

## **PILOTES D'INTERFACES DE MESURE, POUR LES LOGICIELS D'EXPÉRIMENTATION**

**Pierre DIEUMEGARD**

*Résumé : Pour l'expérimentation assistée par ordinateur (EXAO), par l'emploi de "pilotes d'interfaces", on peut écrire des logiciels polyvalents sans connaître le détail du fonctionnement des interfaces de mesure possibles. Inversement, avec une nouvelle interface, il suffit d'écrire un nouveau pilote pour que celle-ci soit immédiatement utilisable par les logiciels précédents. Les exemples proposés fonctionnent sous MS-DOS ou sous MS-Windows, et les programmes-sources sont en Turbo Pascal ou en Borland Pascal 7. L'utilisation de MS-Windows permet d'utiliser les librairies ainsi écrites dans les tableurs QuattroPro pour Windows et Excel.*

Depuis une dizaine d'années, l'EXAO (EXpérimentation Assistée par Ordinateur) se développe dans les lycées et collèges français, en biologie, géologie, physique, chimie technologie...

En sciences expérimentales, une chaîne de mesure informatisée comprend, outre l'ordinateur et le capteur de mesures, une interface, destinée à transformer une valeur électrique pouvant varier de façon continue (volts, en général) en une valeur informatique, codée sur quelques bits.

### **1 - LES PROBLÈMES DE LA DIVERSITÉ DES INTERFACES**

#### **1-1. La situation actuelle**

De nombreuses interfaces sont disponibles dans le commerce, ayant chacune leurs avantages et leurs inconvénients.

Certaines sont très répandues (Jeulin ESAO3), d'autres ont une grande précision (Leybold-Cassy), d'autres sont faciles à brancher (Orphy-GTS), d'autres sont bon marché (MEP-PMB),

Pour chacune de ces interfaces, les constructeurs peuvent fournir des logiciels destinées à les exploiter (prise de mesures en fonction du temps, commande d'appareillage, régulations...). Malheureusement, jusqu'ici, il n'y a guère de logiciels polyvalents, capables de fonctionner avec diverses interfaces.

En règle générale, chaque entreprise réalisant une interface de mesure réalise aussi les logiciels permettant d'exploiter cette interface. Cette situation ressemble un peu à la situation générale de la micro-informatique vers le début des années 80, où de nombreux constructeurs indépendants de micro-ordinateurs faisaient à la fois le matériel (peu performant) et le logiciel (peu puissant et peu agréable à utiliser), sans aucune compatibilité.

## **1-2. Inconvénients de la non-compatibilité entre les matériels et les logiciels**

Cette situation a plusieurs inconvénients :

- en amont, pour les fabricants d'interfaces : pour pouvoir mettre sur le marché une nouvelle interface, en plus des problèmes de construction électronique, ils doivent aussi réaliser les logiciels capables d'exploiter cette interface, ce qui renchérit le prix de l'ensemble.
- en aval, pour les utilisateurs, puisqu'ils sont prisonniers de leur système de mesure : ils ne peuvent pas exploiter des logiciels intéressants, mais fonctionnant avec une autre interface que la leur.
- et par dessus tout pour les enseignants, qui doivent faire des travaux pratiques avec du matériel hétérogène, donc des logiciels différents, ce qui rend les explications au groupe d'étudiants très complexes...

Imaginons une séance de TP dans un lycée qui a eu la chance d'être équipé de plusieurs postes d'EXAO, mais travaillant avec des interfaces différentes, et imaginons les instructions que le professeur doit donner à ses étudiants avant de commencer les manipulations :

"Avec le système Orphy, il faut valider telle option du menu ; avec Jeulin, il faut appuyer sur Alt-O, et avec Cassy, cliquez avec la souris sur l'icône portant le titre de la manipulation..."

Les lecteurs comprendront facilement que de nombreux enseignants renoncent à une telle "pédagogie personnalisée"...

Travaillant depuis plusieurs années à la réalisation de logiciels d'EXAO, j'ai été confronté au problème de la diversité des matériels, aussi bien dans le groupe de recherche académique d'Aix-Marseille que dans le groupe national IDEAO (Université Paul Sabatier, Toulouse), et je propose ici une méthode pour le résoudre.

## **2 - UN EXEMPLE DONT ON PEUT S'INSPIRER : IMPRIMANTES ET TRAITEMENTS DE TEXTE**

Actuellement, il existe des modèles d'imprimantes très divers, depuis les plus simples à aiguilles, jusqu'aux plus sophistiquées qui impriment en couleurs grâce à la technologie laser. Or les bons traitements de texte sont capables d'utiliser ces imprimantes, en fonction de leurs capacités.

