

# MensuRaspberry : le système Mensurasoft pour le Raspberry pi

(pierre.dieumegard@free.fr)

Mensura, ou Mensurasoft, est un système de pilotes d'appareils de mesure permettant à tous les ordinateurs, avec tous les systèmes d'exploitation (au moins MS-Windows et Linux, et probablement Mac), avec tous les langages de programmation (ou presque) de faire des mesures sur tous les appareils et interfaces de mesure connectables à l'ordinateur (ou presque).

Raspberry pi est un nano-ordinateur, tenant sur une plaque électronique de quelques centimètres de côté. La version la plus récente actuellement (juillet 2016) est le Raspberry pi 3, avec 4 prises USB, une connexion réseau et une connexion wifi, ainsi qu'une connexion vidéo HDMI et une connexion audio, comme beaucoup d'autres ordinateurs. De plus, il a 40 broches permettant de connecter des composants électroniques (les premiers Raspberry pi n'avaient que 26 broches).

Plusieurs systèmes d'exploitation sont disponibles pour Raspberry pi ; celui recommandé est Raspbian, un dérivé de Debian : c'est donc celui-ci qui sera décrit ici.

Pour exploiter facilement les broches d'entrée-sortie, il faut charger wiringPi par la série d'instructions :

```
sudo apt-get install git-core
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

## 1 Résumé du système Mensurasoft

De même que tous les modèles d'imprimantes sont utilisables par n'importe quel logiciel de bureautique grâce à un système de « pilotes d'imprimantes », ici, chaque modèle d'appareil de mesure (ou d'actionneur, ou d'interface d'acquisition de données) a un « pilote d'appareil de mesure » qui lui est spécifique, et qui est sous la forme d'une « bibliothèque dynamique » (fichiers .dll de Windows et .so de Linux).

Ces pilotes d'appareils de mesure sont eux-mêmes appelés par le logiciel principal (= logiciel d'application), qui peut être simplement quelques lignes dans un langage de programmation quelconque, ou bien un logiciel complexe, avec menus déroulants, boîtes de dialogue, visualisation graphique des mesures, traitements mathématiques...

## 2 Adaptation des logiciels au Raspberry

Comme le Raspberry n'utilise pas le même type de microprocesseur que les ordinateurs « normaux » (ARM6 au lieu de 386), il faut que les logiciels et pilotes soient spécialement adaptés. Même si le système d'exploitation est un « Debian », on ne peut pas utiliser directement les programmes ou bibliothèques qui ont été compilés pour un ordinateur normal.

## 2.1 Réalisation des pilotes

Il faut obligatoirement un langage de programmation permettant la compilation pour faire des bibliothèques dynamiques. Par défaut, le Raspberry a un compilateur C et C++ (gcc, g++).

On peut assez facilement charger sur internet FreePascal (et Lazarus). Malheureusement (jusqu'ici) ces langages ne font pas des bibliothèques dynamiques fonctionnelles (un problème de réglage des options de compilation?)

Le langage utilisé ici est FreeBasic, disponible sur internet :

(<http://www.freebasic.net/forum/viewtopic.php?t=21433>)

Après décompaction du fichier fbc-0.91.0-pi-raspbian.tar.gz, il faut aller dans le répertoire en question à partir de la console, et frapper `sudo ./install.sh -i`

Après cette installation, il manque libncurses, c'est pourquoi il faut faire `sudo apt-get install libncurses5-dev`

Un bon éditeur pour programmer en FreeBasic est Geany (dans le menu Programmation)

Pour faire des bibliothèques dynamiques avec FreeBasic, il faut mettre `-dll` dans les options de compilation. Pour utiliser wiringPi, il faut l'indiquer aussi dans les options de compilation par `-l wiringPi` (menu Construire / Définir les commandes de construction, Compiler, mettre `fbc -dll -l wiringPi "%f"`).

FreeBasic met une « décoration » aux noms de fonctions de type stdcall. Lorsqu'on veut utiliser des fonctions stdcall avec un seul paramètre, il faut déclarer ces fonctions de type « pascal » au lieu de stdcall ; lorsqu'on veut utiliser des fonctions avec deux ou plusieurs paramètres, c'est impossible avec stdcall déclaré comme « pascal », car les paramètres sont mélangés. Le mieux est d'utiliser le système de paramètres « cdecl » : c'est celui qui a été utilisé ici.

## 2.2 Réalisation de programmes d'application

On peut utiliser les logiciels précédemment cités, et de nombreux autres, à condition qu'ils puissent appeler les bibliothèques dynamiques.

### 2.2.1 Langages Basic

#### *FreeBasic*

(mais c'est difficile de faire des logiciels complexes, avec menus déroulants, fenêtres graphiques et boîtes de dialogue)

#### *Gambas3*

On peut obtenir Gambas 3 par `sudo apt-get install gambas3`

### 2.2.2 C/C++

Les compilateurs sont directement disponibles sur la version de base de Raspbian.

### 2.2.3 Langages Pascal : FreePascal et Lazarus

FreePascal et Lazarus permettent d'utiliser les bibliothèques dynamiques créées en FreeBasic. Le logiciel Mensurasoft-LZ a été correctement compilé, et fonctionne bien sur Raspberry (voir

[http://sciencexp.free.fr/index.php?perma=Mensurasoft\\_LZ](http://sciencexp.free.fr/index.php?perma=Mensurasoft_LZ)).

### 2.2.4 Python (2 et 3)

Python est disponible spontanément sur Raspberry. « python » lance Python2, et pour avoir Python3, il faut demander « python3 ».

On peut aussi utiliser le langage basic Gambas3

