Pierre Dieumegard
professeur de SVT
Lycée Pothier
45000 Orléans
pierre.dieumegard@ac-orleans-tours.fr

Mensurasoft-LZ, all-purpose software for scientific experimentation

Mensurasoft-LZ is a software of measurement and experimentation per computer. With him, you can make measures by means of apparatuses and devices of measurements, and visualize the chart of it. You can send the table of measurements to the clipboard or save this table in a file, which make it possible then to analyze and model these measurements by the analysis software of data, as spreadsheet-graphics packages, statistical software, etc

It is general-purpose for several reasons:

- Mensurasoft-LZ can work with (potentially) all measuring equipment, thanks to the system of drivers of measuring equipment: you must have the pilot of your device, and Mensurasoft-LZ can make the measurement with this device.

- Mensurasoft-LZ uses standard text-files to save data. These files are readable by all the normal software of data-analysis.

- Mensurasoft-LZ is portable: you do not need to make a complex installation when you want to use it, nor to make a complex desinstallation when you do not want any more to use it.

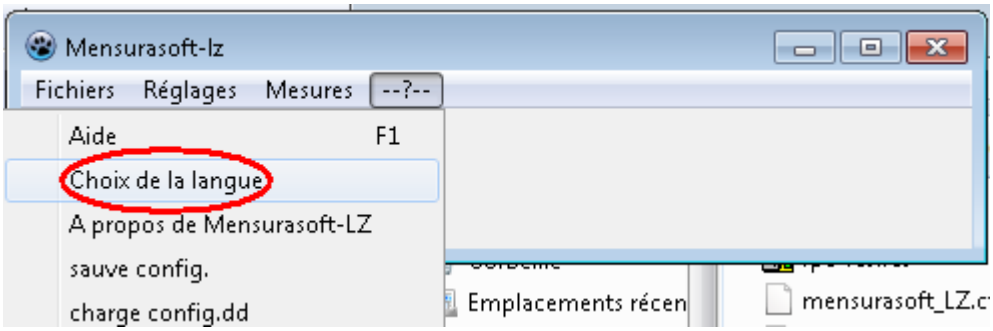
- Mensurasoft-LZ works under the two principal operating systems, Microsoft-Windows and Linux. In these booklet, any screenshots come from Linux, others come from Windows: the aspect is slightly different, but the principle is the same one.

- Mensurasoft-LZ is a free software. You can modify it to adapt it to your own needs.

- When the adjustments of your experimentation are correct, you can save them in files of configuration. Thus, the next time, you will be able to reload the file, and your experiment will be ready, without needing manually adjust the parameters of Mensurasoft-LZ.

- Mensurasoft-LZ can work in various languages. The character strings of the menus and the dialog boxes are defined in files with extension .lng, which one can modify easily.

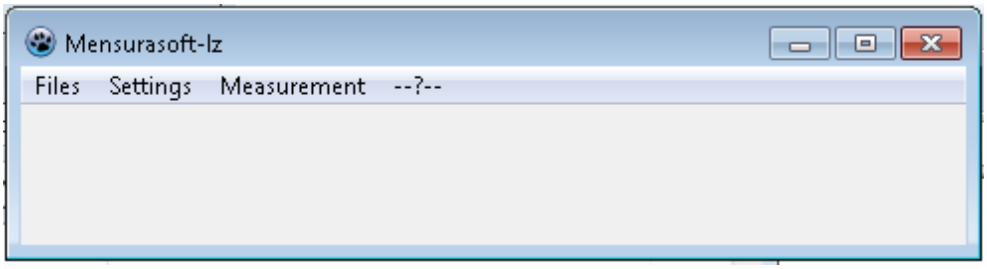
By default, Mensurasoft-LZ uses french language. To have menus and dialog-boxes in english, there are several ways :
The simplest way is to select -- ? -- / Choix de la langue, and select file mensurasoft_lz_en.lng
For the other ways, see explanations about configuration files, and command-line, at chapter 6



Contents

1 The screen when starting up of Mensurasoft-LZ

To launch Mensurasoft-LZ, double-click on the name of the executable file: Mensurasof-LZ.exe for Windows, and Mensurasoft-LZ for Linux.
In this first chapter, screenshots will correspond to a driver for demonstration, the "system-driver ", who does not need real connected measuring device.
Only a small rectangular window appears, with the upper banner indicating "Mensurasoft-LZ", and a menu bar with four menus.



1.1 Files menu

It has only two options: Open, to read a measurement file saved to disk, and Exit to exit the program

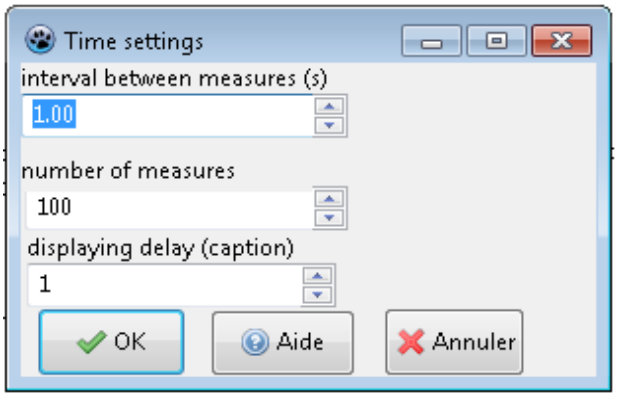
1.2 Settings menu

It is an important menu, with several options.

Time

You can choose :

- interval between measures, which is 1 second by default.
- number of measures in a series. The total time of acquisition for a series will be this number multiplicated by the interval. These values are used only during acquisition of a series of measures (menu "Measurement" / Launch measurement).
- delay for displaying the banner, when a device is choosed.

