessaires pour la conduite des expériences. Enfin, la dernière section concerne les modalités de mesure du temps de réaction.

## I - LA PRESENTATION DES STIMULUS SUR ECRAN

L'Apple peut-il remplacer un tachistoscope pour la présentation de stimulus visuels ? Dans quelle mesure peut-il, de ce dernier, égaler la qualité et la richesse des représentations ? De combien de "canaux" peut-on disposer, et à quel rythme peuvent-ils alterner ?

### 1) La génération du texte et des figures

Toutes ces questions, et d'autres semblables, ne peuvent recevoir de réponses univoques, car tout dépend des contraintes précises du plan expérimental.

La première éventualité correspond au cas où le chercheur peut se contenter d'un graphisme sommaire, caractérisé par une définition d'écran de 40\*48, ou, s'il s'agit de stimulus verbaux, des caractères standards de l'Apple. Il est alors possible, moyennant éventuellement certaines manipulations (e.g. Cavanagh et Anstis, 1980), d'alterner à un rythme rapide un nombre virtuellement infini de stimulus différents. Mais la pauvreté des éléments présentés font que cette solution est rarement jugée satisfaisante.

Dans la majorité des cas, l'utilisateur aura recours à la définition maximale du graphisme (mode "haute résolution"), qui est sur l'Apple de 280\*192 (ou 560\*192 avec une carte additionnelle 80 colonnes, en noir et blanc). Il convient de noter que cette définition est très loin d'égaliser celle d'un dessin à l'encre ou d'une image photographique. Les quelques exemples présentés en figure 1 témoignent de l'effet produit.

Les graphiques haute résolution peuvent être engendrés par différentes méthodes. Pour qui en a la patience, il est évidemment possible de générer un à un les 53760 points de l'écran. Il existe

heureusement d'autres procédures. On peut schématiquement distinguer trois possibilités.

La première est de faire usage de l'instruction HPLOT dans un programme BASIC. Il est très facile, par ce moyen, d'engendrer des droites, ou différentes figures réductibles à une formule algébrique, telles que cercles, ellipses, spirales, sinusoides, etc. (cf. figure 1.A). Plutôt que de mémoriser les figures produites, il est souvent plus économique d'inclure le sous-programme qui les engendre dans le programme d'expérience.

La seconde possibilité consiste à créer un dessin par l'intermédiaire d'un périphérique adéquat, tel que le tableau graphique Apple ou le "Koala pad", ou plus simplement à l'aide d'un logiciel spécialement adapté à cette tâche, tel que l'"illustrator" (Island Company). Entre autres commodités, cette procédure permet de remplir les fonds par un nombre pratiquement illimité de trames différentes (cf. figure 1.B). Dans un autre esprit, le logiciel "Appleworld" permet, par l'entrée de coordonnées numériques, de représenter un objet tridimensionnel et d'en varier, par exemple, l'angle de présentation ou l'échelle. L'utilisation ultérieure des dessins ainsi réalisés implique la sauvegarde de l'ensemble de la page haute résolution, soit de 8192 octets. Il ne sera possible d'introduire en mémoire vive que 2, ou au maximum 3 représentations différentes. L'usage d'un nombre plus important de représentations au sein d'une même expérience est évidemment possible, mais exige la lecture sur disque des images avant leur affichage. Cette opération prend environ 10 secondes (avec le DOS 3.3), un temps qui sera souvent jugé inconciliable avec les impératifs du plan expérimental.

La troisième possibilité est de construire des "formes" ("shapes"). Les formes, en Applesoft, sont des unités figurales qui, une fois créées, peuvent être à volonté rappelées par un simple numéro, et dont la position, l'orientation, et l'échelle peuvent être changées.

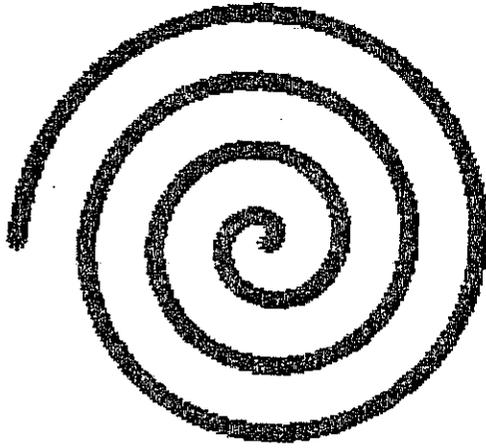
La méthode de construction des formes figurant dans le manuel de l'applesoft est longue et fastidieuse. Il existe heureusement des logiciels spécialement conçus pour faciliter la tâche du programmeur. L'un de ceux-ci est proposé dans le n°4 de la revue POM'S. Il permet la réalisation et la correction de formes s'inscrivant dans un carré de 24\*24 pixels au maximum ; les formes plus importantes peuvent être

