

ISSN 2310-6697

toiser—open transactions on independent scientific-engineering research

FUNKTECHNIKPLUS # JOURNAL

Théorie—Expérimentation—Métrologie—Logiciel—Applications

ISSUE 8 – WEDNESDAY 30 SEPTEMBER 2015 – YEAR 3

REPRINT

The Mensurasoft-System: Software Drivers for Scientific Devices in the Most Common Computer Languages on All Major Operating Systems with a Prototype API for Measurement Data Processing Applications – Practical Examples

P. Dieumegard



*This small European Journal is
In the Defense of Honesty in Science and Ethics in Engineering*

Publisher – otoiser—open transactions on independent scientific engineering research, www.otoiser.org—info@otoiser.org, Hauptstraße 52, 2831 Scheiblingkirchen, Austria

Language – We declare the origins of the Journal by using in the cover page English, German and French, as well as, a Hellenic vignette. However, since we recognize the dominance of US English in the technical literature, we adopted it as the Journal's language, although it is not our native language.

Focus – We consider Radio–FUNK, which still creates a vivid impression of the untouchable, and its Technology–TECHNIK, from an Advanced–PLUS point of view, Plus–PLUS Telecommunications Engineering, Electrical Engineering and Computer Science, that is, we dynamically focus at any related scientific-engineering research regarding Théorie, Expérimentation, Métrologie, Logiciel, ou Applications.

Scope – We emphasize this scope broadness by extending the title of the Journal with a Doppelkreuz-Zeichen # which we use as a placeholder for substitution of our Editorial Team disciplines: # Telecommunications etc. as above, or # High Voltage, # Software Engineering, # Simulation etc. as below.

Frequency – We publish 3 issues per year: on 31st of January, on 31st of May, and on 30th of September, as well as, an extra issue every 3 papers and a volume every 2 years.

Editions – We increase the edition number of an issue only when is needed to reform one or more of its papers—thus to increase their version numbers—but we keep unchanged its 1st edition date shown on its front page and we number its pages sequentially from 1. We count the editions of *About* separately.

Format – We use a fixed-space font, hyphenation, justification, unfixed word spacing, and the uncommon for Journals A5 (half A4) page size to achieve WYSIWYG printing and clear reading of 2 to 4 side-by-side pages on wide-screen displays

Printing-on-Demand – We can email gratis PDFs at 300-4000 dpi in booklet page scaling of either brochure or book type.

Copyright – We publish under a Creative Commons Attribution, CC-BY 3.0 Unported or CC-BY 4.0 International, License only.

Please download the latest *About* edition from
about.ftpj.otoiser.org

Editorial Team

Electrical Engineering

High Voltage Engineering # Insulating Materials
Professor Michael Danikas, mdanikas@ee.duth.gr
EECE, Democritus University of Thrace, Greece

Electrical Machines # Renewable Energies # Electric Vehicles
Assistant Professor Athanasios Karlis, akarlis@ee.duth.gr
EECE, Democritus University of Thrace, Greece

Computer Science

Computer Engineering # Software Engineering # Cyber Security
Professor Vasilis Katos, vkatos@bournemouth.ac.uk
Head of Computer and Informatics Dept, Bournemouth Univ, UK

Internet Engineering # Learning Management Systems
Lecturer Sotirios Kontogiannis, skontog@gmail.com
Business Administration Dept, TEI, Western Macedonia, Greece

Hypercomputation # Fuzzy Computation # Digital Typography
Dr. Apostolos Syropoulos, asyropoulos@yahoo.com
BSc-Physics, MSc-Computer Science, PhD-Computer Science
Independent Researcher, Xanthi, Greece

Telecommunications Engineering

Applied EM Electromagnetics # Applied Mathematics
Dr. Nikolaos Berketis, nberketis@gmail.com
BSc-Mathematics, MSc-Applied Maths, PhD-Applied Mathematics
Independent Researcher, Athens, Greece

