Pierre Dieumegard
pierre.dieumegard@free.fr

Extension des pilotes Mensurasoft pour capture rapide

1 Position du problème

Le système Mensurasoft de pilotes d'appareils de mesure permet de faire un seul logiciel pour faire des mesures, des commandes ou des régulations avec un grand nombre d'appareils. Dans le système de base, c'est le logiciel qui commande les prises de mesure, une par une. Dans la pratique, ceci ne permet pas de faire des mesures où l'intervalle est très court (de l'ordre de la milliseconde ou de la microseconde) parce qu'il faut un certain temps pour que le programme principal donne l'ordre à l'appareil de faire la mesure, reçoive la mesure et la traite, et redemande ensuite de faire une nouvelle mesure. Ceci ne permet pas d'exploiter la rapidité de certaines interfaces de mesure, qui pourraient faire des mesures beaucoup plus rapides (intervalle de l'ordre de la microseconde).

On peut donc essayer de faire des fonctions de "capture rapide" pour certains appareils. Lors de l'appel à une telle fonction, l'appareil ferait très rapidement de nombreuses mesures (par exemple 200 mesures espacées de 20 microsecondes), puis renverrait l'ensemble des résultats au programme principal.

Certaines de ces interfaces peuvent faire simultanément une série de mesures sur deux ou plusieurs voies, alors que d'autres ne peuvent le faire que sur une voie (ou seulement quelques combinaisons de voies sont possibles). Donc les possibilités sont différentes selon les appareils. **Dans un premier temps, on ne cherche à faire que les fonctions pour une voie de mesure : les fonctions pour plusieurs voies de mesure simultanées seront à concevoir ensuite, en harmonie avec les fonctions sur une seule voie.**

1.1 Quels appareils sont concernés ?

Ceux reliés à l'ordinateur par une connexion "lente", telle qu'une prise RS232 ou une prise USB, et qui peuvent faire des mesures plus rapidement que la vitesse de transmission des informations.

Exemples : toute la gamme Orphy de Micrelec, les consoles Eurosmart, ExpEyes, Arduino..(et d'autres, dont malheureusement les codes de commande ne sont pas publiés : ESAO4-USB, Visio de Jeulin).

1.2 Est-il possible de faire des fonctions communes pour toutes ces interfaces ?

Toutes ces interfaces ont en commun de déclencher une série de mesures lorsqu'elles en reçoivent l'ordre, puis d'envoyer vers l'ordinateur les résultats (ou une partie des résultats) de cette série de mesure lorsqu'elles en reçoivent l'ordre. Pendant la prise de mesure (très rapide ici), le programme principal ne peut pas intervenir : tout doit être géré par l'appareil lui-même.

Certaines de ces interfaces peuvent ne déclencher la série de mesure que dans certaines circonstances (lorsque la valeur dépasse un seuil, par exemple), certaines peuvent générer un signal pendant la prise de mesure (signal périodique, passage d'une broche à l'état haut ou à l'état bas en début de prise de mesure,etc). Les caractéristiques de ces opérations doivent être envoyées à l'appareil avant le déclenchement de la série de mesures.

Comme les possibilités sont très différentes selon les appareils, il ne faut pas que ces possibilités

soient gérées par le logiciel principal, mais plutôt par le pilote de l'appareil. On peut donc imaginer une fonction qui l'ordre à l'appareil de se préparer à faire les mesures selon telles modalités, suivie d'une fonction qui donne l'ordre à l'appareil de faire les mesures, puis de renvoyer ces mesures vers le programme principal.

1.3 Faut-il faire des pilotes spéciaux ou des logiciels spéciaux ?

Pour les pilotes, le mieux est d'utiliser ceux du système Mensurasoft normal, auxquels on ajoute les fonctions spéciales de capture rapide. Ainsi, les mêmes pilotes pourront aussi faire les mesures lentes.

Par contre, il est souhaitable de faire des logiciels spéciaux pour ces fonctions de mesure rapide. Les logiciels de mesure lente (MGW32, Mensurasoft-LZ et Mensurasoft-PB) peuvent faire les mesures simultanément sur 3 appareils différents, en commandant une sortie analogique éventuellement sur un quatrième appareil. Ces possibilités seraient inopérantes pour des mesures rapides du type capture : il n'est donc pas souhaitable de modifier ces logiciels, et il vaut mieux en faire de nouveaux, spécialement conçus dans ce but..