établies par changement d'échelles, ou par juxtaposition de formes plus petites. "Pixit", et "E-Z Draw" sont d'autres logiciels, beaucoup plus élaborés, dans lesquels la taille des formes réalisables n'a pas d'autres limitations que la définition de l'écran.

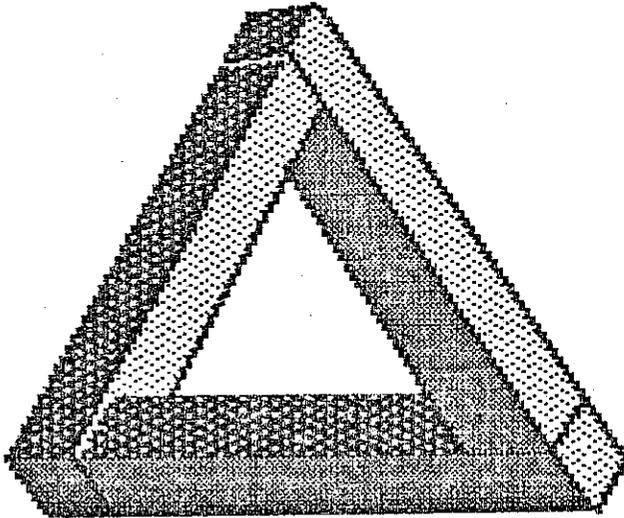
Il faut noter que, quel que soit le logiciel utilisé, le remplissage des fonds n'est pas aisé ; on aura donc intérêt à réserver cette modalité d'élaboration à des dessins qui, tout en restant éventuellement figuratifs et relativement complexes, peuvent se limiter à la représentation des contours et des traits. La représentation de caractères, lettres ou chiffres, différents en taille et/ou en forme des caractères standards de l'Apple, entrent également dans cette catégorie. Des alphabets variés accompagnent la plupart des logiciels permettant la réalisation des formes (cf. figure 1.D). Mais la réalisation d'une police originale de caractères ne prend que quelques dizaines de minutes, si du moins l'on dispose de modèles adaptés à une figuration point par point (support de broderie au point de croix, par exemple).

Le stockage des formes en mémoire est très économique, ce qui permet d'en garder en mémoire vive un nombre important (de quelques dizaines jusqu'à 256, en fonction de leur complexité et de la place disponible en RAM).

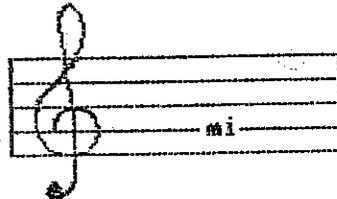
Il faut ajouter qu'il est tout à fait possible de mélanger les différentes modalités d'exécution que nous avons distinguées. Supposons par exemple une expérience requérant une image complexe, avec remplissage des fonds, au sein de laquelle seuls quelques éléments varient d'essais en essais. La partie commune peut être réalisée en utilisant les commodités d'un logiciel comme l'"illustrator", et stockée par enregistrement de la totalité de la page graphique, et les parties variables peuvent être réalisées comme des "formes", facilement manipulables. On peut remarquer incidemment que le "fond commun", aussi chargé soit-il, ne diminue en rien l'espace mémoire laissé disponible pour les formes, le programme, ou les données, puisqu'il est possible de le placer directement dans l'espace mémoire réservé au graphisme haute résolution.



A La spirale a été engendrée par le déplacement d'une petite "forme" ronde (pour donner de l'épaisseur au trait) à partir d'un programme BASIC calculant les coordonnées de chaque point.



B Cet "objet impossible" a été réalisé en 17 min. avec un logiciel de dessin (Illustrator). Sa sauvegarde exige 8 Koctets de mémoire.



