

Comment utiliser PAST pour l'acquisition automatique de données (EXAO)

Pierre DIEUMEGARD
Lycée Pothier
2 bis rue Marcel Proust
45000 Orléans (France)
pierre.dieumegard@ac-orleans-tours.fr

PAST¹ est un logiciel d'analyse de données : d'abord il faut obtenir des données, et ensuite on utilise Past pour faire des statistiques et des graphiques.

Mais il ya une option de menu sous-utilisée, dans la partie droite du menu : “script”. En fait, ce “langage de script” est un langage de programmation, un langage Pascal.

En comparaison des grands langages de programmation comme Delphi ou Python, ce langage de script est très pauvre, mais la nouvelle version de Past (3.11) a une possibilité très puissante : elle peut utiliser des bibliothèques dynamiques (.dll de Windows).

Les bibliothèques dynamiques sont des morceaux de programmes exécutables, avec des fonctions spéciales. Les bibliothèques dynamiques ne peuvent pas être utilisées seules, elle doivent être appelées par un programme principal : ici, le programme principal est Past.

Il y a des bibliothèques dynamiques ayant des rôles très divers : ici nous ne parlerons que de mesurage, et de la façon de donner des données à Past. Parce qu'il existe un grand nombre d'appareils de mesure, ce document décrira seulement quelques appareils faciles à trouver sur un PC : l'horloge interne et la carte-son, mais la recette est la même pour des appareils de mesure scientifique plus sérieux : pHmètres, thermomètres, balances, spectrophotomètres, ou autres appareils. Nous aurons seulement besoin de la bonne bibliothèque dynamique, et de connaître les bonnes fonctions dans ladite bibliothèque dynamique.

Les bibliothèques dynamiques peuvent être programmées par divers langages habituels de programmation : FreeBasic, PureBasic, Delphi, FreePascal, C, C++... On peut obtenir des bibliothèques dynamiques pour de nombreux appareils à <http://sciencexp.free.fr>

Parce que le langage de script de Past est du type Pascal, les exemples de bibliothèques dynamiques décrits ici sont aussi en Pascal (Delphi ou FreePascal) : si vous pouvez comprendre ce langage de script, vous pouvez aussi comprendre le code-source de la bibliothèque dynamique.

1 Un exemple avec l'horloge interne de l'ordinateur (fonction stdeouble)

La bibliothèque dynamique est « bibdysyst_delphi2016.dll » (source : bibdysyst_delphi2016.dpr).

Il y a une fonction « stdeouble », pour l'entrée analogique, et stdeouble(0) donne le nombre de

¹ Hammer, Ø., Harper, D.A.T., and P. D. Ryan, 2001. PAST: Paleontological Statistics Software Package for Education and Data Analysis. Palaeontologia Electronica 4(1): 9pp.

secondes dans la minute, à partir de l'horloge interne. Cette fonction a un paramètre (« double » : a float number with double precision) qui est le numéro de la voie, et donne un résultat (aussi un nombre réel en double précision).

La première ligne charge une fonction nommée `stdeouble` à partir de la bibliothèque « `bibdynsyst_delphi2016.dll` ».

```
function stdeouble(n:double):double;stdcall;external
'bibdynsyst_delphi2016.dll';
```

La boucle entre les lignes 11 et 25 envoie des données vers l'onglet de texte, l'onglet graphique et l'onglet tableur. La valeur X est le temps, en secondes, à partir de la fonction `time` de Past. La valeur Y est aussi le temps en secondes, mais différent : le nombre de secondes à l'intérieur de la minute actuelle, à partir de la bibliothèque dynamique (parce que la voie 0 donne cette valeur). La ligne 24 fait une pause de 1000 millisecondes.