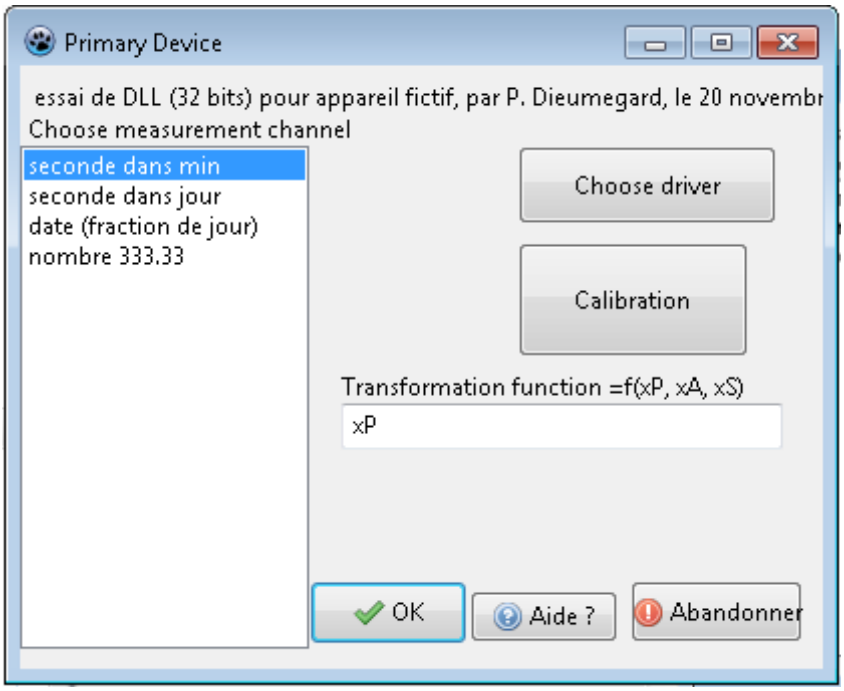


Primary device

This option will choose the driver for measurement device ; this is important. When you click on this option, a dialog-box appears, with a blank list of measurement channels on the left : this blank list is normal, because there is no device selected.

You must click on the button "Choose driver" ("calibration" does not matters now).

After clicking, you can choose a driver (files .dll for Windows, or .so for Linux).



After this choice, and if the driver has the good functions, you can see that the list on the left is full : it has one or several lines.

Some devices have only one measurement channel (for example a pHmeter), others have several (for example Arduino).

Here, example is given with "system-driver", who displays only the time.

This channel for "Primary channel" is xP. Transforming function can change values given by this channel.

By default, transforming function is xP, i.e. there is no transformation. If you want to multiply this number by 1000 (for example to transform volts to millivolts), you can type xP*1000. In this transforming function you can put 4 variables : xP, which is value from primary channel, xA which is value from auxiliary channel, xS which is from supplementary channel, and xT, which is time (in seconds). In these formulae, uppercase or lowercase does not matter.

In the graphical windows, this primary channel is drawn in red.

After validation, the banner of the windows is changed : you can see the title of the channel, and the actual value of this channel, renewed every second (for changing interval : see the menu Settings | time)

Auxiliary device

Operating way is the same for this channel, noted xA, which will be drawn in green in graphical windows.

You can choose either the same driver, either the driver for an other device.

Title of the channel and actual value are shown in the banner of the main window.

Supplementary device

This third option will be the "supplementary channel", denoted by xS, which will be represented in blue in graphical windows. You can choose either the same driver as before, or another driver. Thus, it is possible to make measurements on three different channels, from three different devices.

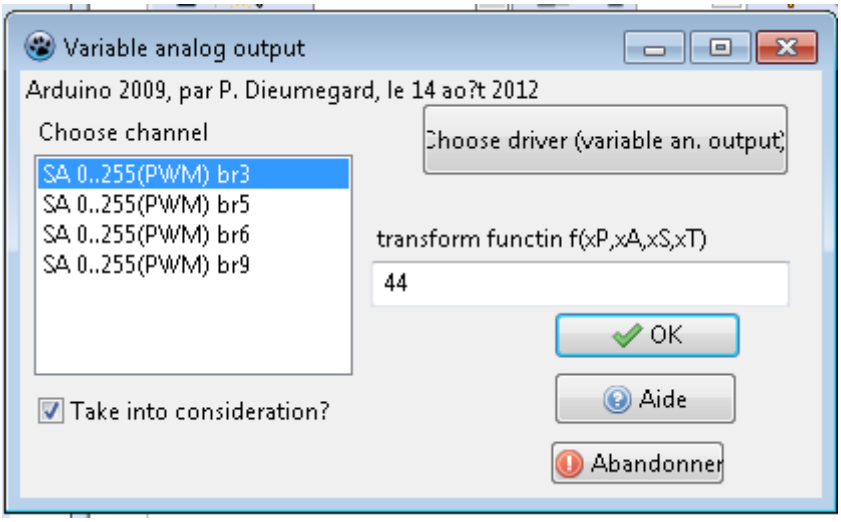
The title of chosen channel, and the current numerical value, are displayed on the right part of the banner (if the window is wide enough !)

Variable analog output

The analog outputs are used to vary a value gradually, for example a voltage between 0 to 5 volts

can vary by steps of a few millivolts.

As before, a dialog box opens, and you must choose a driver to have a list of possible outputs. If the driver does not have functions "Analog outputs", this list remains blank.