Antennas # Metrology # EM Software # Simulation # Virtual Labs

Applied EM # Education # FLOSS # Amateur Radio # Electronics
Dr. Nikolitsa Yannopoulou, yin@arg.op4.eu *
Diploma Eng-EE, MEng-Telecom-EECE, PhD-Eng-Antennas-EECE
Independent Researcher, Scheiblingkirchen, Austria

Dr. Petros Zimourtopoulos, pez@arg.op4.eu *
BSc-Physics, MSc-Radio-Electronics, PhD-Antennas-EE
Independent Researcher, Scheiblingkirchen, Austria

* Copy & Layout Editing – Proof Reading – Issue & Web Management

Technical Support

Konstantinos Kondylis, kkondylis@gmail.com
Diploma Eng-EECE, MEng-EECE, Doha, Qatar

Christos Koutsos, ckoutsos@gmail.com
Diploma Eng-EECE, MEng-EECE, Bratislava, Slovakia

Information for Authors

This is a small, but independent, low profile Journal, in which we are all—Authors, Reviewers, Readers, and Editors—free at last to be Peers in Knowledge, without suffering from Journal roles or positions, Professional–Amateur–Academic statuses, or established "impact factorizations", under the following guiding principles:

Authors – We know what Work means, we respect the Work of the Independent Researcher in Science and Engineering and we want to exhibit his Work. Thus, we decided to found this Free and Open Access Journal in which to publish this Work. Furthermore, as we care indeed for the Work of the technical author—especially a young or a beginner one—we strongly support the publication of his Work, as follows:

- 0 We do not demand from the author to transfer his own copyright to us. Instead, we only consider papers resulting from original research work only, and only if the author can assure us that he owns the copyright of his own paper as well as that he submits to the Journal either an original copy or a revised version of his own paper, for possible publication after review—or even for immediate republication, if this paper has already been published after review—but, in any case under a Creative Commons Attribution, CC-BY 3 Unported or CC-BY 4 International, License, only.
- 1 We encourage the author to submit his own paper written just in Basic English plus Technical Terminology.
- 2 We encourage the author even to select a pen name, which may drop it at any time to reveal his identity.
- 3 We encourage the author to submit an accepted for publication paper, which he was forced to decline that publication because it would be based on a review with unacceptable evaluation or derogatory comments.
- 4 We encourage the author to submit any paper that was rejected after a poor, impotent, inadequate, unreasonable, irresponsible, incompetent, or "just ticking" review.
- 5 We encourage the author to submit an unreviewed paper of his own that he uploaded on some Open Access repository.
- 6 We encourage the author to upload his published papers in our Journal to a truly Free Open repository like viXra.org.
- 7 We provide the author with a decent, express, peer review process, of up to just 4 weeks, by at least 2, either anonymous or onymous, reviewers.

- 8 We provide the author with the option to choose from 2 review processes: the traditional, anonymous, close one, as well as, a contemporary, onymous, open review in our private mailing list for Peer Discussion.
- 9 Under the Clause 0 : We immediately accept for publication a research paper directly resulting from a Project Report, or a Diploma-, Master-, or PhD-thesis, which already the author has successfully defended before a committee of experts, as long as he can mention 2 members of this committee who approved his Work.
- 10 Under the Clause 0 : We immediately accept for publication any paper which is not Openly Accessible on the Internet.
- 11 We immediately publish online a paper, as soon as it is accepted for publication in the Journal.
- 12 We quickly publish an extra issue—that is in excess of the 3 issues we publish a year—as soon as the review process of 3 papers is completed.

Reviewers – Every peer may voluntarily become a reviewer of the Journal in his skillfulness for as long as he wishes. In addition, each author of the Journal must review one paper in his expertness for each one of his published papers.

Readers – Every reader is a potential post-reviewer: we welcome comments and post-reviews in our private mailing list for Peer Discussion.