C Ce genre de stimulus est utilisé dans une expérience en cours pour la mise en évidence d'un "effet stroop" dans la lecture musicale. La clef et le nom de la note sont des formes; la portée est dessinée par HPLLOT. Seuls quelques dizaines d'octets sont nécessaires pour conserver des figures de cette nature.

informatique et sciences humaines  
 Informatique et sciences humaines  
**Informatique et sciences**  
*Informatique et sciences humaines*  
**INFORMATIQUE ET SC**

D Chaque caractère est une forme. Des caractères de ce genre sont disponibles dans la plupart des logiciels de génération de formes (ceux-ci proviennent de "Pixit"). Leur utilisation dans un cadre expérimental n'est pas très compliquée, mais reste moins directe que l'utilisation des caractères standards de l'Apple.

FIGURE 1: Exemples de copies d'écran graphique (définition: 280\*192)

## 2) Le contrôle temporel de l'affichage

Les problèmes posés par l'affichage des représentations sur écran vidéo sont en partie indépendants du mode d'élaboration de ces dernières.

La procédure standard consiste à afficher une page vide (par TEXT, GR, ou HGR, selon que l'on désire représenter du texte, un graphique basse résolution, ou une page graphique haute résolution), puis à remplir cette page en temps réel par les instructions appropriées (PRINT, PLOT, HPLOT, DRAW, etc.). Cette procédure, dans le cas général, se révèle inadaptée à l'expérimentation, car elle n'assure pas une apparition immédiate des représentations. Il est toutefois possible de préparer une page à afficher avant sa présentation effective. Cette possibilité est d'autant plus précieuse qu'il existe sur l'Apple deux pages graphiques haute résolution, dont il est ainsi possible d'alterner l'affichage. Lorsque la haute résolution est en place (par HGR, HGR2, ou par POKE), "POKE 49236,0:POKE 230,64" affiche la p.1, et toutes les instructions graphiques (HPLOT, DRAW, SCALE...) modifient la p.2, et "POKE 49237,0:POKE 230,32" inverse ces rôles.

L'alternance systématique des pages graphiques avec ces instructions apparaît adaptée à un très large éventail de situations expérimentales, et notamment au cas où un point de fixation alterne avec la présentation de différents stimulus. Il est de la même façon possible d'alterner avec d'autres pages, et en particulier avec la page texte, lorsque les caractères ordinaires de l'Apple sont jugés suffisants pour la présentation des stimulus verbaux. Les instructions pertinentes sont présentées dans l'annexe F3 du manuel de programmation.

Si, perceptivement, l'affichage d'une représentation, aussi complexe soit-elle, apparaît le plus souvent comme immédiat lorsque la procédure précédente est utilisée, on peut souhaiter un contrôle temporel de l'exposition encore plus précis. Le balayage de l'écran, qui dure environ 12 millisecondes, n'est initié, comme dans tout écran TV, que toutes les 20 millisecondes. Un ordre d'affichage "lancé au hasard" n'est donc pas assuré d'être suivi par l'affichage effectif et complet de la représentation. Cette imprécision peut être levée. Il est possible d'accéder à certains signaux de synchronisation du signal vidéo. Ceci exigeait, sur les anciens modèles d'Apple II, une petite modification de

circuit, décrite par Cavanagh et Anstis (1980). Sur l'Apple IIe, une simple lecture à l'octet 49177 (\$C019) est suffisante : le bit de poids fort est à 1 pendant les balayages, et à 0 entre deux balayages, pendant l'intervalle de suppression verticale. Il est possible de modifier la page à afficher pendant cet intervalle, et éventuellement, de synchroniser le départ d'un compteur de temps de réaction avec le début de la présentation des stimulus. Le recours au langage-machine est évidemment nécessaire pour travailler à ce niveau de précision. Un sous-programme alternant les pages 1 et 2 pendant l'intervalle de suppression verticale, et renvoyant au programme principal en début de balayage est proposé au tableau 1.

L'utilisation de cette procédure se révèle particulièrement nécessaire lorsque l'on désire présenter des informations pendant des temps très courts. Le nombre de balayages peut être compté par l'incrémentation d'un registre à chaque initiation d'un rafraichissement, et la page à afficher changée lorsque ce nombre atteint une valeur fixée. L'écran standard de l'Apple IIe autorise ce genre de manipulations sous d'assez bonnes conditions, en raison d'un taux de rémanence acceptable : un signal vidéo d'intensité moyenne est réduit à 10% de sa valeur en 1,5 msec. environ, d'après les mesures effectuées par C. et P. Beauvillain sur un oscilloscope connecté à un photorécepteur.