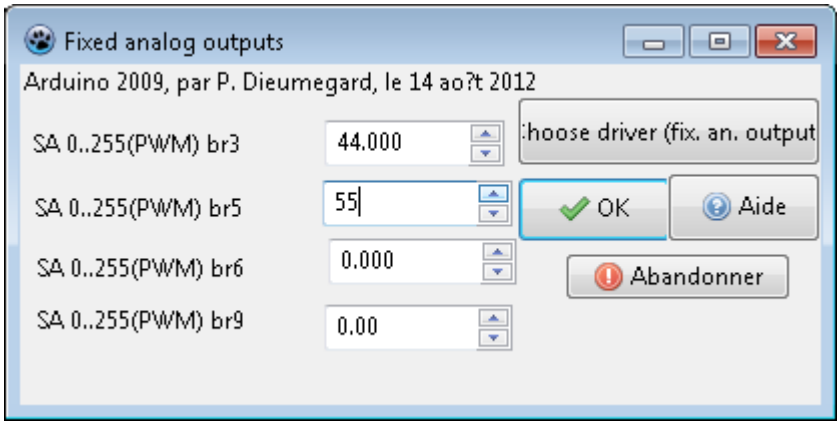
Contrary to the next option "fixed analog output", this option will manifest its effects only when you launch an acquisition (by Measurement | launch measurement).

In the input line of transformation function, you can put :

- either a fixed value (for example 3). In this case, when acquisition is launched, the output will be set to 3 (volts, if this output controls a voltage), if check-box "take into consideration" is checked.
- either a more complex function, if you want to vary this value during the time of acquisition. This function may have variables xP (value from Primary channel), xA (value from Auxiliary channel), xS (value from Supplementary channel) and xT (time in seconds). For example, type xT/100 to cause a gradual increase from zero at the beginning to 1 at 100 seconds, 2 at 200 seconds, etc. By a good function, you can have a lot of values and regulations.

Fixed analog output

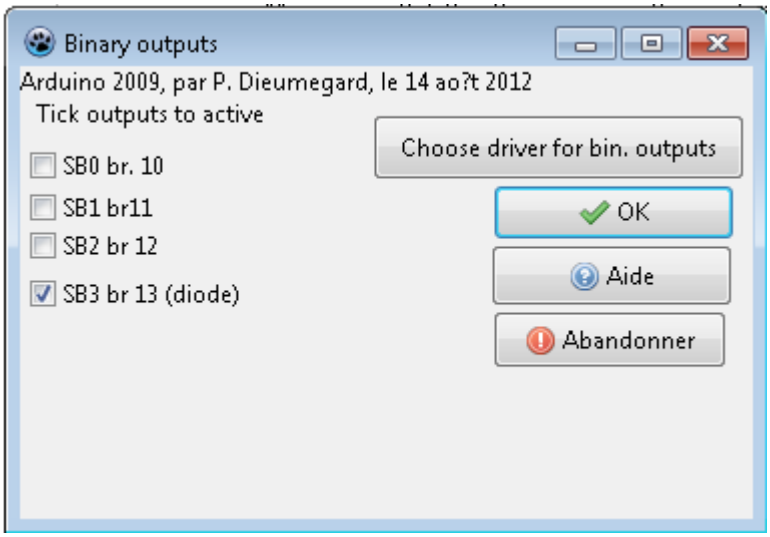
This option allows you to set one or a few analog outputs (whereas the previous option "Variable analog output" there was only one channel). If there is no analog output in the selected driver, the left side of the windows is blank. If there is one possible output, on the left side, there is a line with name of the channel and an input line, for the value of this analog output. This value will be set as soon as you confirm by "OK". If there are multiple outputs (for example with Arduino), several input lines are shown : all of them will be set by clicking on "OK".



There is no transformation function.

Digital outputs (binary outputs)

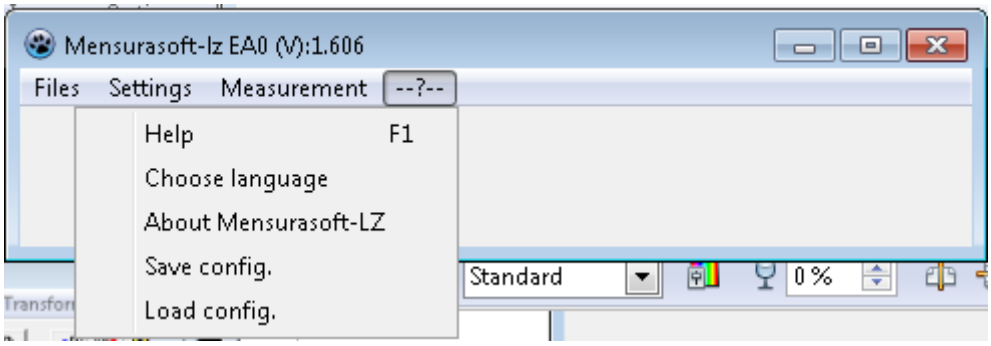
As for fixed analog outputs, if the driver has functions for digital outputs (= binary outputs, varying only between two states), these outputs are shown on the left of the dialog box. When the box is checked, the output will be set to "on" when validating by "OK" ; when it is not checked, the output will be set to "off".



1.3 Measurement menu

This is the main work for Mensurasoft-LZ : make a series of measurements. There is only one option in this menu : "launch measurement !", which launches the acquisition in a new graphical window. For details, see the following chapters.

1.4 Menu "question mark" : choice of language, configuration management



Help by F1

This option triggers the opening of a browser with an html file (a default browser must be declared). The files depends on the language selected (see below).

