

## Adaptation du système Mensurasoft à Scilab (essai fait sous Windows 7)

Scilab est capable d'utiliser des bibliothèques dynamiques, mais avec des fonctions particulières. Les paramètres entiers doivent être passés comme des « pointeurs d'entiers », les paramètres de sortie ne sont pas des résultats de fonctions, mais des paramètres variables : paramètres passés par variable (= par référence = par adresse) au lieu d'être des résultats renvoyés par la fonction comme dans le système Mensurasoft normal.

Deux solutions sont possibles :

- 1) utiliser un pilote adaptateur, contenant des fonctions pouvant échanger des données avec Scilab, et appelant lui-même un pilote normal de Mensurasoft.
- 2) utiliser des pilotes Mensurasoft modifiés, auxquels on a ajouté les fonctions lisibles par Scilab. C'est ce qui est décrit ici. Les fonctions seront préfixées Scilab\_, et la fin du nom de la fonction sera le nom de base de Mensurasoft : titre, detail, ead, nead, sad, nsad, eb, neb, sb et nsb. Il n'y a pas de difficulté fondamentale pour les fonctions numériques ; par contre, je n'ai pas réussi à faire de fonction renvoyant correctement des chaînes de caractères lisibles par Scilab, c'est pourquoi ces fonctions renvoient un tableau de valeurs numériques, et il faut qu'ensuite Scilab transforme ce tableau de valeurs en chaîne.

## Fonctions à ajouter dans le pilote Mensurasoft

Le langage utilisé ici est Pascal (tests effectués en Delphi sous Windows, mais le code est le même pour FreePascal, sous Windows comme sous Linux).

```
//fonctions pour Scilab
type pword=^word;
const longmax=100;
type ttabdouble=array[1..longmax] of double;

procedure Scilab_detail(var res:ttabdouble); cdecl;export;
var iloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=cdetail();
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

procedure Scilab_titre(var res:ttabdouble); cdecl;export;
var iloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=ctitre();
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

procedure Scilab_nead(n:pword; var res:ttabdouble);cdecl;export;
var iloc,nloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
nloc:=n^;
```

```

for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=cnead(nloc);
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
    then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

```

```

procedure Scilab_ead(n:ppword;var res:double);cdecl;export;
var rloc:double;
var nloc:word;
begin
nloc:=n^;
rloc:=cead(nloc);
res:=rloc;end;

```

```

procedure Scilab_sad(n:ppword;var valeur:double;var res:double);cdecl;export;
var rloc:double;
var nloc:word;
begin
nloc:=n^;
rloc:=csad(nloc,valeur);
res:=rloc;end;

```

```

procedure Scilab_nsad(n:ppword; var res:ttabdouble);cdecl;export;
var iloc,nloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
nloc:=n^;
for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=cnsad(nloc);
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
    then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

```

```

procedure Scilab_eb(n:ppword;var res:double);cdecl;export;
var nloc:word;
begin
nloc:=n^;
res:=ceb(nloc);
//res:=1;
end;

```

```

procedure Scilab_neb(n:ppword; var res:ttabdouble);cdecl;export;
var iloc,nloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
nloc:=n^;
for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=cneb(nloc);
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
    then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

```

```

procedure Scilab_sb(n:ppword;var valeur:double;var res:double);cdecl;export;
var rloc:double;
var nloc:word;
begin
nloc:=n^;
rloc:=csb(nloc,round(valeur));
res:=rloc;end;

```

```

procedure Scilab_nsb(n:ppword; var res:ttabdouble);cdecl;export;

```

```

var iloc,nloc,longueur:word;
var chloc:string;
var tabdouble:ttabdouble;
begin
nloc:=n^;
for iloc:=1 to longmax do tabdouble [iloc]:=0;
chloc:=cnsb(nloc);
if length(chloc)>longmax then longueur:=longmax else longueur:=length(chloc);
if length(chloc)>0
  then for iloc:=1 to longueur do tabdouble[iloc]:=ord(chloc[iloc]);
res:=tabdouble;
end;

```

## Appel par Scilab

Il n'y a pas de problème particulier pour les fonctions devant renvoyer des valeurs numériques (les « vraies fonctions » de Mensursoft, celles qui lisent des valeurs sur les appareils de mesure ou qui déclenchent des actionneurs).

Par contre, pour les fonctions devant renvoyer des chaînes de caractères, je n'ai pas trouvé comment faire directement. C'est pourquoi les fonctions renvoient un tableau de nombres, qui est ensuite converti en chaîne de caractères. Ce n'est pas génial, mais ça fonctionne ; si quelqu'un connaît une meilleure solution, je suis intéressé.

```

// script de chargement d'un pilote Mensursoft spécial Scilab
// logiciel libre // libera programaro // free software //
// Pierre Dieumegard, pierre.dieumegard@ac-orleans-tours.fr
// http://sciencexp.free.fr

nomfich='./bibdyn_systeme_delphi_Scilab.dll'; // à changer pour un vrai pilote

numh0=link(nomfich,'Scilab_detail','c');
numh1=link(nomfich,'Scilab_titre','c');
numh2=link(nomfich,'Scilab_ead','c');
numh3=link(nomfich,'Scilab_nead','c');
numh4=link(nomfich,'Scilab_sad','c');
numh5=link(nomfich,'Scilab_nsad','c');
numh6=link(nomfich,'Scilab_eb','c');
numh7=link(nomfich,'Scilab_neb','c');
numh8=link(nomfich,'Scilab_sb','c');
numh9=link(nomfich,'Scilab_nsb','c');
function res=detail();
  pos=1:100;
  res=stripblanks(ascii(call('Scilab_detail',pos,1,'d','out',[1,100],1,'d')));
endfunction;

function res=titre();
  pos=1:100;
  res=stripblanks(ascii(call('Scilab_titre',pos,1,'d','out',[1,100],1,'d')));
endfunction;

function res=ead(n);
  res=call('Scilab_ead',n,1,'i','out',[1,1],2,'d');
endfunction;

function res=nead(n);
  pos=1:100;
  res=stripblanks(ascii(call('Scilab_nead',n,1,'i',pos,2,'d','out',[1,100],2,'d')));
endfunction;

function res=sad(n,val);
  res=call('Scilab_sad',n,1,'i',val,2,'d','out',[1,1],3,'d');
endfunction;

function res=nsad(n);
  pos=1:100;

```

```
res=stripblanks(ascii(call('Scilab_nsad',n,1,'i',pos,2,'d','out',[1,100],2,'d')));
endfunction;

function res=eb(n);
res=call('Scilab_eb',n,1,'i','out',[1,1],2,'d');
endfunction;

function res=neb(n);
pos=1:100;
res=stripblanks(ascii(call('Scilab_neb',n,1,'i',pos,2,'d','out',[1,100],2,'d')));
endfunction;

function res=sb(n,val);
res=call('Scilab_sb',n,1,'i',val,2,'d','out',[1,1],3,'d');
endfunction;

function res=nsb(n);
pos=1:100;
res=stripblanks(ascii(call('Scilab_nsb',n,1,'i',pos,2,'d','out',[1,100],2,'d')));
endfunction;

disp('la bibliothèque '+nomfich+' est chargée');
```