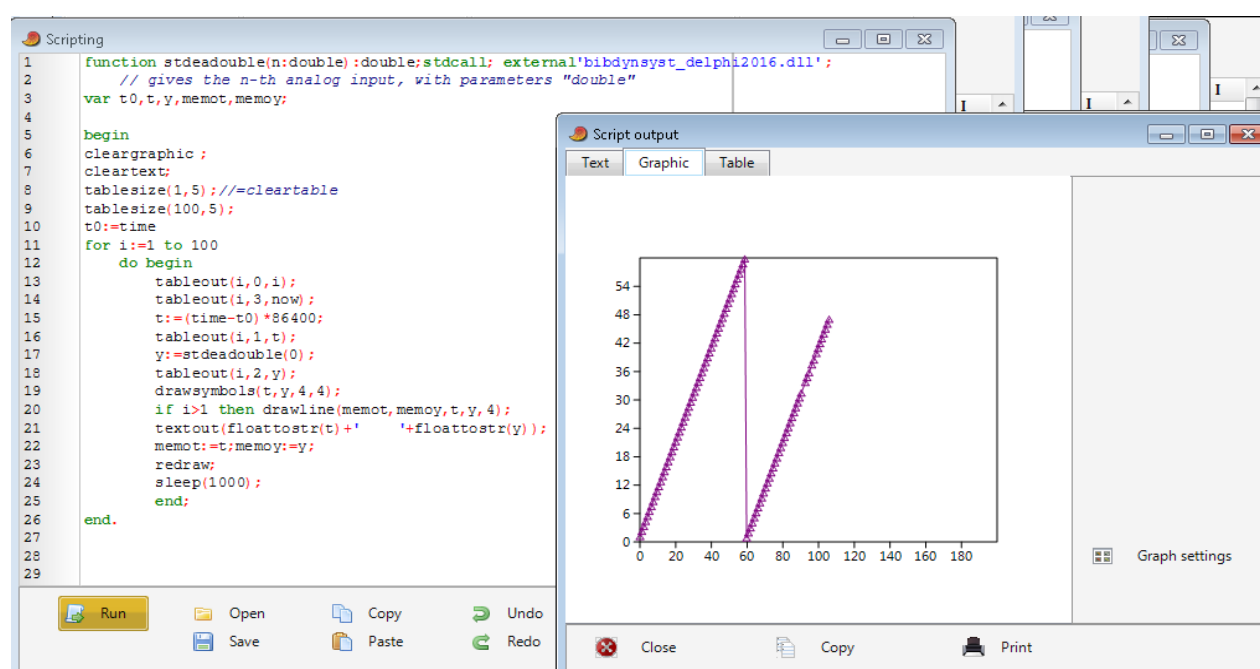


Illustration 1: Mesures faites par `bibdynsyst_Delphi2016.dll` : secondes dans la minute

Exercice : à partir de l'onglet tableur (3e onglet), copiez les données vers le presse-papier, et ensuite collez-les vers le tableur principal de Past. Faites le graphique des colonnes B et C.

Exercice : copiez les données de l'onglet tableur vers le presse-papier, puis collez-les dans votre tableur habituel (LibreOffice, Excel, PlanMaker, Gnumeric...). Faites el graphique de ces valeurs.

2 Comment choisir les voies de mesure : fonction `stdnead`

Dans le système Mensurasoft, toutes les fonctions numériques ont un nom (une chaîne de caractères). Si le nom est vide (longueur de la chaîne égale à zéro), la fonction numérique n'existe pas.

Le second petit programme :

- montre la seconde dans la minute (stddeadouble(0), comme le premier programme.
- montre deux fois le détail de ce pilote (fonction stddetail). La première fois, c'est directement par la fonction stddetail, mais avec des idéogrammes, parce que stddetail donne une chaîne ASCII-ANSI, alors que Past utilise des chaînes Unicode. La seconde fois, c'est avec décodage par « decode_unicode », ce qui donne une chaîne avec des caractères latins.
- montre les six voies d'entrée analogique (= les 6 canaux). Chaque ligne commence par un nombre (0 à 5), donne le nom de la voie, et la valeur numérique de cette entrée analogique. Le canal 0 est la seconde dans la minute, le canal 1 est la date et le temps (comme la fonction now dans le langage Past), le canal 2 est la seconde dans le jour, le canal 3 est la position X de la souris, le canal 4 est la position Y de la souris. Parce que la canal 5 n'existe pas, son nom est une chaîne de longueur nulle, et la valeur est -777.

```

1 function stddead(n: integer):double;stdcall;external'bibdynsyst_delphi2016.dll';
2 function stddeadouble(n:double):double;stdcall; external'bibdynsyst_delphi2016.dll';
3 function stdnead(n:integer):pchar;stdcall;external 'bibdynsyst_delphi2016.dll';
4 function stdneadouble(n:double):pchar;stdcall;external 'bibdynsyst_delphi2016.dll';
5 function stddetail:pchar;stdcall;external'bibdynsyst_delphi2016.dll';
6
7 function decodeunicode(ch):string;
8 //because Past uses Unicode and Mensurasoft uses Ascii
9 begin
10  l:=length(ch); r:='';
11  if l>0 then for i:=1 to l do
12  begin
13    c:=chr(ord(ch[i]) mod 256); if c=chr(0)
14    then break else r:=r+c;
15    c:=chr(ord(ch[i]) div 256);if c=chr(0)
16    then break else r:=r+c;
17  end;
18  result:=r;
19 end;
20
21
22 cleartext();
23 textout(stddeadouble(0)); //similar to stddead(0)
24 textout(stddetail());
25 textout(decodeunicode(stddetail()));
26 for i:=0 to 5 do
27   textout(inttostr(i)+'-'+decodeunicode(stdnead(i))+'-->'+floattostr(stddead(i)));
28
29 textout('The end !!!!');
30