Sous MS-DOS, on peut prendre comme référence un logiciel assez ancien, Word 4, de Microsoft. Il est livré sous forme d'un ensemble d'une dizaine de disquettes. Parmi celles-ci, deux seulement correspondent au programme principal. Parmi les autres, il y en a deux qui correspondent aux "pilotes d'imprimantes" : ce sont de petits fichiers ayant l'extension .DIM. Il suffit de recopier le bon fichier sur sa disquette ou son disque dur pour que Word 4 puisse utiliser l'imprimante disponible au mieux de ses capacités. Si, dans ces différents pilotes d'imprimantes ne figure pas la vôtre, vous pouvez (avec un certain travail...) créer un pilote vous-mêmes, grâce à un programme fourni, ce qui permet d'utiliser ce logiciel de traitement de texte même pour des imprimantes non prévues initialement.

En prenant comme exemple plus moderne l'environnement Windows, il y est défini des fichiers .DRV, qui correspondent à la définition des imprimantes : si l'on a choisi le bon fichier .DRV, l'impression se fera correctement, sans que l'on ait à se soucier du détail de l'impression.

Est-il possible d'utiliser ce principe pour les interfaces de mesures, c'est à dire de créer de petits fichiers, caractéristiques des interfaces, et qui permettraient à un programme principal d'utiliser ces interfaces de mesure, même si celles-ci n'avaient pas été prévues initialement ? En allant plus loin, est-il possible d'écrire des programmes d'expérimen

tation assistée par ordinateur sans se soucier des détails de la prise de mesure ?

La réponse apportée par ce texte est OUI, et la réalisation de ces pilotes peut être assez simple. Une distinction sera faite entre la programmation de logiciels pour Windows et la programmation pour MS-DOS.

### **3 - POUR MS-DOS, DES PILOTES EXÉCUTABLES RÉSIDANT EN MÉMOIRE ET ACTIVÉS PAR UNE INTERRUPTION**

#### **3-1. Principales définitions**

Un programme exécutable résidant en mémoire, c'est un programme (extension .EXE ou .COM) qui, après son lancement, reste dans la mémoire vive de l'ordinateur, bien qu'il soit apparemment terminé et que l'utilisateur puisse lancer d'autres programmes. Il occupe donc une partie de la mémoire, mais tant qu'il n'est pas activé, il ne se manifeste pas.

Les interruptions sont des arrêts de fonctionnement du programme principal, ce qui permet le fonctionnement d'autres programmes, en particulier des petits programmes chargés d'une mission particulière (test du clavier, mise à l'heure de l'horloge interne, contrôle du bon fonctionnement des disquettes...). Il existe 256 "vecteurs d'interruption", c'est à dire des adresses indiquant où est le programme à exécuter pour l'interruption considérée. Certains de ces vecteurs d'interruption ne sont pas définis par le système d'exploitation, et peuvent donc servir de base à une interruption "artificielle", choisie par le programmeur, et qui pourra réaliser quelques opérations simples, par exemple faire des mesures et transmettre le résultat au programme principal...

Si on imagine une interface nommée MACHINE, on pourra utiliser un petit programme nommé MACHINE.EXE, résidant en mémoire, qui sera activé par une interruption "artificielle" déclenchée par le logiciel principal. Il fera la mesure, et renverra le résultat vers le programme principal.

#### **3-2. Réalisation pratique**

Pour réaliser pratiquement un pilote, il faut obligatoirement un langage compilé (ou le langage assembleur) ; le langage Turbo Pascal,

présent dans tous les lycées, convient très bien, et il est probable que les langages C et C++ font aussi l'affaire.

Pour appeler un pilote, il faut pouvoir appeler les interruptions. Les mêmes langages conviennent aussi très bien.

En Basic, le langage compilé Quick Basic peut appeler les interruptions. Les langages interprétés GW-BASIC et QBASIC (fourni avec le système MS-DOS depuis la version 5) ne le peuvent pas, mais il suffit d'y greffer une petite partie en langage machine (quelques lignes de "DATA"... ) pour leur ajouter cette fonction.

#### **4 - POUR WINDOWS, DES LIBRAIRIES À LIAISON DYNAMIQUE (DLL)**

Le système des pilotes exécutables résidant en mémoire est encore utilisable sous Windows, mais cet environnement offre des possibilités supplémentaires.