```
$ sudo apt-get update
sudo apt-get install gambas3
```

### 2.2.5 Ruby

(par sudo apt-get install ruby)

## 3 Le pilote pour les broches du Raspberry

Raspberry peut être « autonome » : il a déjà des broches permettant de faire des mesures, ou de commander des appareils extérieurs.

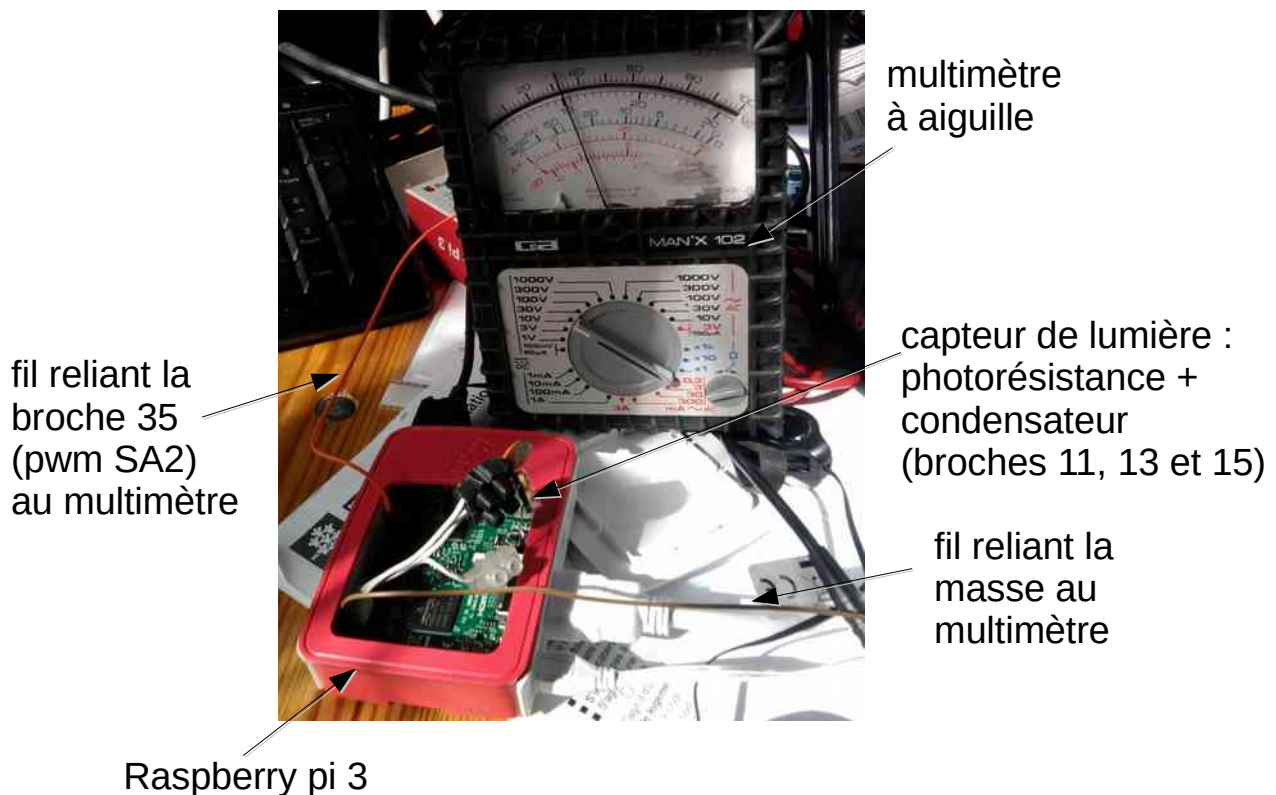
utilisation Mensura	numéro GPIO	Rangée 1	Rangée 2	numéro GPIO	Utilisation Mensura
		1	2		
		3	4		
		5	6		
EBO	7	7	8		
		9	10		
EA0 (condensateur)	0	11	12		1 SA0 (pwm)
	2	13	14		
	3	15	16		4 EA1
		17	18		5 (condensateur)
		19	20		
		21	22		6 SB0
		23	24		
		25	26		
		27	28		
SB1	21	29	30		
SB2	22	31	32		26
SA1 (pwm)	23	33	34		
SA2 (pwm)	24	35	36		27
EA2 (binaire)	25	37	38		28
		39	40		29

en noir : masse  
En vert : broches normales d'entrée-sortie  
En orange : 3,3V  
En rouge : 5V (attention ! Risque de détérioration)

Les broches de Raspberry permettent de commander ou de mesurer des variations de tension, mais normalement, seulement par tout ou rien (entrées binaires et sorties binaires).

On peut faire des sortes de sorties analogiques par des broches pouvant avoir une modulation PWM (un certain temps sous tension, puis un certain temps hors-tension). Elles permettent de commander des servomoteurs, ou d'allumer plus ou moins fort une diode, et même de faire une vraie tension variable, à condition d'y associer un condensateur.

On peut faire une entrée analogique capable de mesurer des résistances (photorésistances, photodiodes, phototransistors, thermistances, potentiomètres rotatifs ou linéaires...) en mesurant le temps de charge ou de décharge d'un condensateur à travers la résistance variable. C'est le principe de mesure des positions des manettes de jeu des anciens Apple II et IBM-PC. Pour cela, il faut 1 broche en sortie et une broche en entrée (+ une broche de masse).