10	REM	VALEUR DECIMALE DU S-P			
20	DATA	169,127,205,25,192,48,251,141			
22	DATA	84,192,205,25,192,16,251,96			
24	DATA	169,127,205,25,192,48,251,141	0350-	A9 7F	LDA £\$7F
26	DATA	85,192,205,25,192,16,251,96	0352-	CD 19 C0	CMP \$C019
50	REM	CHARGEMENT DU S-P	0355-	30 FB	BMI \$0352
55	FOR	I = 0 TO 31	0357-	8D 54 C0	STA \$C054
58	READ	V: POKE 848 + I,V	035A-	CD 19 C0	CMP \$C019
60	NEXT	I	035D-	10 FB	BPL \$035A
70	REM	ALTERNE PAGES HGR 1 ET 2	035F-	60	RTS
71	REM	(SANS LES DETRUIRE)	0360-	A9 7F	LDA £\$7F
72	REM	ENTRE 2 BALAYAGES D'ECRAN	0362-	CD 19 C0	CMP \$C019
73	REM	*****	0365-	30 FB	BMI \$0362
90	POKE	49232,0: POKE 49239,0	0367-	8D 55 C0	STA \$C055
100	P1 =	848:P2 = 864	036A-	CD 19 C0	CMP \$C019
110	CALL	P1: REM AFFICHE P.1	036D-	10 FB	BPL \$036A
120	FOR	I = 1 TO 2000: NEXT	036F-	60	RTS
130	CALL	P2: REM AFFICHE P.2			
135	FOR	I = 1 TO 2000: NEXT			
140	GOTO	110			

TABLEAU 1 : Changement d'affichage entre 2 balayages

## II - LA SAISIE D'INFORMATIONS PAR LE CLAVIER

Compte tenu de la familiarité généralement réduite des sujets vis-à-vis d'un clavier d'ordinateur, la saisie directe des réponses du sujet par cet intermédiaire est certainement un cas d'exception. Mais l'expérimentateur lui-même peut désirer transférer à l'ordinateur, pendant le décours de l'expérience, un certain nombre d'informations : codage des réponses vocales ou du comportement général du sujet, ou signal d'événements perturbateurs, par exemple. Il peut également souhaiter se réserver la possibilité d'interrompre provisoirement l'expérience au moment de son choix. Dans tous les cas, le clavier se présente comme le moyen d'intervention le plus naturel. Mais généralement, l'inclusion à cette fin, dans le programme, des instructions d'entrée normales GET ou INPUT n'est pas satisfaisante, car elle interrompt le déroulement du programme jusqu'à ce qu'une touche soit appuyée. Entre autres inconvénients, cette interruption rend difficile le contrôle des intervalles temporels.

Au moins trois autres procédures sont utilisables.

Si le moment précis de l'intervention n'est pas déterminant, il est possible de lire, à des intervalles choisis (à la fin de chaque essai par exemple), l'octet 49152, qui contient en permanence le code ASCII de la dernière touche appuyée. Il est souhaitable de faire suivre cette lecture par un PEEK (49168), afin de réinitialiser l'octet 49152, ce qui permet de savoir à chaque lecture si une touche a été enfoncée depuis la lecture précédente (en fait, seul le bit de poids fort est mis à zéro ; le code ASCII demeure accessible, mais dans la rangée 0-127, alors qu'il est sinon dans la rangée 128-255).

Si l'intervention au clavier doit être précisément datée, on peut introduire dans une boucle la lecture de l'octet 49168 : si la valeur obtenue est égale ou supérieure à 128, une touche est enfoncée, et le code ASCII de cette touche peut à nouveau être lu à l'octet 49152.

Enfin, une procédure plus délicate, mais qui peut se révéler nécessaire si l'intervention doit être datée sans qu'une interrogation en boucle des données du clavier soit réalisable, consiste à interrompre provisoirement par Ctrl-Reset le déroulement normal du programme, après avoir remplacé par un programme utilisateur le sous-programme de réinitialisation auquel Reset normalement renvoie (Cf. manuel de référence, p.86).

Ces 3 procédures ont l'avantage, par rapport à la procédure d'entrée standard, de rendre l'intervention au clavier facultative, et sans conséquence néfaste sur le contrôle chronologique des événements. Additionnellement, elles n'entraînent pas de modification de l'affichage vidéo, ce qui peut se révéler précieux quand l'écran sert de support au stimulus.