##### **4-1. Principe des DLL**

Les Librairies à Liaisons Dynamiques sont des modules contenant des instructions exécutables destinées à des programmes d'application ou à d'autres DLL. Elles sont utilisables sous Windows et sous DOS en mode protégé. Le code et les ressources d'une DLL sont stockés dans un fichier exécutable dont l'extension est .DLL, qui doit être disponible lorsque l'application est exécutée.

Elles permettent le partage de code exécutable et des ressources par plusieurs applications Windows.

La majorité des logiciels sous Windows utilisent des DLL. Pour le constater, vous pouvez rechercher les fichiers ayant l'extension .DLL dans votre répertoire C:\WINDOWS\SYSTEM: vous en trouverez sûrement au moins une quinzaine.

Ces DLL contiennent donc des fonctions et procédures "exportables" vers des programmes extérieurs, c'est à dire utilisables par ce programme extérieur.

Dans leur principe, les DLL sont proches du concept d'unité tel qu'on peut l'utiliser dans les langages Pascal, mais contrairement à ces unités, le code des DLL n'es pas lié au programme qui l'utilise : les appels de fonctions et de procédures dans le programme sont liés de façon

dynamique à leur point d'entrée dans la DLL, c'est à dire au moment même de l'exécution de l'application. De plus, les DLL ne peuvent exporter que des procédures et des fonctions, alors que les unités classiques peuvent aussi exporter des types, des constantes, des variables, des objets...

Les DLL sont définies dans le standard Windows. Par conséquent, un programme pourra utiliser une DLL initialement écrite pour un autre, à condition de respecter le protocole d'appel de ses fonctions. De plus, on peut changer de DLL en cours d'exécution du programme, exactement comme on peut changer de pilote d'imprimante au cours de l'utilisation d'un logiciel de traitement de texte. C'est là un autre avantage par rapport aux pilotes résidant en mémoire, utilisables sous DOS.

Les DLL sont particulièrement bien adaptées aux projets de programmation multi-langages, puisqu'une DLL écrite et compilée avec un langage de programmation peut parfaitement être utilisée avec un programme d'application écrit dans un autre langage. On peut donc appeler dans un programme écrit en Visual Basic ou en C++ une DLL écrite en Turbo Pascal pour Windows, et inversement.

#### **4-2. Importation de fonctions de mesure par les tableurs grâce aux DLL**

Les tableurs disponibles dans le commerce n'ont pas été spécialement conçus pour faire des mesures à partir d'interfaces électroniques. Si on regarde leur publicité et leur documentation, ils sont plutôt conçus pour des applications comptables, bancaires ou de gestion commerciale. Néanmoins, les principaux d'entre eux, notamment EXCEL et QuattroPro pour Windows, peuvent appeler ("importer") des fonctions à partir de DLL externes, écrites indépendamment d'eux.

Notre propos étant d'utiliser ces tableurs pour faire des mesures à partir d'interfaces électroniques, il va falloir utiliser des DLL spécifiques de ces interfaces, qui "exporteront" leurs fonctions de mesure vers QPW ou Excel.

La documentation livrée avec ces tableurs est souvent insuffisante pour comprendre la programmation et l'importation de fonctions, mais l'essentiel est là : on peut écrire des DLL spécifiques des interfaces de mesure, et appeler les fonctions de ces DLL à partir des tableurs, ce qui

rend les tableurs capables d'obtenir des valeurs numériques correspondant aux grandeurs physiques mesurées.

## **5 - NORMALISATION DES PILOTES**

Il est hautement souhaitable que tous ceux qui souhaitent utiliser ce système de pilotes s'accordent pour utiliser les mêmes interruptions, avec les mêmes numéros de fonctions et de paramètres, de manière que les pilotes écrits par les uns puissent fonctionner immédiatement avec les logiciels d'application écrits par les autres (ou par les mêmes).

Ce système de pilotes des du DOMAINE PUBLIC. Donc si vous voulez écrire des logiciels utilisant ces pilotes, faites-le : ils fonctionneront directement avec toutes les interfaces, même celles dont vous ne connaissez pas les détails de programmation. Inversement, si vous avez une interface ne figurant pas sur la liste ci-dessous, il suffit d'en écrire le pilote, et tous les logiciels pourront l'utiliser.

### **5-1. Détails pour les pilotes exécutables pour DOS**

Le principe a été publié dans la revue "Pascalissime" de Avril-Juin 94 (adresse : Pascalissime, 26 rue Lamartine, 75009 PARIS, tél (1) 42 85 10 82), avec divers détails pour l'identification en mémoire, la désinstallation du pilote, les numéros de voies, etc.