Editors – Every editor owns a PhD degree—to objectively prove that he really has the working experience of passing through the dominant publishing system. An editor pre-reviews a paper in order to check its compliance to our guiding principles and to select the appropriate reviewers of it. We can accept for consideration papers only in the expertise areas currently shown in the Editorial Team page, above. However, since we are very willing to amplify and extend the Scope of the Journal, we welcome the volunteer expert, in any related subject, who wants to join the Editorial Team as long as he unreservedly accepts our guiding principles.

Electronic Publishing

We regularly use the Free Libre Open Source Software Libre Office with the Free Liberation Mono font and the Freewares PDFCreator and PDF-Xchange Viewer. We also use, with some basic html code of ours: the Free Open Source Software Open Journal System OJS by the Public Knowledge Project PKP installed in our website, and the Free Open Digital Library of

About

Internet Archive website, where we upload Issues, Paper reprints, *About* documents, and Volumes, in both portrait and landscape orientations, for exceptionally clear online reading with the Free Open Source BookReader.

Submissions

We can consider only papers submitted in a format which is fully compatible with LibreOffice—preferably in odt format.

Legal Notice – It is taken for granted that the submitter–correspondent author accepts, without any reservation, the totality of our publication conditions as they are analytically detailed here, in this *About*, as well as, that he also carries, in the case of a paper by multiple authors, the independent will of each one of his coauthors to unreservedly accept all the aforementioned conditions for their paper.

Internet Addresses

Internet Publishing Website : ftpj.otoiser.org

Internet Archive Digital Library : archive.org/details/@ftpj

Printing-on-Demand : pod@ftpj.otoiser.org

Principal Contact : principal-contact@ftpj.otoiser.org

Technical Support : technical-support@ftpj.otoiser.org

Editorial Team–Technical Support List : etts.ftpj.otoiser.org

Peer Discussion List : peers.ftpj.otoiser.org

Submissions : sub@ftpj.otoiser.org

Sample Paper Template : template.ftpj.otoiser.org

ARG NfP AoI

Antennas Research Group

Not-for-Profit Association of Individuals *

arg.op4.eu – arg@op4.eu

Hauptstraße 52, 2831 Scheiblingkirchen, Austria

* The Constitution of Greece, Article: 12(3) 2008:

www.hellenicparliament.gr/en/Vouli-ton-Ellinon/To-Politevma

* The Hellenic Supreme Court of Civil and Penal Law:

www.areiospagos.gr/en/ – Court Rulings:Civil|A1|511|2008

This document is licensed under a Creative Commons Attribution 4.0 International License – <https://creativecommons.org/licenses/by/4.0/>

The Mensurasoft-System: Software Drivers for Scientific Devices in the Most Common Computer Languages on All Major Operating Systems with a Prototype API for Measurement Data Processing Applications - Practical Examples

P. Dieumegard *

Independent Researcher, Orléans, France

Abstract

For years now, although many scientific devices have a port for computer connection, the diversity of various connection types results in lack of software able to use the totality of them. This paper presents how to: (1) build software drivers with a prototype Application Programming Interface API for scientific devices using dynamic libraries under various combinations of connection ports, programming languages and operating systems, (2) employ these drivers in software programs capable of managing more than one device at the same time, and (3) use two specific software implementations of this whole Mensura (Measurement) System to select appropriate built driver, define communication settings, and collect-store-process-present experimental data.

Keywords

Mensurasoft, measurement, devices, dynamic library, programming languages, MS-Windows, Linux, driver

Introduction

In this paper, "scientific devices" are mainly measurement devices: pHmeters, voltmeters, thermometers etc, but sometimes they are actuators: motors, heaters, voltage-generators etc. Many scientific devices can be connected to a computer; in the past, usu-

ally through an ISA, PCI or RS-232, but nowadays via USB too.

Sometimes there is software to use them, often there is no software. When a software is sold or given with a device, it can be used only with this device or similar devices from the same sup-

plier. This causes problems to professional scientists, as well to teachers of sciences, and to simple amateurs of sciences. Can we imagine programs able to manage all of them?

This problem is similar to that of office software: how to make programs (spreadsheets, word-processors etc) able to manage all printers - not only already existing printers, but also the printers which will appear in the next years. The solution is similar; by using device drivers. There is a printer driver for each printer, and we can make a driver for each measurement driver.