```

Script output

```

44.312
44.312
DLL (32 bits) for system PC, by P. Dieumegard, 2016/03/01
0:seconds in the minute-->46.277
1:date,time-->42431.8685911806
2:secondes in the day-->75046.279
3:mouse X-->515
4:mouse Y-->548
5:-->-777
The end !!!!

```

Illustration 2: un petit programme pour explorer les entrées analogiques : stdnead(n) donne le nom de la fonction d'entrée stddead(n). Il y a 5 vraies fonctions (0 à 4), et la suivante n'existe pas.

Les premières lignes du programme chargent les fonctions à partir de la bibliothèque. « pchar » est une chaîne (à zéro terminal).

Exercice : modifier le premier programme, pour tracer la position Y de la souris en fonction du temps.

3 Explorer les autres fonctions : sorties analogiques, entrées et sorties binaires

stdsad(n, v) peut mettre la valeur v à la n-ième voie de sortie analogique (n est un entier, et v est un double ; cette fonction donne un double)

stdeb(n) donne l'état de la n-ième voie d'entrée binaire (n est un entier, et cette fonction renvoie un entier)

stdsb(n, v) peut mettre la voie de sortie binaire n à la valeur v. Les paramètres sont des entiers. Cette fonction donne un entier, mais ce n'est pas important.

Cette bibliothèque bibdynsyst_delphi2016.dll n'a pas de sortie analogique. Les entrées binaires sont les boutons de la souris, et les sorties binaires sont les touches de verrouillage du clavier, et le lecteur de CD : avec ce programme, le lecteur de CD doit s'ouvrir.

Nous pouvons obtenir ces fonctions par :

```
function stdsad(n:integer ; v:double):double;stdcall;external
'bibdynsyst_delphi2016.dll';

function stdeb(n:integer):integer;stdcall;external
'bibdynsyst_delphi2016.dll';

function stdsb(n:integer;v:integer):integer;stdcall;external
'bibdynsyst_delphi2016.dll';
```

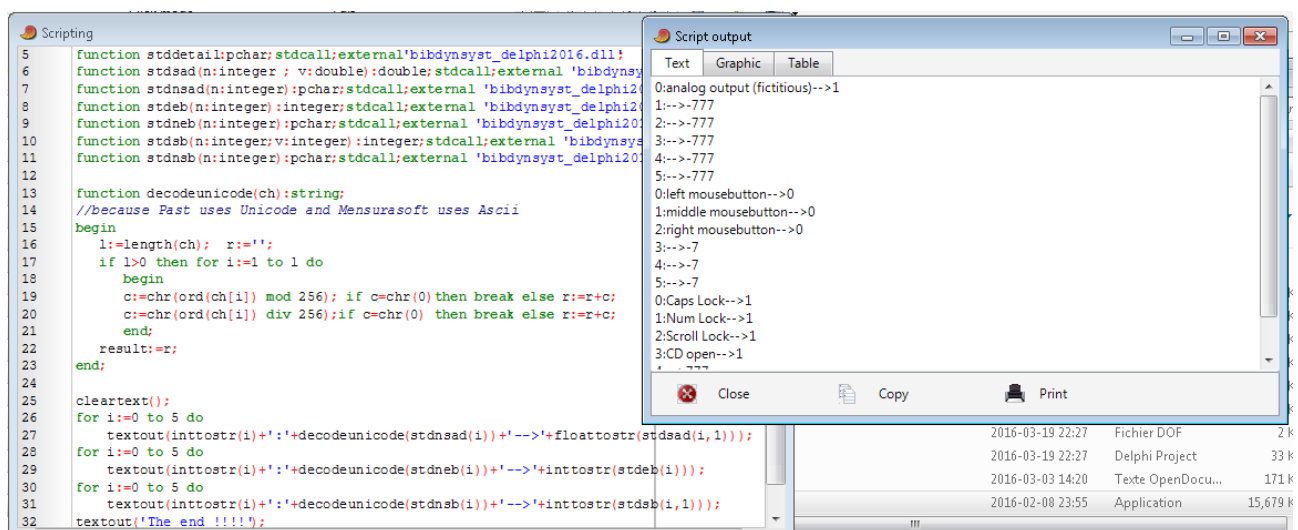


Illustration 3 : emploi des sorties analogiques, entrées binaires et sorties binaires.

