

Pierre Dieumegard
professeur de SVT,
Lycée Pothier, 2 bis rue Marcel Proust,
F 45044 Orléans
courriel : pierre.dieumegard@ac-orleans-tours.fr

Mensurasoft-Freemat : Mesures et expérimentations scientifiques en Freemat

Freemat est un logiciel de calcul numérique, présenté comme un "clone" de Matlab, qui existe aussi bien sous Linux que sous MS-Windows ou Mac-OS. Il est très puissant pour analyser et visualiser les données, mais on peut aussi l'utiliser pour faire l'acquisition automatique des données, à l'aide d'un appareil de mesure connecté à l'ordinateur. Pour que Freemat puisse visualiser ou faire le traitement statistique de données scientifiques, il faut d'abord lui fournir de telles données. L'objet de ce texte est de montrer comment on peut programmer Freemat pour qu'il fasse directement l'acquisition de données à partir d'appareils de mesure scientifiques (thermomètres, pHmètres, interfaces de mesures variées).

Par rapport à Matlab, Freemat est plutôt moins puissant pour ce qui est de la variété des fonctions de calcul numérique, mais il a plusieurs avantages intéressants :

- C'est un logiciel libre : on peut (à la fois légalement et techniquement) modifier le programme-source pour changer certaines propriétés. On peut le télécharger sur <http://freemat.sourceforge.net/>
- C'est un logiciel portable dans le système d'exploitation Windows : on peut le copier sur une clé USB et l'utiliser sur un autre ordinateur. Pour le système d'exploitation Linux, il est moins portable, mais on peut utiliser l'émulateur Wine pour faire fonctionner Freemat pour Windows sous Linux.
- Freemat pour Windows peut utiliser des bibliothèques dynamiques, c'est à dire des fichiers contenant du code exécutable, sous forme de fonctions. Par l'appel de ces bibliothèques dynamiques (d'extension traditionnelle .dll sous MS-Windows), Freemat peut ainsi être doté de nouvelles fonctions. Malheureusement, cette possibilité est sous-employée et sous-documentée.

Freemat a quelques inconvénients :

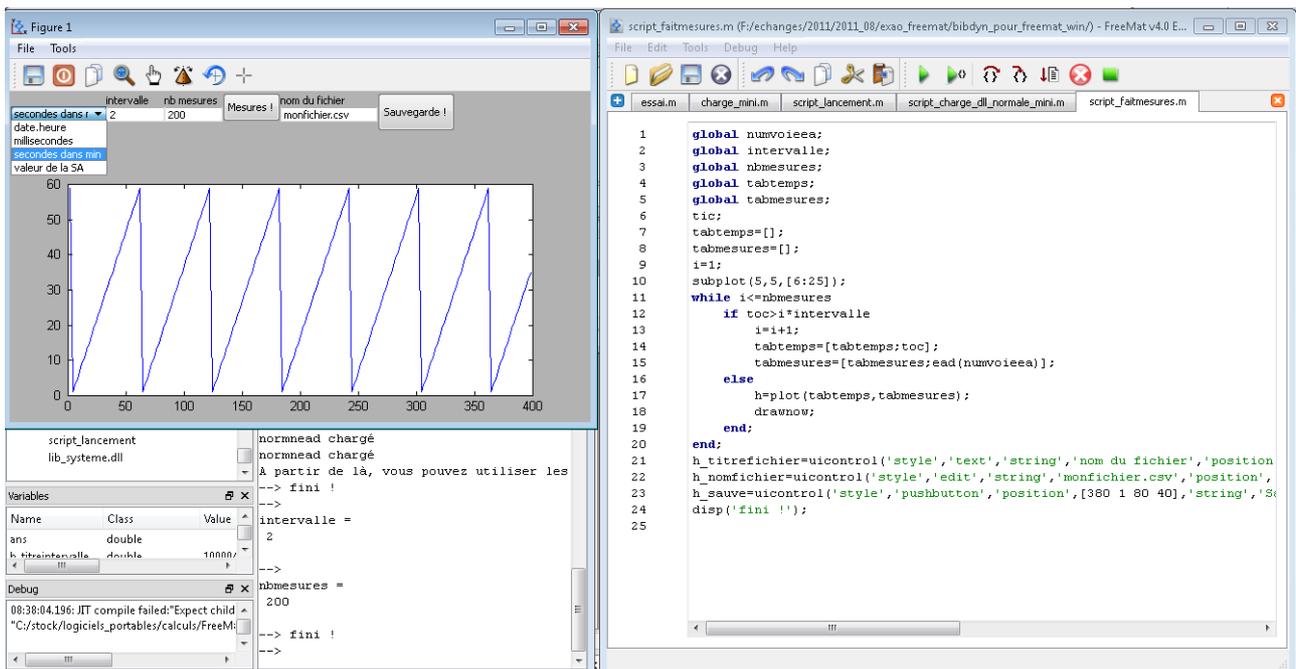
La version 4.0, qui est la version actuelle (été 2011), est censée être plus puissante que la version 3.6, mais elle a quelques problèmes.