Choix arbitraire :

EA0 = broches 9, 11, 13

EA1 = broches 14, 16, 18

(EA2 = broche 37, 0 ou 1)

SA0 = broche 12

SA1 = broche 33

SA2 = broche 35

EB0 = broche 7

SB0 = broche 22

SB1 = broche 21

SB2 = broche 22

Nom du pilote : libbibdyn\_raspi\_fb2016.so, à partir de bibdyn\_raspi\_fb2016.bas

## 4 Pilotes pour d'autres appareils, reliés au Raspberry

### 4.1 Pilote pour le « système »

Il affiche simplement l'heure à partir de l'horloge de l'ordinateur

## 4.2 Pilote pour Arduino

L'Arduino Uno fait apparaître un port `/dev/ttyACM0` (ou `/dev/ttyACM1` si un autre appareil est déjà relié à une prise USB).

## 4.3 Pilotes pour autres appareils

(non encore réalisés ; normalement, les appareils à brancher sur une prise USB et faisant apparaître un nouveau port série (`/dev/ttyUSB0`, ou `/dev/ttyACM0...`) doivent être programmables en FreeBasic)

## 5 Que faire si ça ne fonctionne pas correctement ?

Pour que l'appareil fasse des mesures, il faut qu'il en ait l'autorisation. Il faut que le logiciel principal (Mensurasoft-LZ) et le pilote aient le droit d'accéder au système. Pour cela, le mieux est de les exécuter en mode « super-utilisateur » :

- soit en ouvrant une console, puis en frappant « `sudo pcmanfm` » pour ouvrir l'explorateur de fichier en mode super-utilisateur
- soit en ouvrant une console, puis en se déplaçant par l'instruction « `cd` » vers le répertoire où est l'exécutable `mensurasoft-LZ`, et en frappant « `sudo ./mensurasoft_LZ` »