However, there are two differences between office programs and scientific programs. Scientific devices are more various than printers, and usually produced in small quantities. Scientist often must make himself/herself the

program for his/her aim.

In the present work we will show how to use "scientific drivers" from the system Mensurasoft named by the Latin word mensura which means measure. Fig. 1 gives an illustration of the Mensurasoft system architecture: separation between programming for devices and programming for humans is an obvious necessity.

The necessity of Drivers

The construction of scientific devices is not easy. Making a pHmeter is a job for specialists, with special glasses for the electrode, special electronics, correction for temperature, etc. In the same way, building a precise scale to measure weight, or an accurate lux-meter are works of specialists. These specialists can supply a program for their devices, but not for others.

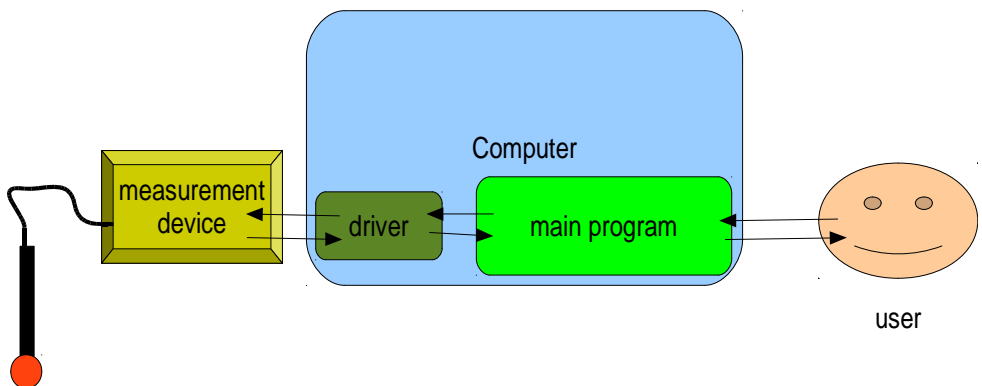


Fig. 1: Mensurasoft System

A professional scientist needs several scientific devices. For example, in ecology, we must measure accurately temperature AND pH AND light: there are not programs from devices makers for all them. A teacher of sciences needs to show to the students how to make an experiment. If there are several models of measurement devices in the classroom, with different software, the explanations will be very difficult for the teacher, who dreams about one piece of software for all these devices.

Drivers for measurement devices must be simple and easy to use, because science workers may not have great skills in programming. They must be usable with a lot of programming languages, in order to be used by many people.

Dynamic Libraries as Drivers

Dynamic libraries are pieces of compiled program, with functions callable by others programs; they are standardized and available in major operating systems: MS-Windows, Linux (named "shared objects") and Mac-OS [1]. Many programming languages can make dynamic libraries, often available as free software, but they must be compiled ones, such as: C/C++, Pascal, and some kinds of Basic.

On the other hand, almost all programming languages can use dynamic libraries, including office software such as Openoffice/LibreOffice, or the calculation specialists for example FreeMat [2], Scilab [3] Julia [4], R [5], and tools for general programming as Basic, C/C++, Pascal, Python, Ruby, Rebol.

In order to use dynamic libraries for all scientific devices, we must standardize the functions, giving them the same name in all the drivers. "Mensurasoft system" is a proposal for a possible standard: the names are short and consist of combination of letters.

Fundamental functions and additional properties

Computer must receive measures from the measurement device, and the driver needs a function for this input. In Mensurasoft system, the name of this input function has the letter "e" (as "entrée" in French, "eingang" in German, "entrada" in Spanish, "entrata" in Italian and "entry" in English). This is the main job.

If the device has an actuator, computer must send a value to the device, in order to change its state. In Mensurasoft system, the name of this output function has the letter "s" (as "sortie" in

French, "salida" in Spanish).