- D'une part, la fonction input (entrée de données au clavier) n'accepte pas les valeurs numériques, ce qui est gênant pour un logiciel de calcul numérique. La solution est facile, modifier le fichier input.m (dans le répertoire toolbox/io), par exemple en le remplaçant par celui de la version 3.6.
- D'autre part, lorsqu'un programme doit tracer des graphes à intervalles de temps régulier, le traçage ne se fait qu'à la fin du programme, ce qui ne permet ni de faire des animations, ni de tracer le graphique de mesures scientifiques pendant le déroulement de l'expérience. La solution est moins évidente, car cela ne semble pas dépendre de fonctions .m.
- **Sous Windows, le mieux est donc d'utiliser la version portable de Freemat**, disponible sur http://portableapps.com/apps/education/freemat_portable, chez qui ces deux bogues sont rectifiés. Il est aussi possible d'utiliser la version 3.6, qui fonctionne beaucoup mieux que la version 4.0 non portable.

Freemat existe aussi pour Linux, avec les mêmes caractéristiques que pour Windows, avec les mêmes inconvénients, plus quelques autres.

- Les versions récentes de Linux, comme Ubuntu 10 ou 11, peuvent charger automatiquement Freemat à partir de leurs dépôts de logiciels : il faut pour cela cocher la case "logiciels maintenus par la communauté (universe)" dans la boîte de dialogue des dépôts du logiciel Aptitude. Mais ce chargement automatique aboutit à la version 4, qui, comme sous Windows, ne peut pas faire d'animation, en traçant la courbe expérimentale peu à peu. De plus, cette version 4 ne peut pas utiliser les bibliothèques dynamiques : lorsqu'on essaie d'utiliser la fonction "import", il apparaît un message disant que Freemat n'a pas été compilé pour pouvoir utiliser les bibliothèques dynamiques. Le cas n'est pas forcément désespéré car la version 3.6, que l'on peut quand même utiliser, en est capable : voir le paragraphe 3 pour **utiliser quand même Freemat 3.6 sous Linux**

Le système proposé ici repose sur l'emploi de bibliothèques dynamiques spécifiques des divers appareils de mesure. Ces petits fichiers jouent le rôle de "pilotes d'appareils de mesure" (= "drivers") : le programme rédigé en langage Freemat appelle les fonctions de ces bibliothèques dynamiques. Ainsi, potentiellement, ils rendent Freemat capable de converser avec tous les divers appareils de mesure.. C'est le système Mensurasoft, qui permet d'utiliser un grand nombre de langages informatiques (C++, Pascal, Basic, Scilab, Freemat, Logo, Python...), sous divers systèmes d'exploitation (MS-Windows, Linux...).



1 Freemat pour Windows peut utiliser les bibliothèques dynamiques

Le système Mensurasoft est (relativement) simple, car il est destiné à permettre l'utilisation des pilotes d'appareils de mesure par divers logiciels et langages informatiques, qui n'avaient pas été conçus spécialement pour lui. On peut trouver divers renseignements sur <http://sciencexp.free.fr>.

Les langages informatiques "classiques" permettent d'appeler (relativement) simplement les bibliothèques dynamiques. Par contre, Freemat, qui est un logiciel de calcul numérique, a quelques

lacunes dans la gestion des bibliothèques dynamiques et ne semble pas gérer les fonctions devant envoyer une chaîne de caractères.