"Oscapture" est un logiciel de démonstration dont quelques copies d'écran sont montrées ci-après.

2 Ce qui marche correctement : fonctions pour faire la série de mesure

Comme pour le reste du système Mensurasoft, chaque fonction numérique (qui renvoie la série de mesure au programme principal) est associée à une fonction de nom (qui renvoie le nom de la voie de mesure) avec le préfixe n.

Comme pour le reste du système Mensurasoft, les fonctions existent en deux versions selon le passage des paramètres : stdcall (préfixe std-), et cdecl (préfixe c).

Les autres fonctions numériques du système Mensurasoft renvoient des valeurs numériques de type "entier" (sur 32 bits) ou de type "réel" (de type double-précision). Ici, comme il faut renvoyer des tableaux de valeurs numériques, le mieux est d'envoyer ces tableaux sous forme de chaînes de caractères, où chaque ligne sera une mesure. Comme pour le reste du système Mensurasoft, ces chaînes seront de type "pointeur de chaîne à zéro terminal".

2.1 Fonctions stdcapture1t et ccapture1t

Le "1" signifie que les mesures seront faites sur une voie ; le t signifie que le tableau de valeurs renvoyées (sous forme d'une grosse chaîne de caractères, avec les sauts de ligne codés par chr(13)+chr(10)) comprendra le temps dans le premier champ, séparé de la valeur mesurée par une tabulation (chr(9)).

Elles ont trois paramètres d'entrée : le numéro de la voie (entier), le nombre de mesures à faire (entier), et l'intervalle entre deux mesures, en microsecondes (double). Même si la majorité des appareils ne peut pas faire de mesures avec une période inférieure à la microseconde, il faut prévoir le cas, c'est pourquoi le paramètre de l'intervalle est un nombre réel.

Les déclarations seront donc :

- en FreeBasic

```
public function ccapture1t cdecl alias "ccapture1t" (byval numvoie as integer, byval n as integer,
byval intervalle as double) as zstring pointer export
public function stdcapture1t stdcall alias "stdcapture1t" (byval numvoie as integer, byval n as
integer, byval intervalle as double) as zstring pointer export
```

- en PureBasic

```
ProcedureDLL .s stdcapture1t(numvoie .i, nbmesures .i , intervalleus .d)
```

```
ProcedureCDLL .s ccapture1t(numvoie .i, nbmesures .i , intervalleus .d)
```

- en FreePascal ou Lazarus

```
function stdcapture1t(numvoie:longint ; n : longint;intervalle:double):pchar;stdcall;export;alias :
'stdcapture1t';
function ccapture1t(numvoie:longint ; n : longint;intervalle:double):pchar;cdecl;export; alias :
'ccapture1t';
```

2.2 Fonctions stdncapture1t et cncapture1t

Elles ont un paramètre d'entrée (entier) qui est le numéro de la voie, et renvoie sous forme de chaîne le nom de cette voie. Comme dans le reste du système Mensurasoft, si le nom est une chaîne vide (de longueur nulle), cela signifie que la voie en question n'existe pas.

- en FreeBasic :

```
public function stdncapture1t stdcall alias "stdncapture1t" (byval numvoie as integer) as zstring
pointer export
public function cncapture1t cdecl alias "cncapture1t" (byval numvoie as integer) as zstring pointer
export
```

- en PureBasic :

```
ProcedureDLL .s stdncapture1t(n .i)
ProcedureCDLL .s cncapture1t(n .i)
```

- en FreePascal/Lazarus :

```
function  stdncapture1t (numvoie:longint):pchar;stdcall;export;
function  cncapture1t (numvoie:longint):pchar;cdecl;export;   alias:'cncapture1t';
```