Choose language

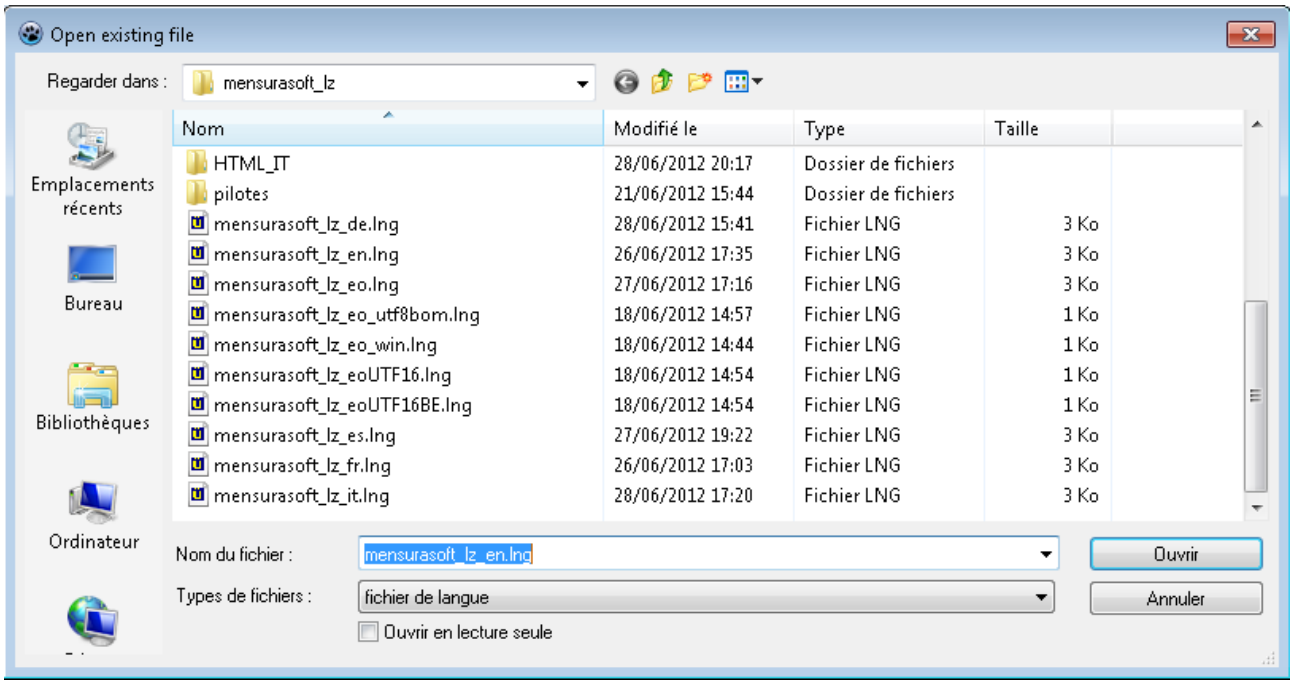
This option opens a window searching files with extension .lng (as LaNGue, LaNGuage, LiNGvo, LleNGua, etc). Normally, the last two letters of the filename correspond to the language code (en for english, fr for french, eo for esperanto, it for italian, de for german, es for spanish).

After choosing a language and validating, the menu of the main window of Mensurasoft-LZ is changed for choosed language.

By cons, if you had a graphical windows with a series of measures, the menu of this graphical window is not changed. Only the future windows will have menus in the new language.

Changing language can also change language of help files.

Changing language does not change language of drivers, because it is described in the driver, not in Mensurasoft-LZ.



About Mensurasoft-LZ

This option gives a window, with the origin of Mensurasoft-LZ.

Save config

This option will save the current configuration, in a file with extension .cfg : name of drivers, used channels, transforming functions, time for acquisition, language, etc.

For example, if you have spent many hours developing an experiment about enzymatic kinetics, you can save all the settings in a file "enzymes.cfg".

Load config

This is the opposite work : it loads the configuration, and avoids having to reselect acquisition time, measurement channels, transforming functions, etc.

Reloading the file "enzymes.cfg", you will directly experiment on enzymes with your good settings.

2 Launch a simple experiment

This example is shown with Expeyes, a device to plug into an USB port. When used in Windows, it is (often, generally...) connected to Com13: ; when used on Linux, it is (generally...) connected to /dev/ttyUSB0.

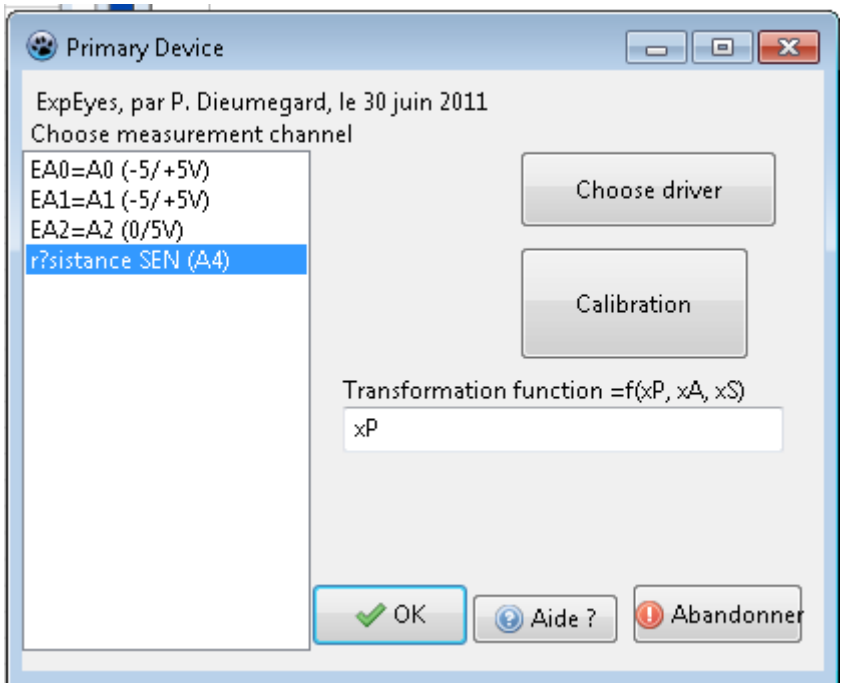
The experiment will measure the light falling on a photocell (or a photodiode with a good polarity), connected between connector "SEN" and ground. Fist, you must do this connection.