Exercice : vous êtes malheureux parce que ce programme ouvre le lecteur de CD et bloque votre clavier. Modifiez-le !

4 Comment étudier le son, avec un microphone de la carte-son

Jusqu'à maintenant, la bibliothèque était seulement pour un test, elle ne pouvait pas faire de vraies mesures intéressantes.

Sur Internet, on peut trouver « port.dll » (<http://www.b-kainka.de/port.zip>). Cette bibliothèque peut être utilisée avec les ports sériels (pour un grand nombre de pHmètres, de balances, de spectrophotomètres, etc), et aussi avec le microphone de la carte-son : tous les ordinateurs portables

peuvent l'utiliser.

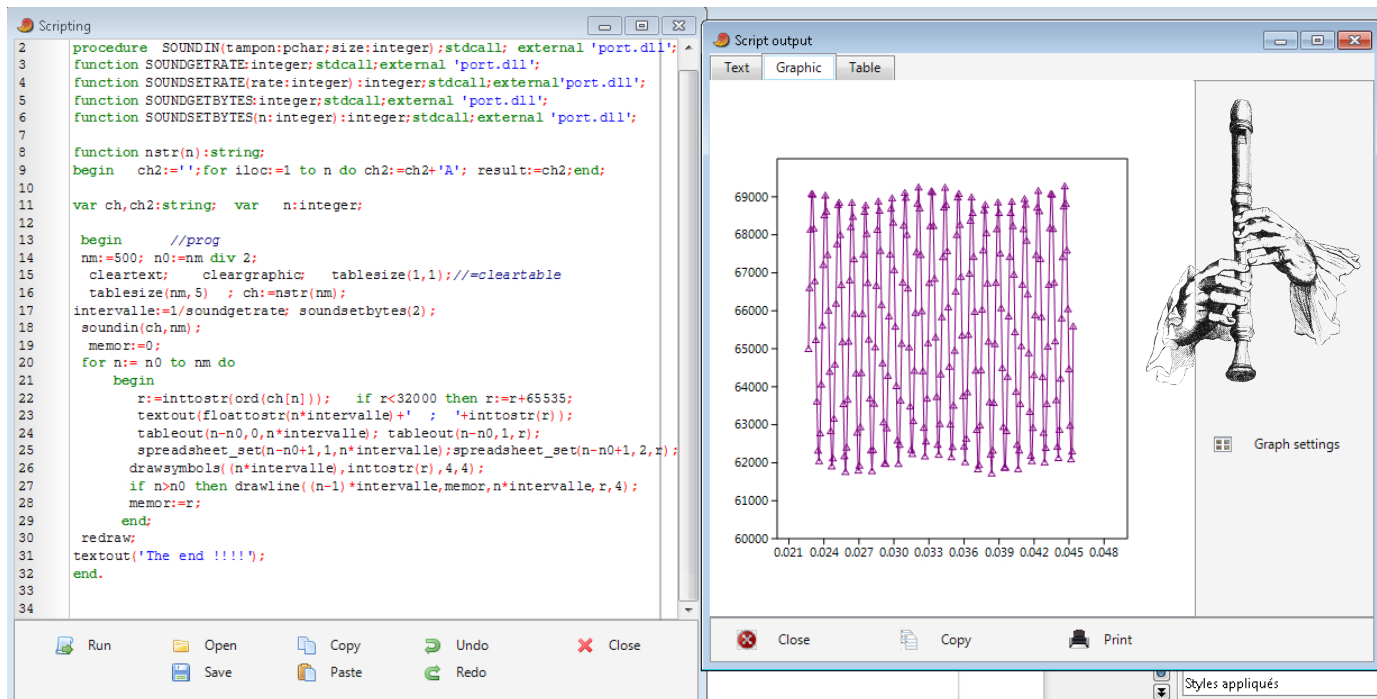


Illustration 4: Enregistrement d'une flûte à bec (soprano) : A (= la)

Ce petit programme envoie des données au tableur principal de Past. Avec lui, nous pouvons utiliser « Time series / Spectral analysis / Simple periodogram », et obtenir la fréquence de ce son.