2.3 Logiciel de démonstration Oscapture (pour Windows et Linux, réalisé en Lazarus)

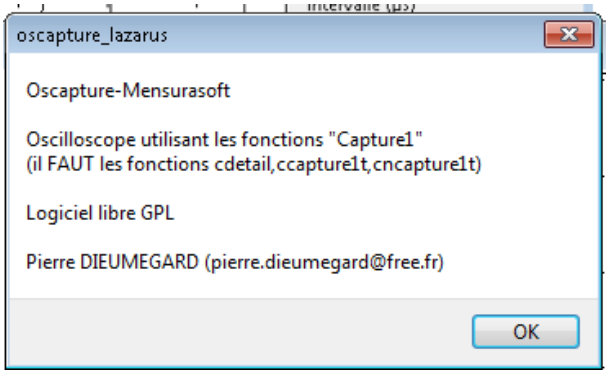

Illustration 1: boite d'aide de Oscapture

C'est un logiciel simple. Contrairement à Mensurasoft-PB et Mensurasoft-LZ, il n'est pas multilingue, n'a pas de fichiers de configuration, ne peut pas faire de transformations de variables...

Il peut charger les fonctions capture1t d'un pilote. Par défaut, il fait 100 mesures espacées de 400µs, mais ces valeurs sont modifiables. Lorsqu'on clique sur "Capture ?", une série de mesures est réalisée et affichée dans le graphique. Lorsqu'on coche "capture continue", une série de mesure est faite et affichée toutes les secondes.

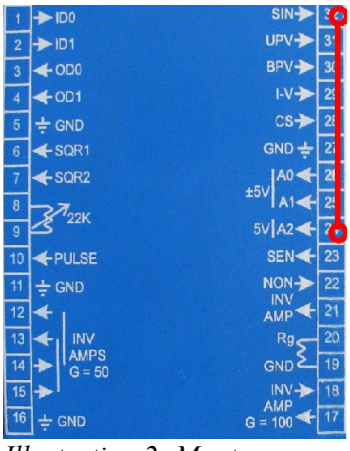

Illustration 2: Montage d'ExpEyes : SIN relié à A2

Ensuite, en cliquant sur "Sauve dans fichier", on peut enregistrer les données dans un fichier .csv, et lorsqu'on clique sur "Copie presse-papier", elles sont copiée en mémoire, et récupérables dans un tableur par "coller".

(on peut noter qu'au dessus du bouton "Capture", il y a de l'espace : il est prévu d'y mettre un bouton pour régler les caractéristiques supplémentaires de l'acquisition : voir ci-après).

Le bouton "Aide" provoque l'apparition d'une petite fenêtre indiquant les fonctions nécessaires dans le pilote : cdetail, ccapture1t, cncapture1t.