Inputs and outputs can be digital or binary if there are only two values as "yes/no", "1/0", "on/off", represented with the letter "b" in Mensurasoft. They are analog if there are a great number of values, for example when we measure temperature, voltage or pH: in Mensurasoft, the letter is "a" for them.

Computers use different type of numbers: integers or reals ("floating"). Reals are often coded as "double" (double precision), using 8 bytes (64 bits), and integers can be coded on 4 bytes (32 bits). When the result of an input function (or the value to be set by an output function) is an integer, there is no special letter, but when the result or the value is a "double", a letter "d" is added.

Sometimes (often) there are several inputs. For example, the little Arduino Uno has 6 analog inputs; a simple thermometer only measures temperature at one sensor, but we could have the results either in kelvins or in Celsius or even in Fahrenheit system, thus three (3) analog inputs are needed. So the input functions have one parameter meaning the channel for measurement which is usually coded as integer without a special letter; 0 corresponds

to the first channel, 1 to the second and so on.

Output functions have two parameters. The first is the number of the channel, as for input functions, and the second is the value to be set by the device (integer or "double").

In programming languages, there are several ways to pass parameters to a function: `stdcall`, `cdecl`, `pascal`, and others. "`stdcall`" is more used in Windows world, "`cdecl`" in Linux world, but this is no mandatory. In order to use all of them, we must put a prefix "`std`" if the declared function is "`stdcall`" and "`c`" if the function is "`cdecl`".

For example, in Mensurasoft system, the first analog input channel sending a real (double) is `stdead(0)` with the calling convention "`stdcall`"; the second binary output is `stdsb(1,1)` if it is set to "On" and `stdsb(1,0)` if it is set to "Off"; with "`cdecl`" convention, the third binary input is `ceb(2)` and the fourth analog output set to 3.44 is `csad(3, 3.44)`.

That is enough for main programming languages, but there are some among them which want other functions. Some languages accept parameters only as "double", even for the channel number: instead of "d", we put "double" to show that all parameters

and results are in double precision. Some languages deals with parameters and results only as "strings": we can add "str" at the end, and then the word `stdeastr (0)`, for example, stands for the first analog input. Others need special functions (R, Scilab) in the dynamic library, but they are not included in the present work. A detailed outline of the naming procedure for functions in the Mensurasoft system is shown in Fig. 2.

Additional string functions

In Mensurasoft system each numerical function as already has been mentioned above has its name, coded by a string of characters. This string is a null-terminated string, as `pchar` in Pascal, `zstring` in FreeBasic, `s` in PureBasic or `LPTSTR` in C++. The name of the function begins by a "n" (as "name", "nom", "nomo", "nome", "nombre"), e.g. `nead(1)` is the name of the second analog input. If a numerical function does not exist, its name is an empty string, i.e. a string whose length is 0; for example, `cnead(133)` is usually an empty string, because normal devices have only a few channels for measurement.

These "name functions" are very useful since:

a. when we do not know the channels of the device: we can ask `cnead(0)`, `cnead(1)`, `cnead(2)` and so on in order to find out them

b for general purpose software with menus and dialog boxes: to show all analog inputs the programmer is able to use a loop, asking the name of the analog input until this name is empty.

Additional string functions can be added for more comfortable use. A function "detail" could provide information about the device such as the company identity, the device name, the programmer's name, the date of programming etc. Even if there are not parameters, the name will be `cdetail()` and `stddetail()`. We could also define a title function, which will send a smaller string, for example only the name of the device, as: `stdtitre()` and `ctitre()`. This small string could be useful, e.g., as a title in a dialog box.

An optional function for calibration `stdcalibration(ch)` and `ccalibration(ch)` takes a string as input and gives a string as output. This function is useful only for specific devices, and the programmer can put anything in it.