1.1 Les bibliothèques dynamiques de Freemat ne travaillent bien qu'avec les valeurs numériques

- Les bibliothèques de Freemat ont un passage de paramètres de type "cdecl" (il en existe plusieurs autres types, dont le plus fréquent est "stdcall"). Le système Mensurasoft est prévu pour les deux types.
- Il n'y a pas de difficulté particulière pour les fonctions renvoyant des valeurs numériques : les fonctions purement numériques du système Mensurasoft sont appelables directement par Freemat. Ce sont les fonctions les plus importantes.
- Elles ne permettent pas de renvoyer directement des chaînes de caractères. De telles fonctions sont utilisées par le système Mensurasoft pour donner le nom des différentes fonctions travaillant avec les valeurs numériques. Pour pouvoir quand même utiliser les fonctions de Mensurasoft travaillant avec des chaînes de caractères, la solution proposée ici est d'utiliser un programme exécutable extérieur, capable de lire les fonctions de chaînes de caractères, et d'en renvoyer le résultat à Freemat par la fonction system.

1.2 Comment Freemat appelle ses bibliothèques dynamiques travaillant sur des nombres

C'est simple, beaucoup plus simple que pour beaucoup de langages informatiques.

Par exemple, l'instruction ci-dessous

```
import('monappareil.dll', 'cead', 'cead', 'double', 'int16 n');
```

va chercher dans le fichier monappareil.dll une fonction nommée cead, qui renvoie une valeur de type réel en double précision, et qui a un paramètre de type entier codé sur 16 bits. De telles fonctions existent dans le système Mensurasoft, et permettent de lire les entrées analogiques et binaires, et de commander les sorties analogiques et binaires.

1.3 Utilisation des fonctions de chaînes de caractères par Freemat grâce à un programme exécutable

Un tel adaptateur a pour fonction essentielle de permettre à Freemat d'utiliser les fonctions du système Mensurasoft qui travaillent avec des chaînes de caractères : titre de l'appareil, détail de l'appareil, nom des entrées et sorties analogiques et binaires.

Freemat possède la fonction "system" qui exécute la commande qu'on lui passe en paramètre, et qui récupère l'ensemble des chaînes de caractères qui sont renvoyées par la commande. Par exemple, "x=system('dir')" envoie dans la variable x le résultat de la commande dir, c'est à dire la liste des fichiers et répertoires.

On a donc réalisé en FreePascal un programme pdll.pas (compilé en pdll sous Linux et pdll.exe sous Windows) qui fonctionne en ligne de commande. Son premier paramètre est le nom de la bibliothèque dynamique à utiliser, son deuxième paramètre est le nom de la fonction de cette bibliothèque à appeler, et éventuellement les troisième et quatrième paramètres sont les paramètres

de cette fonction.

Par exemple, "pdll lib_systeme.dll titre" renvoie le titre du pilote ; "pdll lib_systeme.dll nead 2" renvoie le nom de l'entrée analogique numéro 2.

L'exemple ci-dessous ne possède que les fonctions stdtitre, stddetail, stdnead. Un programme complet aurait aussi les fonctions des noms des entrées binaires, sorties analogiques et sorties binaires.

```

program pdll;
{$APPTYPE CONSOLE}
uses
  SysUtils, dynlibs;

var l:thandle;
    repchar:array[0..80] of char;
    stdnead:function(n:word):pchar;
stdcall;
    var
stddetail, stdtitre:function:pchar;stdcall;
    var
chemindll, nomfonction, chparam1, chparam2:string;
    var valparam1:double;
    var valparam2:double;
    var i:integer;
begin
    valparam1:=11;valparam2:=22;
    chemindll:=paramstr(1);
// writeln('chemindll:',chemindll);
    nomfonction:=paramstr(2);
//writeln('nomfonctin:',nomfonction);
    if paramcount>=3 then
        valparam1:=strtofloat(paramstr(3));
        //writeln('valparam1:',valparam1);
        if paramcount>=4 then
            valparam2:=strtofloat(paramstr(4));
            //writeln('valparam2:',valparam2);
            strpcopy(repchar,chemindll);
            @stdnead:=nil;@stddetail:=nil;@stdtitre:=nil;
            l:=safeloadlibrary(repchar);
            @stddetail:=getprocedureaddress(L,'stddetail');
            @stdtitre:=getprocedureaddress(L,'stdtitre');
            @stdnead:=getprocedureaddress(L,'stdnead');
            if nomfonction='stdtitre' then
                writeln(stdtitre);
            if nomfonction='stddetail' then
                writeln(stddetail);
            if nomfonction='stdnead' then
                writeln(stdnead(round(valparam1)));
            //writeln('xxxxfini');
            //freelibrary(L);
            unloadlibrary(L);
        end.
    end.