### III - LES AUTRES DISPOSITIFS D'ENTREE/SORTIE

La présentation des stimulus à l'écran n'est pas toujours possible ou souhaitable ; elle exige parfois l'intermédiaire de différents dispositifs : magnétophones, générateur de fréquence, projecteur, moteur, etc. L'asservissement de ces dispositifs par l'ordinateur s'opère par des sorties logiques, ou éventuellement analogiques. De même, et plus encore, l'usage du clavier pour la saisie des réponses est rarement suffisant, et

un appareillage plus ou moins sophistiqué est souvent requis : bouton-poussoir, clef vocale, électrodes, et capteurs divers. Les signaux issus de ces capteurs sont intégrés par des entrées logiques et analogiques.

Le contrôle des entrées/sorties peut s'opérer par l'intermédiaire de cartes périphériques adaptées. Il n'y a évidemment aucun problème à loger une ou plusieurs de ces cartes, puisque 5 connecteurs d'extension restent disponibles quand l'unité centrale a reçu les périphériques usuels (moniteur, disques, et imprimante), et le choix est large. En France, MID (96 Bld Richard-Lenoir, 75011 Paris), et Progetec Informatique (8 ave de Grande Bretagne, 66000 Perpignan) fabriquent et commercialisent des cartes variées, cumulant généralement plusieurs fonctions. Mais on trouve également une quantité d'autres cartes de fabrication américaine, telle que la carte Adalab, diffusée par Alpha systèmes, qui comporte en particulier des entrées/sorties logiques et analogiques et 3 timers. La gestion de ces cartes leur est spécifique.

Mais avant l'achat d'une carte périphérique de ce genre, les possibilités du "connecteur de jeux" doivent être évaluées. Le connecteur interne à 16 broches, en haut et à droite de la carte-mère de tous les modèles d'Apple II, comporte en effet 4 sorties logiques, 3 entrées logiques et 4 entrées analogiques, qui sont souvent en mesure de satisfaire les besoins d'un laboratoire de psychologie. L'intérêt du connecteur de jeux, outre l'économie qu'il permet éventuellement de réaliser sur le plan financier, est que la gestion des entrées/sorties qu'il supporte est extrêmement simple : si elle peut être effectuée en langage machine, elle est également possible en BASIC, par des instructions spécifiques présentées dans les manuels d'introduction de l'Apple.

Les entrées logiques peuvent être facilement reliées à des boutons-poussoirs (cf. p.171 du manuel de référence), et peuvent être utilisées chaque fois que les impératifs expérimentaux n'exigent pas plus de 3 réponses différentes en tout-ou-rien. L'utilisation des entrées analogiques est plus restrictive. En effet, les convertisseurs A/D de l'Apple n'opèrent que sur 8 bits (c'est-à-dire qu'ils transforment les variations analogiques sur une échelle numérique allant de 0 à 255), et leur durée de conversion est particulièrement longue. Deux conversions

successives ne peuvent être sollicitées à un intervalle inférieur à 3 ou 4 msec., sans encourir un risque d'erreur important.

#### IV - LA MESURE DES TEMPS DE REACTION (TR)

Le TR est la principale variable dépendante d'un grand nombre d'expériences. Son unité de mesure est généralement la milliseconde. Cette précision implique un recours obligatoire au langage machine. En effet, le temps d'exécution d'une instruction en BASIC interprété dépasse déjà souvent la msec. fatidique, et de plus, comme pour toute instruction d'un langage évolué, ce temps est difficilement contrôlable (e.g. Dlhopsky, 1983). Outre sa rapidité, le langage machine a l'avantage de permettre une estimation exacte de la durée d'exécution d'un programme : pour un micro-processeur donné, chaque instruction dure un certain nombre de cycles d'horloge (ce nombre n'est pas indiqué dans la documentation Apple, mais peut être consulté dans des ouvrages généraux sur le microprocesseur 6502 qui équipe l'Apple ; e.g. Zaks, 1979) et la durée d'un cycle d'horloge est elle aussi connue ; elle est sur l'Apple très légèrement inférieure à 1 microseconde. Par-delà ce recours nécessaire au langage machine, la mesure du TR peut s'opérer de différentes façons.

