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

Freemat est capable d'utiliser des bibliothèques dynamiques du système Mensurasoft de base (à condition que les fonctions soient déclarées « cdecl », mais seulement pour les paramètres numériques. C'est l'essentiel pour faire des mesures et commander des actionneurs, mais pour un emploi plus confortable, il est souhaitable aussi de pouvoir utiliser des fonctions renvoyant des chaînes de caractères (fonctions « titre », « detail », « nead », « nsad », « neb », « nsb »), ce qui n'est pas possible directement avec les fonctions normales de Mensurasoft.

Deux solutions sont possibles :

1) utiliser un pilote adaptateur, contenant des fonctions pouvant échanger des données avec Freemat, et appelant lui-même un pilote normal de Mensurasoft.
2) utiliser des pilotes Mensurasoft modifiés, auxquels on a ajouter les fonctions lisibles par Freemat. C'est ce qui est décrit ici. Les fonctions seront préfixées Freemat_, et la fin du nom de la fonction sera le nom de base de Mensurasoft : titre, detail, nead, nsad, neb, nsb. Le principe est de transformer les fonctions renvoyant une chaîne de caractères en une fonction ayant un paramètre de plus, le numéro du caractère de la chaîne, et qui renvoie le numéro ASCII du caractère en question. Par exemple, la fonction Freemat_titre aura un paramètre d'entrée, le numéro du caractère du titre, et renverra le code ASCII de ce caractère. Si le code est égal à zéro, c'est que le numéro demandé est au delà de la longueur de la 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 Freemat
function Freemat_titre(pos:word):word; cdecl;export;
var chloc:string;
begin
chloc:=string(ctitre());
if pos<=length(chloc) then Freemat_titre:=ord(chloc[pos]) else Freemat_titre:=0;
end;
function Freemat_detail(pos:word):word; cdecl;export;
var chloc:string;
begin
chloc:=string(cdetail());
if pos<=length(chloc) then result:=ord(chloc[pos]) else result:=0;
end;

function Freemat_nead(n,pos:word):word; cdecl;export;
var chloc:string;
begin
chloc:=string(cnead(n));
if pos<=length(chloc) then result:=ord(chloc[pos]) else result:=0;
end;

function Freemat_nsad(n,pos:word):word; cdecl;export;
var chloc:string;
```

```
begin
chloc:=string(cnsad(n));
if pos<=length(chloc) then result:=ord(chloc[pos]) else result:=0;
end;

function Freemat_neb(n,pos:word):word; cdecl;export;
var chloc:string;
begin
chloc:=string(cneb(n));
if pos<=length(chloc) then result:=ord(chloc[pos]) else result:=0;
end;
function Freemat_nsb(n,pos:word):word; cdecl;export;
var chloc:string;
begin
chloc:=string(cnsb(n));
if pos<=length(chloc) then result:=ord(chloc[pos]) else result:=0;
end;
```

Appel par Freemat

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

- d'une part faire un script qui charge les fonctions du pilote (fonctions Freemat_titre, Freemat_detail, cead, Freemat_cnead, csad, Freemat_cnsad, ceb, Freemat_cneb, csb, Freemat_cnsb).
- d'autre part une série de fichiers de fonctions. titre.m reconstitue le titre à partir de la fonction Freemat_titre, detail.m reconstitue le détail à partir de Freemat_detail, nead utilise Freemat_nead, etc.

```
%script qui importe les fonctions à partir du pilote Mensurasoft
nompilote='./bibdyn_systeme_delphi_freemat5.dll';
import(nompilote,'Freemat_titre','Freemat_titre','int32','int32 pos');
import(nompilote,'Freemat_detail','Freemat_detail','int32','int32 pos');
import(nompilote,'Freemat_nead','Freemat_nead','int32','int32 n, int32 pos');
import(nompilote,'Freemat_nsad','Freemat_nsad','int32','int32 n, int32 pos');
import(nompilote,'Freemat_neb','Freemat_neb','int32','int32 n, int32 pos');
import(nompilote,'Freemat_nsb','Freemat_nsb','int32','int32 n, int32 pos');

import(nompilote,'cead','cead','double','int32 n');
import(nompilote,'ceb','ceb','int32','int32 n');
import(nompilote,'csad','csad','double','int32 n, double v');
import(nompilote,'csb','csb','int32','int32 n, int32 v');

%fichier de fonction pour detail
function res=detail()
    ch='';n=1;
    while Freemat_detail(n)>0;
        ch=[ch,Freemat_detail(n)];
        n=n+1;
    end;
    res=ch;

%fichier de fonction pour titre
```

```
function res=titre()
    ch='';n=1;
    while Freemat_titre(n)>0;
        ch=[ch,Freemat_titre(n)];
        n=n+1;
    end;
    res=ch;

%fichier de fonction pour nead
function res=nead(n)
    ch='';pos=1;
    while Freemat_nead(n,pos)>0;
        ch=[ch,Freemat_nead(n,pos)];
        pos=pos+1;
    end;
    res=ch;


%fichier de fonction pour nsad
function res=nsad(n)
    ch='';pos=1;
    while Freemat_nsad(n,pos)>0;
        ch=[ch,Freemat_nsad(n,pos)];
        pos=pos+1;
    end;
    res=ch;

%fichier de fonction pour neb
function res=neb(n)
    ch='';pos=1;
    while Freemat_neb(n,pos)>0;
        ch=[ch,Freemat_neb(n,pos)];
        pos=pos+1;
    end;
    res=ch;


%fichier de fonction pour nsb
function res=nsb(n)
    ch='';pos=1;
    while Freemat_nsb(n,pos)>0;
        ch=[ch,Freemat_nsb(n,pos)];
        pos=pos+1;
    end;
    res=ch;
```