2.1 Choice of measurement device for primary device

Choose it by menu "Settings | Primary device". You must choose the driver (.dll for Windows, .so for Linux) for ExpEyes.

2.2 After selecting the driver : selection of measurement channel

There are four channels for analog input. Choose "SEN" channel, which can measure resistance, while the first three channels measure voltages.



2.3 Possible choice of a transforming function

If you know the characteristics of the sensor, and the correspondence between the displayed number and the number of luxes or watts per square meter, you can change the transforming function.If not, do not change anything, and keep "xP", i.e. there will be no transformation.

2.4 Time settings

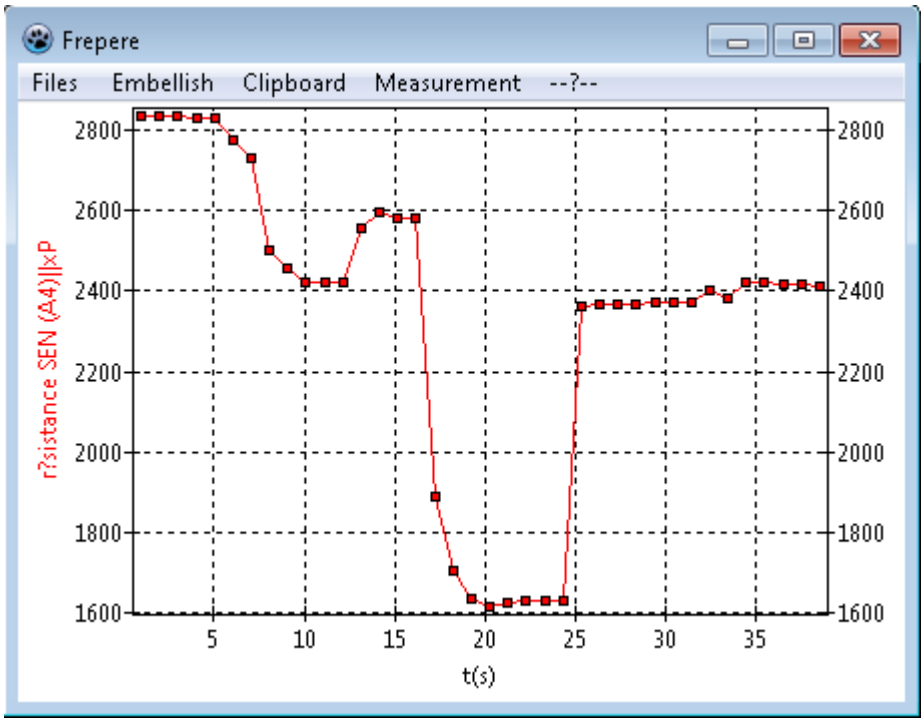
By default, an experiment will have 100 measures, spaced by one second. Keep these settings.

2.5 Launch a series of measurements

Pressing "Measurement | launch measurement !" opens a new graphical window, where measures are displayed gradually, as red dots connected by a red line.

Coordinates automatically are adapted to spread the curve. X-axis is time, and Y-axis is the resistance of the sensor. When you illuminate the photoresistor, the value decreases, and when is the dark, the value increases.

If you want to stop this measurements, just click on "Measurement | stop", otherwise the action continue until the end, at 100 seconds.



3 Graphical measurement window, and its menu

The graphic window has its own menu, which does not have the same role as the main menu of main window. The menu of graphical window applies only to the data processing of this window.

3.1 File menu, for saving the measurement series

The values are stored in text-files "CSV", with each measurement in a row, and each column for a variable (left column is the time).

These files are easily readable by spreadsheets (OpenOffice Calc, Excel, Gnumeric), by statistical software (Past), or numerical software (Matlab, Freemat, Scilab, Octave, R...).

But there are several ways to save such files :

- The true standard CSV format uses a point as decimal separator, and a comma as separator between fields. This is the first option in this menu ("Save CSV (.,)")

- Some spreadsheet, for some countries (France, for example) wish to have comma as decimal separator and semicolon as separator between fields. This is the second option "Save CSV (,;)".

- TAB can be used to separate fields. This is the third option.

- All of the above options had a special first line, with titles of the variables. This is useful with spreadsheet, but not for numerical software (Freemat, Scilab...), who want only numerical values. This fourth option save only values, with a point as decimal separator and a comma as field separator.

3.2 Embellish, to give titles to the chart

You can give a general title for the chart, in the banner of the window. You can change title of X-axis and title of Y-axis.

3.3 Clipboard menu, to copy data to the clipboard

It sends the measurement result to the clipboard, to give data to spreadsheet (or other software).

These copied data use TAB as field separator. As for recording in file, there are several options : copy only numerical values (either comma or point as decimal separator), or also copy titles (either point or comma...)

The last option is only to copy the chart, in order to a text-processor or a graphical software.

3.4 Measurement menu, to stop measurement

This option is for single use.

3.5 Help menu

This option launch the browser, with an HTML file, if you forget these instructions for use.

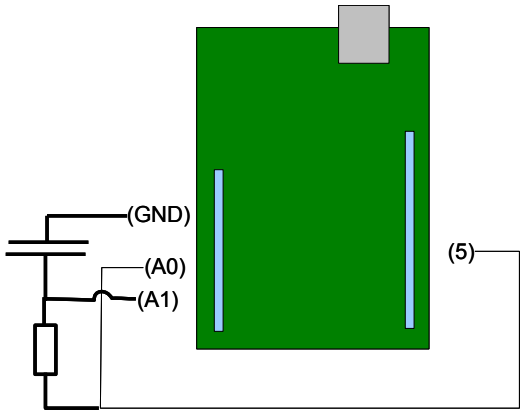
4 More complicated experiment, using an analog output

In this chapter, you will study charge and discharge of a capacitor through a resistor, with the Arduino board.