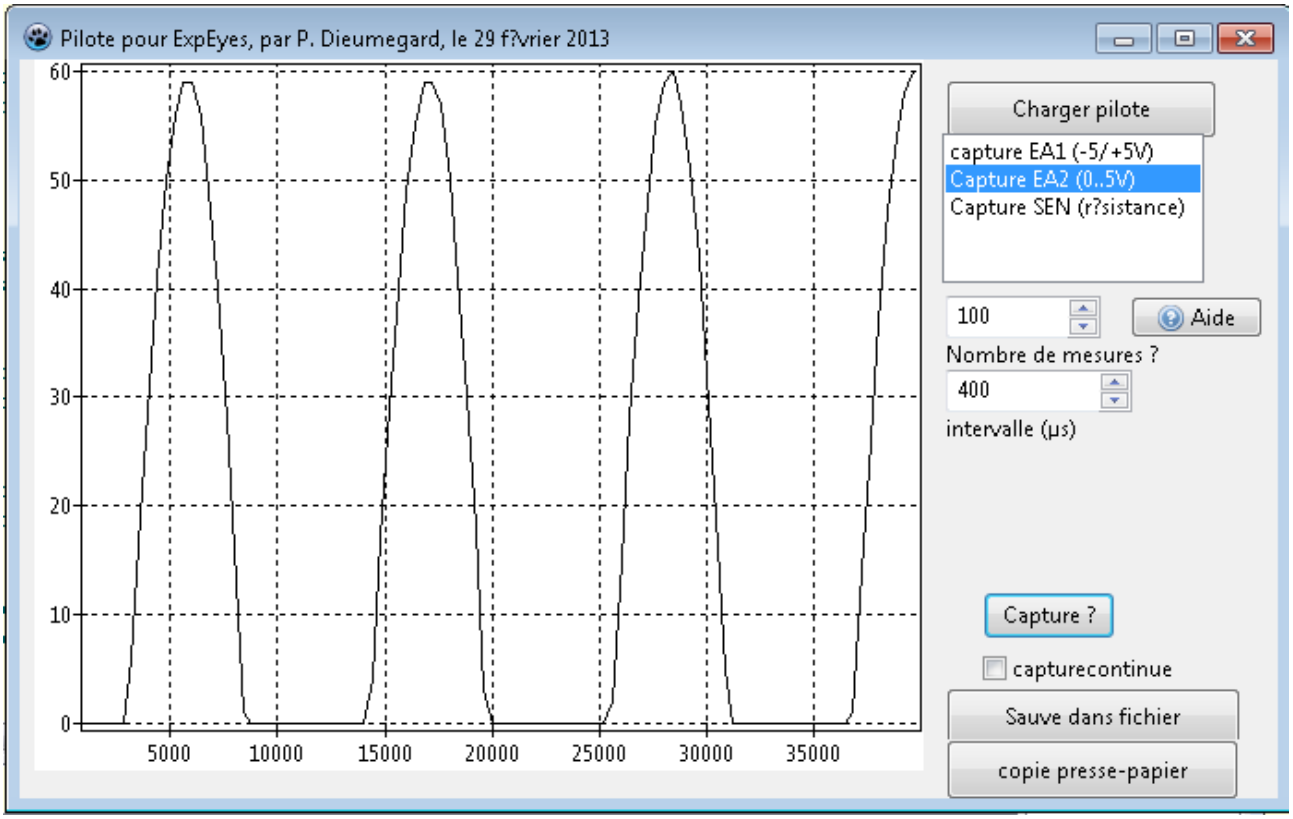

Illustration 3: Acquisition par Oscapture sur Expeyes (SIN relié à A2)

Donc, à l'aide de ces deux fonctions, ccapture1t et cncapture1t, on peut faire un oscilloscope de base, pouvant étudier les phénomènes périodiques. On peut considérer l'affaire comme résolue, non seulement pour Expeyes, mais aussi pour la famille Orphy, et Arduino, et probablement pour Eurosmart.

Que faire pour augmenter la puissance de cet oscilloscope, en particulier pour ne déclencher l'acquisition que lorsqu'un seuil est franchi, ou bien provoquer un changement de tension lors du déclenchement de l'acquisition ?

3 Augmenter la puissance par fixation des caractéristiques de l'acquisition : fonctions fixcapture1 et calcapture1

Les diverses interfaces ont des possibilités différentes.

Voici deux exemples qui n'ont pas exactement les mêmes possibilités :

- ExpEyes peut déclencher l'acquisition seulement si une entrée digitale passe à l'état haut, ou bas, ou montant, ou descendant. Il peut aussi faire passer une sortie digitale à l'état haut ou à l'état bas au déclenchement de l'acquisition.

- Orphylab peut déclencher l'acquisition sur un front montant ou descendant, avec ou sans précourse, avec un seuil réglable.

Il n'est donc pas possible que le logiciel principal puisse prévoir tous les cas possibles de déclenchement dans certaines circonstances, avec émission d'un signal de telle ou telle forme.

Par conséquent, les réglages possibles doivent être sous la responsabilité du pilote, de même que pour les imprimantes modernes, le pilote a une boite de dialogue où l'on peut choisir d'imprimer

recto-verso, ou en livret, en couleurs ou en noir et blanc, etc.

On peut imaginer deux fonctions supplémentaires (avec les deux variantes, stdcall et cdecl, comme pour les autres fonctions du système Mensurasoft) :

3.1 fixcapture1, pour fixer les caractéristiques de la capture future

Le logiciel principal enverrait une chaîne de caractères au pilote, qui déterminerait alors les caractéristiques supplémentaires de l'acquisition future. Par exemple "SIG>15" indiquerait qu'il faudrait lancer l'acquisition uniquement si le signal est supérieur à 15. Envoyer une chaîne vide indiquerait l'arrêt de ces conditions, et le renvoi immédiat de la chaîne de mesure.