```

1.4 Des fonctions Freemat pour communiquer avec cet adaptateur

La fonction Freemat nead, contenue dans le fichier nead.m appelle le programme exécutable pdll que l'on suppose dans le répertoire de travail, avec le nom de la bibliothèque dynamique (que l'on suppose aussi dans le répertoire de travail), ainsi que le nom de la fonction (stdnead), et le numéro de la voie de mesure.

```

function res=nead(i)
    global nompilote
    if length(computer)==length('PCWIN')
        chainecommande=['pdll.exe
',nompilote,' stdnead ',num2str(i)'];
    else
        chainecommande=['./pdll ./',nompilote,'
stdnead ',num2str(i)'];
    end
res=system(chainecommande){1};
end

```

Ce dispositif fonctionne correctement, mais beaucoup plus lentement que les bibliothèques dynamiques directes : il faut que Freemat appelle un fichier exécutable, qui lui-même charge le fichier de la bibliothèque dynamique, qui fait la mesure, et envoie le résultat.

Là encore, pour gérer les sorties analogiques et les entrées et sorties binaires, il faudrait d'autres fonctions.

2 Réalisation d'un (mini)logiciel d'expérimentation en Freemat

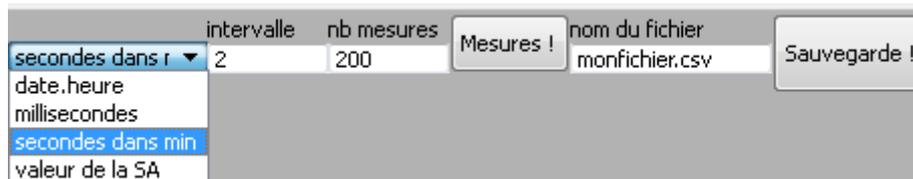
2.1 Mise en place d'éléments de dialogue dans la fenêtre graphique

Pour faire les réglages de l'expérience, on pourrait utiliser la méthode ancienne d'affichage de messages et de frappe de caractères dans la console Freemat, mais on peut utiliser une méthode plus moderne : parmi les nombreuses fonctions Freemat, certaines permettent de mettre des boutons, des lignes de saisie ou diverses boîtes de saisie.

Les réglages nécessaires sont (après le chargement de la bibliothèque dynamique de mesure, que l'on suppose choisie préalablement) :

- choisir la voie de mesure (en supposant qu'on utilise un appareil ayant plusieurs voies possibles, comme par exemple Arduino, Orphy portable, Jeulin ESAO...)
- choisir l'intervalle entre les points de mesure, c'est à dire un nombre de secondes.
- choisir le nombre total de points de mesure à réaliser : la durée de l'expérience sera le produit (intervalle)*(nombre de points).

Comme les éléments de dialogue sont en haut de la fenêtre graphique, il ne faut pas que le traçage du graphe les détruise. C'est pourquoi le traçage du graphe est limité à la partie inférieure de la fenêtre graphique, par l'instruction subplot.



```

global nbmesures;global intervalle;global numvoieea;global nompilote;
%lancement de mesures
nbmesures=10; intervalle=1; numvoieea=2;
if length(computer)==length('PCWIN')
    dir *.dll
else
    dir *.so
end
nompilote=input('Entrez le nom du pilote (bibliothèque dynamique : dll ou so) : ','s');
import(nompilote,'cead','cead','double','uint16 i');
i=1;
ch_choixvoie=[nead(0)];
while length(nead(i))>0
    ch_choixvoie=[ch_choixvoie,'|',nead(i)];
    i=i+1;
end
plot(0,0);
h_titrechoixvoie=uicontrol('style','text','position',[1 1 100 15],'string','voie :');
list_choixvoie= uicontrol('style','popupmenu','string',
ch_choixvoie,'value',numvoieea+1,'position', [1 15 100
15],'callback','numvoieea=get(list_choixvoie,'value')-1');
h_titreintervalle=uicontrol('style','text','string','intervalle','position',
[100,1,100,15]);
h_intervalle=uicontrol('style','edit','string',num2str(intervalle),'position',
[100,16,100,15],'callback','intervalle=eval(get(h_intervalle,'string'))');
h_titrenbmesures=uicontrol('style','text','string','nb mesures','position',
[200,1,100,15]);
h_nbmesures=uicontrol('style','edit','string',num2str(nbmesures),'position',
[200,16,100,15],'callback','nbmesures=eval(get(h_nbmesures,'string'))');
h_bouton_lancemesure=uicontrol('style','pushbutton','string',
'Mesures !','callback','script_faitmesures','position', [300 1 100 30]);