1) La méthode de mesure la plus traditionnelle consiste à interroger répétitivement l'état d'un circuit ou d'une mémoire, et à interrompre un compteur lors d'un changement d'état.

L'utilisateur peut faire usage, soit d'un compteur interne, soit d'une horloge externe. Dans le premier cas, le microprocesseur gère lui-même un compteur, en incrémentant une valeur toutes les millisecondes. Il lui est pratiquement impossible de remplir une autre tâche dans le même temps ; non pas que son travail l'occupe beaucoup : l'interrogation des boutons-poussoirs et l'incrémentation de la valeur ne lui prennent que quelques microsecondes, ce qui le laisse donc libre pendant près de 99% de son temps. Mais il lui faut mesurer ce temps, et pour ce faire, il ne peut guère qu'exécuter le même programme soigneusement étalonné. Dans le second cas, le microprocesseur ordonne le départ d'une horloge, sur laquelle il peut aller "lire l'heure" au moment voulu. En apparence, il sera libéré pour exécuter d'autres opérations.

Mais cette libération est en grande partie illusoire, car il devra toujours interroger l'état des boutons-poussoirs, au moins toutes les msec. Même s'il n'a pas à mesurer très précisément cet intervalle, on conviendra qu'il est difficile d'utiliser intelligemment un temps aussi morcelé. L'usage d'une horloge externe ne sera profitable que si un délai relativement long, de l'ordre de plusieurs secondes, s'interpose entre l'envoi du stimulus et le moment où l'interrogation des boutons-poussoirs devient nécessaire. Ce temps pourra être utilisé pour exécuter des opérations variées en langage évolué. Dans le cas général, où un tel délai n'existe pas, un horloge externe n'est pas d'un grand secours, et représente même une source de complication superflue.

Le sous-programme du tableau 2 mesure les TRs sans horloge externe. Il interroge les entrées logiques 0 et 1 de l'Apple (également accessibles par les touches pomme-ouverte et pomme-pleine sur l'Apple IIe), et peut donc être utilisé pour les mesures de TR simple et à deux choix.

10	REM	VALEUR DECIMALE DU S-P			
20	DATA	169,255,141,68,3,141,70,3	0300-	A9 64	LDA £\$64
22	DATA	169,250,141,69,3,141,71,3	0302-	8D 44 03	STA \$0344
24	DATA	162,255,160,250,169,127,205,97	0305-	8D 46 03	STA \$0346
26	DATA	192,48,27,205,98,192,48,29	0308-	A9 FA	LDA £\$FA
28	DATA	169,1,32,168,252,169,6,32	030A-	8D 45 03	STA \$0345
30	DATA	168,252,169,15,32,168,252,136	030D-	8D 47 03	STA \$0347
32	DATA	208,226,202,208,221,96,142,68	0310-	A2 64	LDX £\$64
34	DATA	3,140,69,3,96,142,70,3	0312-	A0 FA	LDY £\$FA
36	DATA	140,71,3,96	0314-	A9 7F	LDA £\$7F
40	REM	CHARGEMENT DU S-P	0316-	CD 61 C0	CMP \$C061
50	FOR I = 0 TO 67		0319-	30 1E	BMI \$0336
55	READ V: POKE 768 + I,V		031B-	CD 62 C0	CMP \$C062
60	NEXT I		031E-	30 1D	BMI \$033D
62	REM	PROGRAMME DE DEMONSTRATION	0320-	A9 01	LDA £\$01
63	REM	*****	0322-	20 A8 FC	JSR \$FCA8
65	HOME		0325-	A9 06	LDA £\$06
70	PRINT	"LE S-P DE MESURE DU TR REVIENT"	0327-	20 A8 FC	JSR \$FCA8
72	PRINT	"AU BASIC IMMEDIATEMENT APRES"	032A-	A9 0F	LDA £\$0F
74	PRINT	"LA REPONSE DU SUJET"	032C-	20 A8 FC	JSR \$FCA8
80	PRINT	"OU A DEFAULT"	032F-	88	DEY
85	PRINT	"APRES N * 250 MSEC.": PRINT	0330-	D0 E2	BNE \$0314
90	INPUT	"ENTRER N (1 < N < 255): ";N	0332-	CA	DEX
150	POKE	769,N: POKE 765,N	0333-	D0 DD	BNE \$0312
160	HOME		0335-	60	RTS
170	HTAB (9): PRINT	"X"	0336-	8E 44 03	STX \$0344
180	CALL	768	0339-	8C 45 03	STY \$0345
190	X = PEEK	(836):Y = PEEK (837)	033C-	60	RTS
200	T1 = ((N - X) * 250) + (250 - Y)		033D-	8E 46 03	STX \$0346
210	X = PEEK	(838):Y = PEEK (839)	0340-	8C 47 03	STY \$0347
220	T2 = ((N - X) * 250) + (250 - Y)		0343-	60	RTS
230	PRINT	T1,T2			
240	FOR I = 1 TO RND (1) * 2000 + 2000				
245	NEXT				
250	GOTO	170			