Cette fonction renverrait un nombre entier, 0 si tout s'est bien passé (si la chaîne est correcte, et si les caractéristiques sont possibles), et un nombre non nul, par exemple -1 s'il y a des problèmes (si la chaîne n'est pas interprétable, ou si les paramètres sont hors-limites).

- en FreeBasic :

```
public function cfixcapture1 cdecl alias "cfixcapture1" (byval ch as zstring pointer) as integer
export
public function stdfixcapture1 stdcall alias "stdfixcapture1" (byval ch as zstring pointer) as
integer export
```

- en PureBasic :

```
ProcedureCDLL .i cfixcapture1(ch .s)
ProcedureDLL .i stdfixcapture1(ch .s)
```

- en FreePascal/Lazarus :

```
function cfixcapture1(ch:pchar):longint; cdecl;export;alias:'cfixcapture1';
function stdfixcapture1(ch:pchar):longint ; stdcall;export;alias : 'stdfixcapture1';
```

3.2 calcapture1, pour une boite de dialogue des caractéristiques de la capture future ("cal" comme calibration)

Dans le système Mensurasoft normal, la fonction "calibration" peut déjà exister, même si elle est peu utilisée : elle permet l'ouverture d'un dialogue pour une calibration de l'appareil (réglage du blanc d'un spectophotomètre, par exemple).

Ici, la fonction calcapture1 recevrait en paramètre une chaîne de caractères décrivant les caractéristiques actuelles de capture. Elle renverrait la chaîne de caractères modifiée par l'utilisateur (qui aurait coché par exemple la case "déclenchement par valeur supérieure à" et fixé la valeur numérique à 15).

- en FreeBasic :

```
public function ccalcapture1 cdecl alias "ccalcapture1" (byval ch as zstring pointer) as zstring
pointer export
public function stdcalcapture1 stdcall alias "stdcalcapture1" (byval ch as zstring pointer) as
zstring pointer export
```

- en PureBasic :

```
ProcedureCDLL .s ccalcapture1(ch .s);
ProcedureDLL .s stdcalcapture1(ch .s);
```

- en FreePascal/Lazarus :

```
function ccalcapture1(ch:pchar):pchar;cdecl;export;alias:'ccalcapture1';
function stdcalcapture1(ch:pchar):pchar;stdcall;export;alias:'stdcalcapture1';
```

3.3 Pourquoi deux fonctions différentes, calcapture1 et fixcapture1 ?

Bien sûr, elles sont destinées à être associées : déterminer la chaîne de calibration par calcapture1 doit être suivi par utiliser cette chaîne par fixcapture1.

Il est quand même souhaitable de séparer les deux fonctions.

- Des programmes plus élaborés de Oscapture pourront dans le futur utiliser des fichiers de configuration (comme Mensurasoft-LZ et Mensurasoft-PB), de façon que l'utilisateur ait

directement les bons réglages pour l'expérience souhaitée. Donc la chaîne de réglage serait lue dans le fichier de configuration et envoyée vers l'appareil par fixcapture1, sans qu'il y ait affichage d'une boite de dialogue.

- On peut faire des programme qui changent progressivement les réglages, et enregistrent le résultat. Par exemple un programme peut faire une boucle "pour stimulation variant de 0 à 5 V par pas de 0,05 V, mesurer la réponse, et tracer la courbe du maximum de réponse". Dans ce cas, il faut faire appel à la fonction fixcapture1 à chaque tour de boucle, avec des paramètres un peu différents, sans faire appel à la boite de dialogue.

Quand seront disponibles ces fonctions supplémentaires, pour faire de vraies expériences ?

Normalement, dans le courant 2013 (au moins pour ExpEyes et Orphy).

"Il n'y a qu'à" faire les fonctions correspondantes pour les pilotes qui existent déjà, et ajouter quelques lignes au programme Oscapture (il y a de la place pour un ou deux boutons au dessus de "Capture ?").

Mais afin de ne pas s'engager dans des impasses, il serait souhaitable que ces programmes et pilotes soient essayés par divers utilisateurs. Il serait aussi souhaitable que plusieurs programmeurs essaient de faire des programmes et/ou des pilotes, pour voir les difficultés...

Pierre Dieumegard

pierre.dieumegard@free.fr