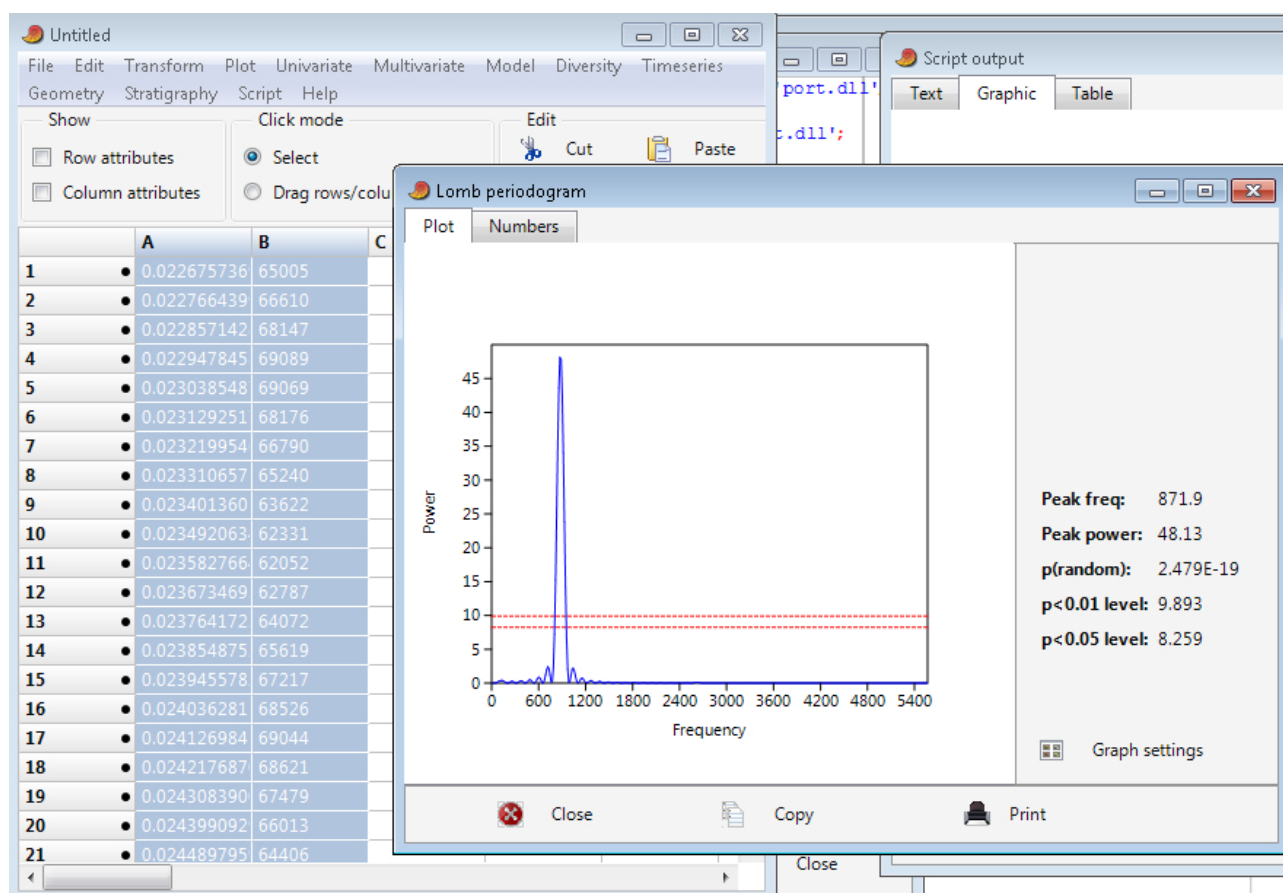


Illustration 5: Périodogramme du son "A" (la) d'une flûte à bec soprano

Nous pouvons constater que la fréquence de ce son est 871,9 Hz (en théorie 880 Hz).

Exercice : Parlez vers le microphone pendant l'exécution de ce programme, et essayez l'option "periodogram" pour voir les principales fréquences de votre voix. Comparez la voix d'un homme à la voix d'une femme.

5 Conclusion

Ces petits programmes montrent que Past peut utiliser les fonctions des bibliothèques dynamiques (DLL).

Le travail principal de Past est de tracer des graphiques et d'analyser des données. Son langage de script, en utilisant des bibliothèques dynamiques, peut faire un pont vers le monde réel.

La bibliothèque port.dll est utilisable pour de nombreux appareils.

A <http://sciencexp.free.fr> vous trouverez des bibliothèques pour divers appareils.

Vous pouvez aussi utiliser vos propres DLL, faites par vos soins avec un langage compilé (FreeBasic, PureBasic, C/C++, Pascal/Delphi...).