```

2.2 Faire les mesures

Cette option est déclenchée par le cliquage du bouton "Mesures !".

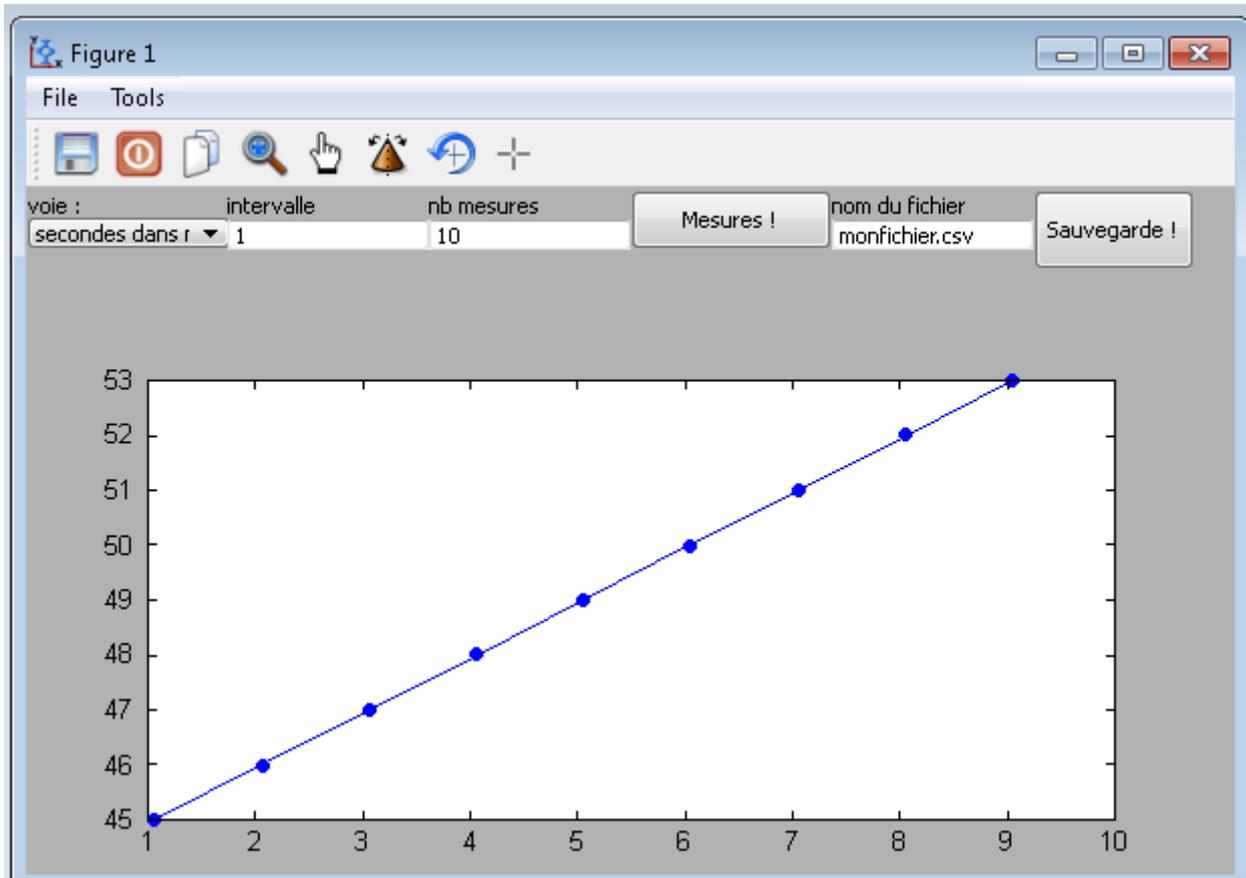
```

global numvoieea;
global intervalle;
global nbmesures;
global tabtemps;
global tabmesures;
tic;
tabtemps=[];
tabmesures=[];
i=1;
subplot(5,5,[6:25]);
while i<=nbmesures
    if toc>i*intervalle
        i=i+1;
        tabtemps=[tabtemps;toc];
        tabmesures=[tabmesures;ead(numvoieea)];
    else
        h=plot(tabtemps,tabmesures,'o-');
        drawnow;
    end;
end;
end;