TABEAU 2 : Mesure des TRs simples et à 2 choix (en msec.)

2) L'interrogation répétée des boutons-poussoirs a l'inconvénient majeur de mobiliser pratiquement le microprocesseur pendant toute la période où le sujet est susceptible de répondre. La stratégie alternative est de provoquer "de l'extérieur" une interruption dans le déroulement d'un programme.

L'usage d'une horloge externe est ici indispensable pour donner sens à la procédure. Après avoir ordonné la mise en route de cette horloge, le programme peut exécuter toutes les opérations souhaitées, éventuellement en langage évolué, sans se soucier, ni de l'état des boutons-poussoirs, ni du temps écoulé. Son exécution sera interrompue par la réponse du sujet ; le microprocesseur est alors orienté vers un sous-programme rédigé par l'utilisateur avant la poursuite du programme principal. Il suffit pour mesurer le TR que ce sous programme inclue la lecture de l'horloge.

Les interruptions ne peuvent être provoquées par les entrées du connecteur de jeux. Elles sont par contre possible, sur l'Apple IIe, à partir de chacun des 7 connecteurs d'extension (Manuel de référence p.135). Elles impliquent donc l'achat (outre d'une carte horloge) d'une carte périphérique permettant cette opération, telle que par exemple la carte ES-16 CP, de MID, ou la carte PIA-2T, de Progetec. Il faut souligner qu'outre cet investissement financier, les interruptions exigent pour leur gestion un support logiciel plus sophistiqué que la méthode traditionnelle de scrutation des entrées, qui, dans bien des applications, se révèle suffisante.

## CONCLUSION

Notre objectif n'était pas de porter un jugement évaluatif. Il reste que, par nécessité, l'accent a été placé sur les capacités de l'Apple, ce qui peut apparaître à tort comme une défense inconditionnelle des qualités de ce matériel. Pour éviter toute ambiguïté, il est sans doute préférable de confronter directement l'Apple à ses concurrents.

L'Apple II est né en 1977. Bien qu'ayant bénéficié depuis d'améliorations successives, il garde certaines limitations inhérentes à l'ancienneté de sa structure. Certains micro-ordinateurs plus récents, également construits autour d'un microprocesseur 8 bits, offrent par exemple une résolution graphique un peu plus fine, ou un surcroît de performance sur tel ou tel aspect, pour un prix comparable et parfois inférieur.

Les limites de l'Apple apparaissent toutefois beaucoup plus nettement si on le compare (et les autres micro-ordinateurs 8 bits avec lui) aux micro-ordinateurs basés sur les microprocesseurs 16 bits (sans évoquer les microprocesseurs 32 bits) actuellement sur le marché. La supériorité de ces derniers est indiscutable.

Deux points méritent d'être soulignés.

Le premier est relatif à la vitesse d'exécution. Le fait d'opérer sur des "mots" de 16 bits divise le temps d'exécution des programmes par une valeur au moins égale à 7. Or le temps d'exécution est un paramètre critique pour la conduite des expériences, particulièrement lorsque l'analyse des réponses doit s'opérer en temps réel pour déterminer le déroulement ultérieur des événements. Certes, la vitesse de traitement peut être accélérée, sur l'Apple, par différents moyens. Le plus simple et le moins coûteux est de compiler les programmes, communément écrits en BASIC interprété. Il existe plusieurs compilateurs pour Applesoft, qui diffèrent sur un certain nombre de points : temps de compilation, durée d'exécution des programmes compilés, place occupée en mémoire, etc. Les programmes compilés avec TASC occupent une place relativement réduite (bien que restant supérieure d'environ 50 % à celle de la version interprétée), et permettent de diviser le temps d'exécution de programmes d'expérience par un facteur de 5 environ. Une autre possibilité plus onéreuse est d'adjoindre un co-processeur au 6502. Le choix est étendu.