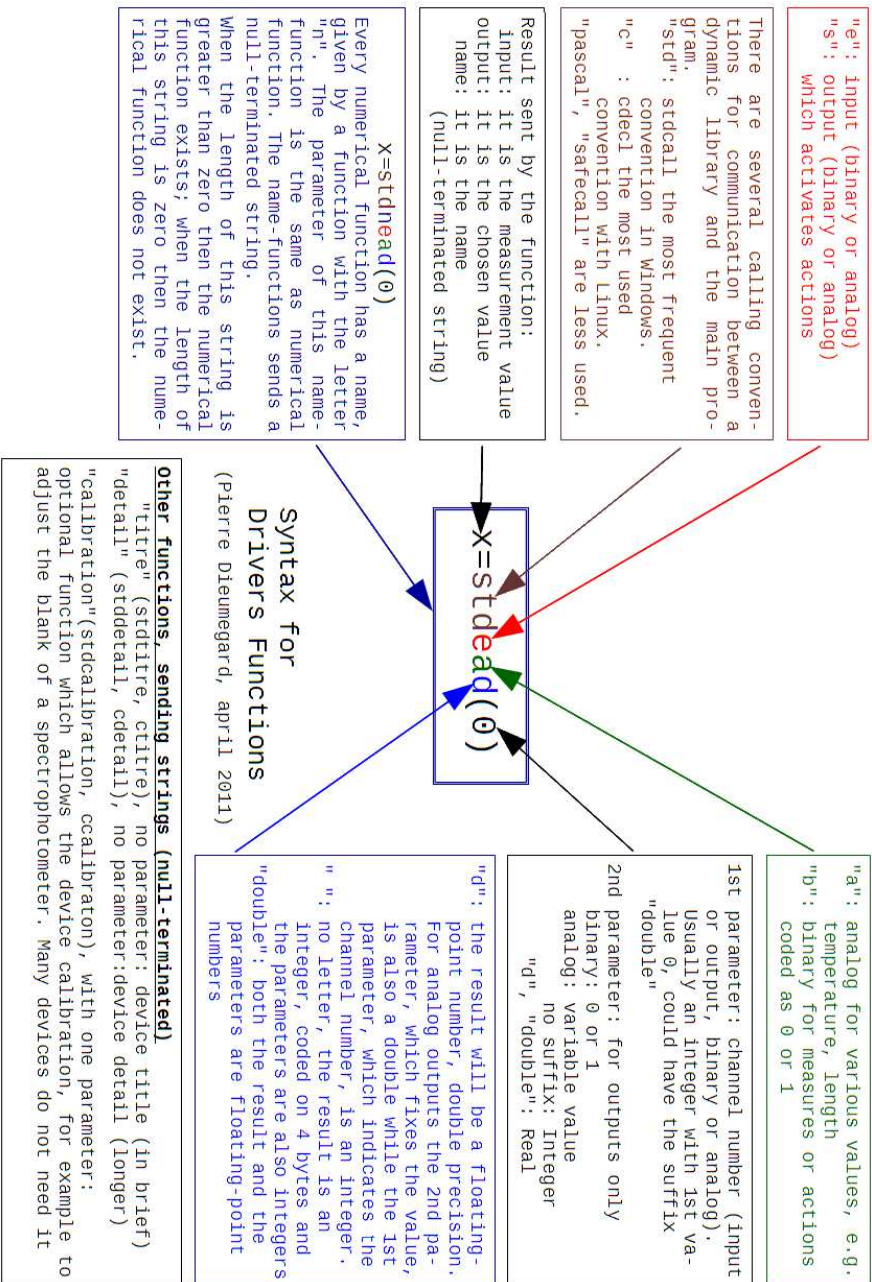


Fig. 2: Syntax of measurement function for Mensurasoft

A free project

Dynamic libraries are well known, and well standardized in programming languages. Mensurasoft is only a proposal for the adoption of a standard for the functions' names in order to allow all programmers to be able to use dynamic libraries of others.

Until now, all drivers and all application programs of Mensurasoft system are free software. The best license for drivers seems to be LGPL [6], because the drivers are small programs, and the best license for application programs seems to be GPL [7]. Nevertheless, we can imagine non-free drivers and non-free application programs, co-operating with free drivers and applications.