### **5-2. Pour les librairies à liaisons dynamiques sous Windows**

Le détail en sera publié dans un livre à paraître chez Masson sur l'emploi des tableurs par les sciences expérimentales.

### **5-3. N'hésitez pas à contacter l'auteur !**

Un grand nombre d'essais ont déjà été faits, et je pense avoir sélectionné des solutions convenables pour tous les cas de figure. Je n'ai rencontré aucun échec, et toutes les interfaces se sont laissées "piloter" : si l'on peut afficher une valeur numérique à l'écran, on peut la transmettre par une interruption ou par une liaison à librairie dynamique.

Si vous avez une interface non listée ci-dessous, ou si vous programmez des logiciels d'expérimentation, contactez-moi, pour que nous unissions nos efforts :

Pierre DIEUMEGARD  
 professeur de Sciences Naturelles  
 Lycée Thiers, 13232 Marseille

## **QUELS LOGICIELS ET QUELS MATÉRIELS SONT UTILISABLES AVEC CES PILOTES D'INTERFACES DE MESURE ?**

### **Pilotes existant déjà :**

Cette liste ne demande qu'à s'allonger, grâce à votre aide. Plusieurs pilotes n'ont été programmés que grâce à des documents de seconde main, sans optimisation. Il est donc probable qu'on peut faire mieux.

CANDIBUS EXE	(carte Candibus, de Langage et Informatique)
CAPMAN EXE	(prise "manettes de jeu")
CAPPAR1 EXE, CAPPAR2 EXE	(système Cappar, prise parallèle 1 ou 2)
CASSY EXE	(interface Cassy, de Leybold)
CHROPC1 EXE	(pour la mesure des fréquences sur ChronoPC, prise parallèle 1)
FREQ250 EXE	(mesure la fréquence du pilote installé à l'interruption 250)
FREQCPMN EXE	(mesure la fréquence de l'interface Capman=manettes de jeux)
FREQPMB EXE	(mesure la fréquence de l'interface PMB)
HLAB21 EXE et HLAB22 EXE	(pHmètre Schott Handylab2 à sortie RS232, connecté en Com1 : ou Com2 :)
JEULIN EXE	(interface Jeulin ESAO3)
MY77E1 EXE et MY77E2 EXE	(Multimètre MY77 en Com1 : ou Com2 :, diffusé par Maison des Enseignants)
ORPHY1 EXE, ORPHY2 EXE	(système Orphy en Com1 : ou Com2 :, diffusé par Micrelec)



PCMES EXE	(interface Eurosmart PCMES2)
PMBA EXE	(interface PMB, de la Maison des Enseignants de Provence)
PMBB EXE	(interface PMB, installé à l'adresse 310h)
SARTOR1 EXE	(balance Sartorius PT600, vendue par Maison des Enseignants, en Com1 :)
SARTOR2 EXE	(balance Sartorius PT600, en Com2 :)
SMF10 EXE	(Système Pierron SMF10)
SOURIS EXE	(pour faire des mesures de déplacement à l'aide de la souris)

### **Logiciels d'application**

Outre divers programmes du domaine public, j'ai fait des logiciels commerciaux, diffusés par la Maison des Enseignants de Provence (B.P. 194, 13264 Marseille cedex 07, fax 91 52 30 87)

SYMBIOTE, SEMISYMB et SYMBTAB sont des logiciels résidant en mémoire qui injectent les données dans le "tampon-clavier" de l'ordinateur, ce qui permet de les faire apparaître directement dans les cases des tableurs-grapheurs sous DOS.

Mesugraf est un logiciel généraliste d'acquisition pour les mesures lentes (intervalle entre les mesures de l'ordre de la seconde), qui affiche les mesures (1 ou 2 voies) sur un écran graphique, sauve les résultats dans des fichiers compatibles avec les tableurs, fait des transformations de variables... Oscillo en est la version pour les mesures rapides (de type oscilloscope : intervalle entre les mesures de l'ordre de la milliseconde), avec en plus la représentation de la transformée de Fourier. La version 1995 permet notamment la sauvegarde et la relecture de fichiers de configuration.

Depuis 1994, ces logiciels fonctionnent avec les pilotes résidents décrits ici. Avec ces logiciels sont fournis tous les pilotes cités précédemment, sous forme exécutable, ainsi que quelques uns en programme-source (Turbo Pascal).

Une version pour Windows, avec les pilotes sous forme de DLL est à l'étude, mais non encore réalisée... En attendant, n'oubliez pas que les

tableurs commerciaux, avec quelques macrocommandes peuvent vous permettre de faire rapidement des mesures, et de les représenter graphiquement.