When connecting Arduino to a computer with Windows, it is often Com4: (or Com7:, or others...), when plugged into a computer with Linux, it is often /dev/ttyUSB0 or /dev/ttyACM0.

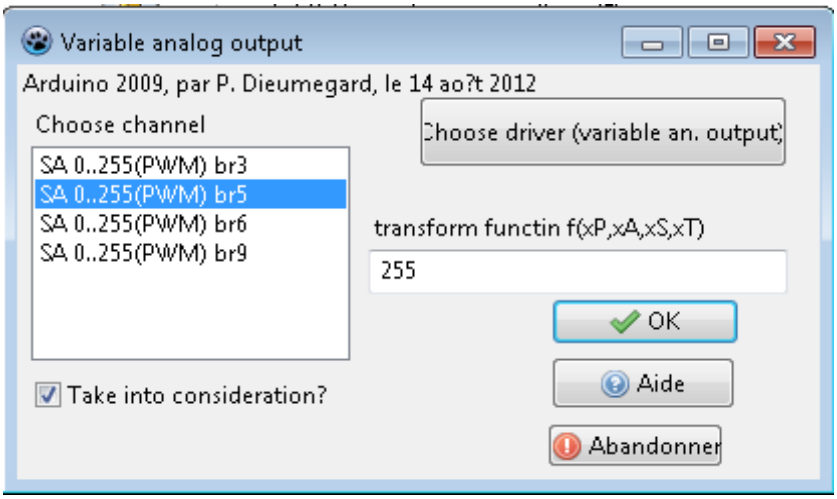
4.1 Experimental material

Connect:
- ground to a capacitor (100µF for example), itself connected to a resistor (100 kOhms for example), itself connected to the analog output in connector 5.
- analog output connector 5 will be connected to analog input A0 (for reference voltage)
- the junction capacitor-resistor will be connected to the analog input A1.



4.2 Experiment without complicated formula for analog output

Settings for channels:



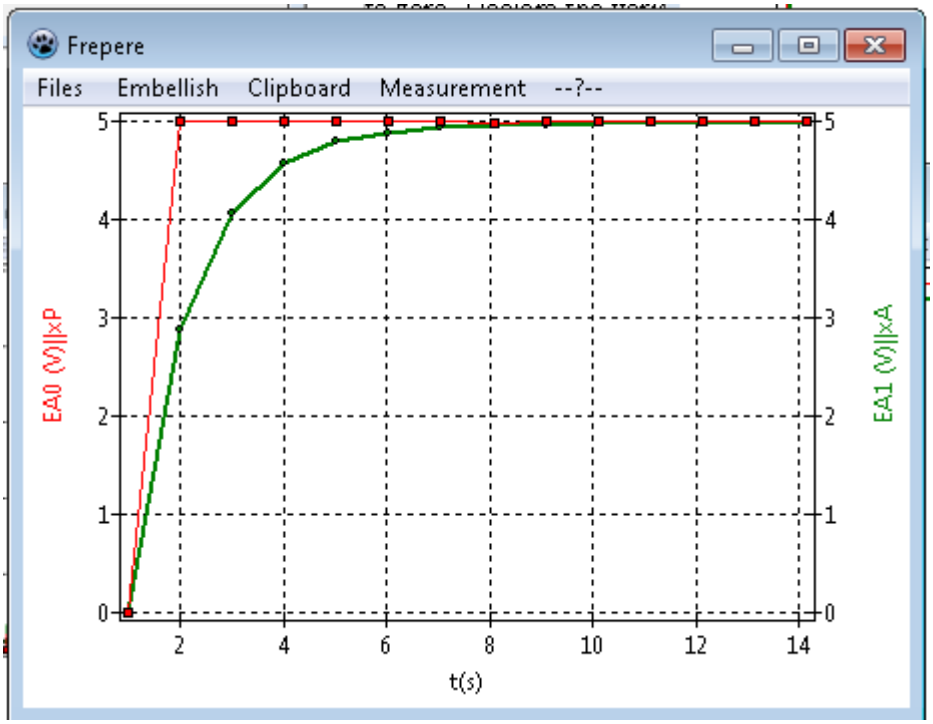
- primary channel : analog input A0
- auxiliary channel : analog input A1

Initially, voltage at pin 5 is equal to zero. Declare the variable analog output at pin 5 with a fixed voltage 5V (type 255 in the input line).

This connector will be to 5V only at the beginning of data acquisition. (simple validating and closing the dialog box does not change the voltage for *variable* analog outputs).