Three frequently asked questions need to be answered:

- a. how to get existing drivers
 - b. how to make new drivers,
 - c. how to build programs calling them
- a. The repository of the developed drivers for a lot of devices is at:

<http://sciencexp.free.fr>

mainly in French [8], but Google Translate may provide a decent translation for other languages. In this site, there are also application prog-

rams for Windows and Linux. They can use three (3) channels, for three (3) different drivers, make a graph of measured values, and save the values in files. They can transform the measured values by a formula (for example, to transform tension in volts from a pH-meter with analog output to pH). They are multilingual by language files.

Mensurasoft-PB [9] is written in PureBasic. A booklet in pdf is available in French, English, Spanish and Esperanto. Another program, Oscillo-PB, is very similar to Mensurasoft-PB, but is designed for fast measurement, like an oscilloscope. Fig. 3 shows a window from Mensurasoft-PB.

Mensurasoft-LZ is written in FreePascal/Lazarus. A booklet in pdf is available in French, English and Esperanto language. Fig. 4 shows an example of its usage in Linux illustrating the behavior of a common capacitor.

- b. The best way to build a new driver is to modify an existing driver found on [8]. You can modify a "true driver" for a real device, or a "demonstration driver" [10].

Available drivers are in Delphi, FreePascal and Lazarus, FreeBasic, PureBasic, Oxygen-Basic, CodeBlocks, Dev-C++, C++Builder, Visual-C++.

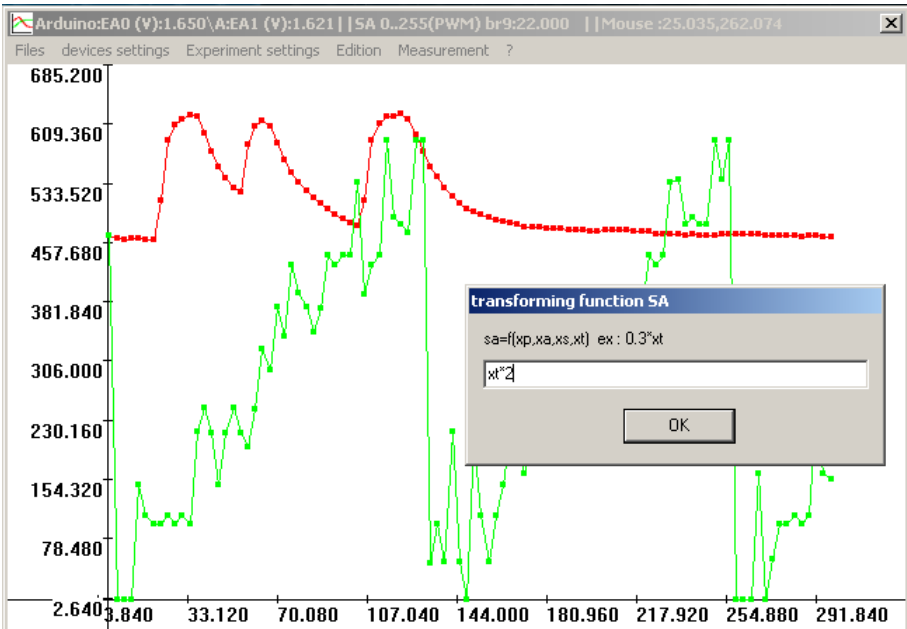


Fig. 3: An example window from Mensurasoft-PB

For example, if you have a new pH-meter, you can modify a driver only for a few functions: `stdead`, `stdnead`, `std-detail`, `stdtitre`. After compilation, you will be able to use Mensurasoft-PB, Mensurasoft-LZ and others with your new driver, and measure pH easily. Details of programming is available with booklets in French [11], English and Esperanto.

c. The booklets for programming also show how to use several languages to call these dynamic libraries: a lot of Basic, C/C++, Pascal, Logo, Python, Rebol, Ruby, Free-

mat, Scilab, Julia, R ...

Small programs could also be found at [12].