```

2.3 Sauvegarder les mesures dans un fichier

On choisit le nom du fichier dans la ligne de saisie, puis on clique sur "Sauvegarde !"



3 Utiliser quand même Freemat 3.6 sous Linux ?

Comme on l'a vu en introduction, les versions modernes de Linux sont trop modernes : elles vont chercher directement dans leur dépôt la version la plus récente de Freemat, c'est à dire la version 4 (en août 2011), qui d'une part ne permet pas d'appeler les bibliothèques dynamiques, et d'autre part ne permet pas de tracer les points peu à peu (avec l'instruction `drawnow`).

Si on prend une version trop ancienne de Linux, Freemat n'est pas chargeable.

Ubuntu 9 offre l'idéal : le chargement automatique de Freemat 3.6, qui est parfait : il utilise les bibliothèques dynamiques, et l'instruction `drawnow` permet de tracer le graphique peu à peu (et l'instruction `input` fonctionne correctement). Pour pouvoir charger Freemat, lancer le gestionnaire de paquets Synaptic (par Système/Administration), puis, dans Configuration/dépôts/Logiciels Ubuntu, cocher la case "Logiciels libres maintenus par la communauté (universe)". Fermer la boîte de dialogue, cliquez sur Recharger : normalement Freemat 3.6 doit être dans la liste des logiciels disponibles.

Que faire si on a une version plus récente d'Ubuntu ? On peut modifier la liste des dépôts pour utiliser ceux de Ubuntu 9. La méthode peut être la suivante :

- ouvrir le gestionnaire de paquets Synaptic. Dans le menu de configuration, choisir "dépôts". Dans le premier onglet, il ne faut pas cocher "Logiciels libres maintenus par la communauté (universe)", car cela chargerait Freemat 4.0. Laissez cette case décochée, et allez dans le deuxième onglet

"Autres logiciels".

Dans ce deuxième onglet, ajoutez une nouvelle source, en mettant les réglages suivants : URI : <http://archive.ubuntu.com/ubuntu> | distribution : karmic | composants : universe.

(karmic signifie qu'on chargera les paquets pour Ubuntu 9).

Validez, puis fermez la boîte de dialogue des dépôts, cliquez sur Recharger, attendez que la mise à jour de la liste des paquets disponible se fasse par internet, et finalement, normalement, Freemat 3.6 doit être disponible au téléchargement dans la liste des paquets.

4 Comment améliorer ce programme ?

Dans le script d'acquisition de données, on peut faire la moyenne de mesures rapprochées, plutôt que de prendre un point de mesure toutes les n secondes.

On pourrait aussi faire des mesures sur deux ou plusieurs voies, si l'appareil de mesure et le pilote le permettent.

En utilisant les autres fonctions de sorties analogiques, et d'entrées et sorties binaires, on pourrait faire une régulation d'une grandeur, ou ne déclencher les mesures que si une valeur dépasse un seuil.

Au lieu de faire des mesures toutes les n secondes, on pourrait faire des mesures toutes les n millisecondes, en faisant une sorte d'oscilloscope. Pour ceci, il ne faudrait pas retracer le graphique à chaque nouvelle mesure, mais seulement après la prise de l'ensemble des points de mesure d'une série.

Copyright 2012 Pierre Dieumegard pierre.dieumegard@free.fr

Le système Mensurasoft fait partie des logiciels libres.

Les pilotes sous forme de bibliothèques dynamiques sont des logiciels libres au sens LGPL (<http://www.gnu.org/licenses/lgpl.txt>)

Les programmes d'application, ici Mensurasoft-Freemat sont des logiciels libres au sens GPL (<http://www.gnu.org/licenses/gpl.txt>)