Launch experiment for charge

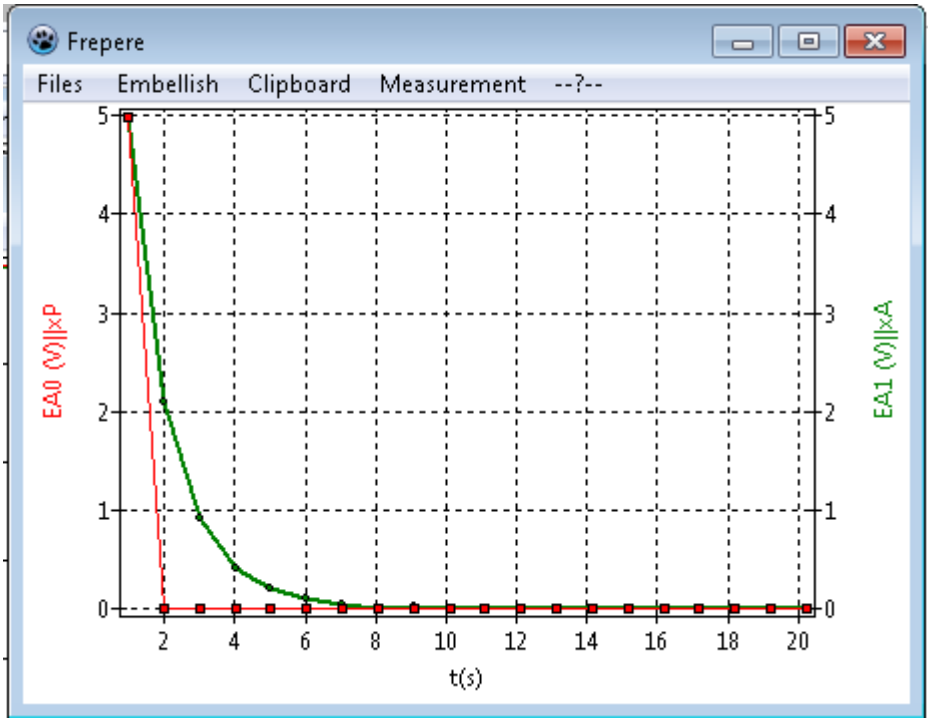
When acquisition begin, connector 5 immediately is put to 5V, so A0 (analog input) also. Its values are shown in red.



By cons, analog input A1 only increase progressively the voltage, when the capacitor is charging to 5V (shown by the green curve).

Experiment for discharge

Change the declaration of variable analog output, and put it to 0
Launch a new series of measures.
Connector 5 goes to 0, and analog input 0 too (in red). But voltage of analog input A1 drops only gradually, as the capacitor discharges (in green)
At the end of this series, the capacitor is discharged, and its voltage is 0V.



You can start again a series of measurement, by declaring the analog output (variable) of connector 5 to 5V, and so on...

4.3 Experiment with a more complicated formula for analog output

Do not change experimental material.

In the same series of measure, there will be initially the charge of capacitor, then its discharge, by programming of analog output by a "less simple formula" for transforming function in the input line.

Declare a duration for 40 seconds.

During the first 20 seconds, capacitor will be charged, then during the 60 last seconds, capacitor will be discharged.

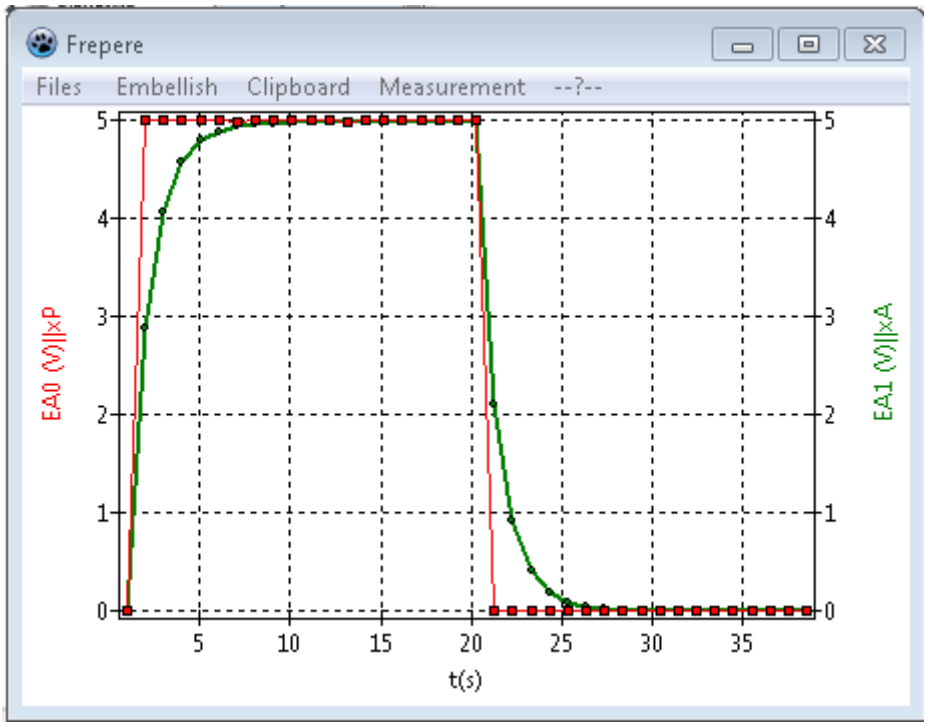
This is allowed by this formula for transform function : 255*(1-int(xT/20)).

255 means 5 volts, the maximum voltage to reach.

At the beginning of acquisition, xT (time) is equal to zero, then increases (1 for each second).

As long as xT is lower than 20 (20 seconds), xT/20 is lower than 1, therefore int(xT/20) is equal to zero : voltage at connector 5 is equal to 5V, and capacitor is charged through resistor.

From 20 seconds, xT/20 is higher than 1, therefore int(xT/20) is 1, and voltage of connector 5 is 5*(1-1) = 0. There is thus discharge of the capacitor through the resistor.



Thus, by launching only one acquisition, but with a more complicated formula for variable analog output, you can have successively charge and discharge of the capacitor.

5 How to use transforming functions

These functions are used for analog inputs and for variable analog output.

Source program comes from a file of SWAG library (http://webtweakers.com/swag/MATH/0130.PAS.html).

5.1 Possible variables

Analog inputs are xP (value sent from driver of primary channel), xA (value sent by driver of auxiliary channel), xS (value sent from driver of supplementary channel).

For variable analog output, you can use xT, which is the time (in seconds). By it, you can make square waves, or triangle waves, or saw tooth waves...

5.2 Opérators and functions

The usual operators are possible : +, -, *, / , ^, !