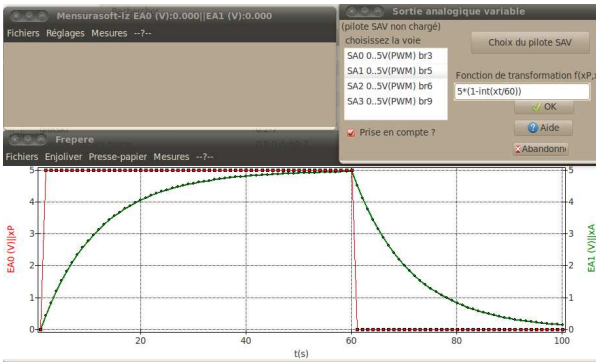
Conclusion

Mensurasoft system evolved for twenty years, and works well, at least for slow measurement, and for Linux and MS-Windows operating systems. For fast measurement, like an oscilloscope, limitations are due to the speed transfer of USB. A rather simple but systematic notation of functions implemented in dynamic libraries has been adopted and successfully applied and tested. The same system is available for Mac OS, but never

has been tested.

The author encourage those who would like to be involved to this continuing effort and to contribute their driver for another device to the

public domain via a link on the website [8] or to share a new written related application program. Do not hesitate to contact the author for any comment or related work.



Mensursoft_LZ under Linux:

Charge and discharge of a capacitor

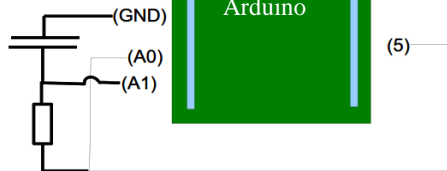


Fig. 4: Application of Mensursoft-LZ in Physics

References

- [1] Dynamic-link library: https://en.wikipedia.org/wiki/Dynamic-link_library
- [2] FreeMat: <http://freemat.sourceforge.net/>
- [3] Scilab: Scilab Enterprises, www.scilab.org
- [4] Julia: <http://julialang.org/>
- [5] R: www.stats4stem.org/r-math-functions.html
- [6] GNU Lesser General Public License: ["https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License"](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License)
- [7] GNU General Public License: https://en.wikipedia.org/wiki/GNU_General_Public_License

- [8] Sciencexp - the site of the Experimental Sciences:
http://sciencexp.free.fr
- [9] Mensurasoft-PB:
http://sciencexp.free.fr/index.php?perma=Mensurasoft-PB
- [10] Pilot demonstration:
"http://sciencexp.free.fr/index.php?perma=pilotes_demonstration"
- [11] Mensurasoft: drivers of measurement devices and actuators system under MS-Windows and Linux:
http://sciencexp.free.fr/index.php?perma=Programmation
- [12] Demonstration applications:
"http://sciencexp.free.fr/index.php?perma=Applications_demonstration"

Previous Publication in FUNKTECHNIKPLUS # JOURNAL

None

*** About The Author**

Pierre Dieumegard, was born in Limoges, France, in 1955. He teaches biology, geology and computer science in Lycée Pothier in Orléans, France. Since his thesis in plant genetics (1984), he started to use computers. After that, for many years, he was involved in research on the use of computers in science education.

pierre.dieumegard@ac-orleans-tours.fr

[This Page Intentionally Left Blank]

[This Page Intentionally Left Blank]

[This Page Intentionally Left Blank]

In case of any doubt,
download the genuine papers from
genuine.ftpj.otoiser.org

FRONT COVER VIGNETTE

A faded synthesis of an anthemion rooted in a meandros

The thirteen-leaf is a symbol for a life tree leaf.
"Herakles and Kerberos", ca. 530–500 BC,
by Paseas, the Kerberos Painter,
Museum of Fine Arts, Boston.

www.mfa.org/collections/object/plate-153852

The simple meandros is a symbol for eternal immortality.
"Warrior with a phiale", ca. 480–460 BC,
by Berliner Maler,
Museo Archeologico Regionale "Antonio Salinas" di Palermo.

commons.wikimedia.org/wiki/File:Warrior_MAR_Palermo_NI2134.jpg