Il est par exemple possible d'utiliser le 6502C, qui divise le temps d'exécution de tous les programmes par 3,5. Le co-processeur 68000 (un 16/32 bits) divise, selon la notice, le temps d'exécution de programmes BASIC par un facteur allant de 4 à 20. Ces solutions ne sont pas à négliger, mais ne peuvent manquer d'apparaître comme des prothèses inélégantes, face au choix initial d'un micro-ordinateur directement conçu autour d'un microprocesseur plus puissant.

Un second avantage décisif des micro-ordinateurs 16 bits est leur aptitude à gérer une mémoire d'une capacité considérable. Cette différence est elle aussi d'une importance fondamentale pour l'expérimentation. Il est souvent nécessaire de stocker en mémoire une quantité importante de données, et les 64 Koctets que les microprocesseurs 8 bits permettent d'adresser directement n'offrent pas toujours une place suffisante. De plus, l'extension de la mémoire permet d'en réserver une part importante à la représentation de l'écran, et donc d'améliorer considérablement la définition du graphisme. Là encore, l'Apple a quelques prothèses à sa disposition. Elles peuvent prendre par exemple la forme des cartes 80 colonnes étendues, qui comportent 64K additionnels. L'accès à cette mémoire est cependant malcommode, et son utilisation alourdit beaucoup la tâche du programmeur.

La supériorité évidente des nouveaux microprocesseurs 16 bits semblent condamner irrémédiablement leurs frères de 8 bits à disparaître des laboratoires. Si cette condamnation semble, à terme, inéluctable, il n'est pas certain toutefois qu'elle prenne effet dans un avenir immédiat. Les possibilités techniques ne représentent qu'un critère de choix.

Un autre critère est d'ordre financier : sauf exception, la différence de prix entre les 8 et les 16 bits reste encore tout à fait sensible.

Mais un autre élément, dont l'importance capitale a parfois été sous-estimée, est à prendre en compte ; il est relatif au degré de divulgation du produit. L'Apple a su recueillir, à un niveau international, une sorte de consensus chez les psychologues expérimentalistes, dont l'avantage est substantiel : il se manifeste par exemple dans les collaborations inter-laboratoires, ou à la lecture d'une revue comme Behavior Research Methods, Instruments, and Computers, qui consacre à l'Apple une part majoritaire de ses articles. La divulgation

parallèle à d'autres milieux professionnels et au grand public a d'autres conséquences bénéfiques, telles que l'abondance et le cout réduit des cartes périphériques et des logiciels d'utilité générale. Ces avantages maximisent les chances que l'acquéreur d'un Apple a de disposer, pour un investissement financier et temporel limité, d'un outil immédiatement opérationnel à toutes les étapes de la démarche expérimentale. A ma connaissance, aucun autre micro-ordinateur n'offre aujourd'hui une garantie comparable. Que dans une decennie, nombre d'Apple initialement affectés à l'expérimentation soient reconvertis à des taches "subalternes" telles que le traitement de texte ne saurait, néanmoins, surprendre.

Pierre PERRUCHET

Laboratoire de Psychologie différentielle  
 Université René Descartes, EPHE (3e section)  
 et CNAM (INETOP) CNRS UA 656  
 28, rue Serpente 75006 Paris

#### BIBLIOGRAPHIE

- CAVANAGH, P., ANSTIS, S.M.,  
 Anstis S.M. Visual psychophysics on the APPLE II : getting started. Behavior Research Methods & Instrumentation, 1980, 12, 614-626.
- De LAGARDE, J.,  
 Initiation à l'analyse des données. Dunod, Paris, 1983.
- DLHOPOLSKY, J.G.,  
 Limitation of high-level microcomputer languages in software designed for psychological experimentation. Behavior Research Methods & Instrumentation, 1983, 459-464.
- KERGUELEN, A.,  
 FIDAP. Programme d'inférence fiduciaire sur Apple II. Document photocopié, Laboratoire de Psychologie du travail de l'EPHE, 1983.

PERRUCHET, P.,

Programmes de description et d'analyses inférentielles de données expérimentales pour micro-ordinateur, Informatique et Sciences Humaines, 1982, 55, 87-101.

ZAKS, R.,

Programmation du 6502. Sybex, 1979.