Available functions :

sin, cos, tan, cot, acos, asin, atan

int,sqrt, abs, log, ln, exp, pi

5.3 Comments after the function

You can type comments after the function. These comments can be useful in configuration files, to remind you why is this transformation.

For example :

-3.5*xP // this is a comment.

6 A few files to use Mensurasoft-LZ more easily

6.1 Language files .lng

They are text-files, with extension .lng, whose two last letters show the language (de for german, en for english, eo for esperanto, fr for french, it for italian, es for spanish...). They are read by Mensurasoft-LZ (menu -- ? -- "choose language", or from a configuration file).

Every line has "=" as separator. On the left is name of the function : do not change it. On the right is the text in this language : you can change it as you like.

These language files are important, not only for menus and dialog boxes, but also for the help files. When a line is "html=HTML_DE", this means that html files for help are in a sub-directory "HTML_DE".

When there is no file of language, menus and dialog boxes are in french.

For load automatically a language when Mensurasoft-LZ starts, you can put the name of a language file in command line.

If a language file (in the same directory as Mensurasoft-LZ) is named "mensurasoft_LZ.lng", it is automatically loaded.

6.2 Configuration files

When an experiment is correct, you can save the settings in order to be able to start it again easily.

In menu --?--, option "Save config" records the settings in a text-file, with extension .cfg.

Then, the following day, or several years later, after launching Mensurasoft-LZ, if you use option "load config", these settings will be automatically reloaded. Drivers for analog inputs and measurement channels will be immediately operational. But fixed outputs (binary and analog) will not be actives : for activate them, you must open dialog box (menu Settings) and validate by "OK".

If there is a file with name "mensurasoft_LZ.cfg", it will be automatically loaded.

You can also load automatically a configuration file by putting his name in a command line. For example, with Windows, type mensurasoft_LZ.exe enzymes.cfg.

6.3 Help files

They are HTML files. When language file did not indicate an other place, these files must be in subdirectory HTML.

Normally, for various languages, help files are in subdirectories whose last letters give the language. For example, help files for german are in subdirectory HTML_DE, and files for esperanto are in HTML_EO, etc.

7 Devices usable by Mensurasoft-LZ

This list is incomplete : it is (relatively) easy to make new drivers for new devices, by several programming languages. A suitable language must be compiled, to be able to make dynamic libraries.
For Linux and Windows, good languages are FreePascal, PureBasic, FreeBasic, and several C++.

Until autumn 2012, Mensurasoft-LZ was using "stdcall" functions. From october 2012, new versions have "cdecl" functions. It is thus possible/probable that old drivers do not have all these functions, and are not usable with new versions of Mensurasoft-LZ. There are two solutions : add these cdecl function to the driver, and recompile it, or use a version of Mensurasoft-LZ using stdcall functions. (source files to be modified are ucommun.pas and u_chargebibdyn_base.pas)

In Mensurasoft-system, integers are coded by 4 bytes, reals are "double precision" (by 8 bytes), and strings are "string pointers, zero ended"..

7.1 Principle of drivers (dynamic libraries)

They **must have** following functions :

stdtitre and ctitre : without parameter, give a short string (title of the driver)

stddetail and cdetail : without parameter, give a less short string (detail of the driver)

stdead and cead : analog input, with an integer parameter, which is number of the channel (from 0), and gives a double (real).

stdnead and cnead : name of analog input, with an integer parameter (number of the channel, from 0) ; it gives a string.

They **may have** following functions :

stdsad and csad : analog output, with two parameters, an integer (number of the channel) an a double (value for the output) ; it gives the value.

stdnsad and cnsad : name of the analog output, with an integer parameter (number of the channel) ; gives a string

stdcalibration and ccalibration, with a string parameter, and giving a string. These strings are not used by Mensurasoft-LZ. When this function exists, it is possible to use the button "Calibration" of dialog box of input device. This option launch this function of the driver, by sending an empty string. This calibration function is useful particularly for spectrophotometers or colorimeters, to do blank calibration, or for oxymeters, to set zero and slope.

Again problems about coding of strings. Characters can be coded on one byte (Ascii-Ansi coding) or two bytes (Unicode or UTF8 coding). Sometimes there are problems, and the best practice is to make drivers with ANSI-coding.

A lot of drivers are available at http://sciencexp.free.fr

7.2 General purpose devices

Unless otherwise stated, drivers exists both for Windows and Linux.

To plug in an USB-connector

Arduino

ExpEYES
Velleman K8055 (Windows only)
Orphylab (Windows only)
Eurosmart (Windows only)

To plug in a serial connector (or an USB-serial converter)

Orphy GTS, Orphy GTS2, Orphy GTI, Orphy Portables 1 and 2...

To plug in an ISA or PCI connector

Jeulin ESAO3 and ESAO4
Candibus and Candiplus
Pierron SMF10-SMF100-Expert
MEP : PMB and PMB>

To plug in a parallel connector

MEP Ades

7.3 Specialized devices, to plug in a serial connector or an USB-serial converter

Multimeters MAS345, TES 2730, My77, Metex...
pHmeters Tacussel PHM210, LPH330, PHN130 ; Schott Handylab2
scales Sartorius PT600, Precia, Adam PGW, Ohaus Scout
thermometers TM906A,
luxmeters LX105
spectrophotometerss Secomam, Jenway

and gameport for joysticks....

8 Useful complements for Mensurasoft-LZ :

programs for data analysis and modelling

PAST : statistical software http://norges.uio.no/past/download.html

Freemat : numerical computing http://freemat.sourceforge.net/

Scilab : numerical computing http://www.scilab.org/

OpenOffice : spreadsheet http://www.openoffice.org/

Gnumeric : spreadsheet http://projects.gnome.org/gnumeric/

Regressi : modelling software http://jean-michel.millet.pagesperso-orange.fr/telechargement.htm