

BUILD YOUR OWN CAPILLARY ELECTROPHORESIS INSTRUMENT WORKSHOP

BOOK OF ABSTRACTS

Hotel Continental, Brno, Czech Republic, October 15-16, 2018

Instrumentino – A practical approach for fast setup, control and monitoring of experimental CE instruments

Roland Fuiko¹

¹ TU Wien, Institute for Water Quality and Resource Management, Karlsplatz 13/E226-1, 1040 Vienna, Austria

Email: rfuiko@iwag.tuwien.ac.at

Abstract

During scientific research in the field of capillary electrophoresis (CE), many expert groups faced the need to build and control their purpose made experimental set-ups. Depending on the complexity of the experiments, a wide variety of different hardware from different vendors must be combined to an integrated composition. On the one hand from the fluid side to the other in an appropriate electronic way. The most common hardware interface these days is a USB connector, but also RS232/485 and analog interfaces are prevalent. To automate the experiments for autonomous operation the use of a personal computer is reasonable and plays the role of a mediator between hardware and software control. The most common interface on a PC, to enable hardware communication, is only a USB port. For analog and digital input and output, additional hardware is required to meet this challenge. In the scientific community a new trend emerged in recent years controlling self-made experiments with the “Arduino” open-source hardware platform^{1,2}. The core of an Arduino board is a microcontroller on a standardized circuit board that connects to different components of experimental systems to enable monitoring and control abilities. The widespread acceptance of the Arduino platform stems from the corresponding integrated development

environment for easy programming compiling and uploading code to the microcontroller. Countless code examples and straightforward support is available through the constantly growing user community.

The drawback of using the Arduino platform is the limited ability to interactively control and monitor processes. The control code is stored inside the microcontroller and can only be modified by uploading a recompiled version. To overcome this limitation the research group of Peter C. Hauser (University of Basel, Department of Chemistry) identified the need for an easy-to-use and adaptable graphical user interface (GUI) to control purpose-made experimental systems and developed the Instrumentino software^{3,4}. Instrumentino is a user-friendly open source Python⁵ framework and released under GNU General Public License. Downloading and also contributing is possible at the GitHub repository⁶

One part of Instrumentino is the Controlino sketch (controlino.ino) that needs to be uploaded on the Arduino. This sketch enables a textual master / slave communication between Instrumentino and the Arduino board over a USB connection. The complete communication overview is shown in Figure 1.

for interaction and debugging. For example to set the digital pin 2 on the Arduino to high state, define pin 2 as output pin with the command “set D2 out” followed by “write D2 digi high”. When successful the Arduino response is “Done!”. The so far implemented command strings can be found in ³.

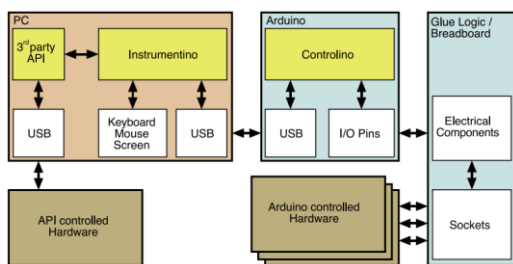


Figure 1: Data flow using instrumentino in purpose-made experimental CE setups (reprinted with permission from I.J. Koenka)

The Instrumentino Python code generates the user front-end for communication with Controlino and acts as the master in the defined communication protocol. During initial setup, the user needs to configure the system description file in Python, which keeps all controlled system components and their connection ports to the Arduino microcontroller (e.g. solenoid valve, connected to Arduino on pin 5 to switch it on and off). All system actions can also be defined here, which can then be executed in the GUI (e.g. close the valve, wait X seconds

and then open it again). The GUI features three sections to gain full control and monitoring over the corresponding hardware (see Figure 2). Far left is the section for direct manual control of components (e.g. setting voltage, pressure...). In the middle section, the user can define and save methods out of system actions and run them as sequences for a defined number of repetitions. The log panel on the right side enables logging of all system actions with execution timestamp. The log-file is stored locally and acts as a lab journal to document conducted experiments with all settings and parameters. The second mode of the panel is the signal-log, a timeline graph that draws all measured signals during method execution.



Figure 2: Instrumentino user interface (GUI) to control and monitor CE experiments (reprinted with permission from I.J. Koenka)

Acknowledgements

Special thanks to Israel Joel Koenka (University of Basel, Department of Chemistry) the inventor of this excellent

software, also for his professional support during the implementation of Instrumentino at TU Wien.

References

1. D'Ausilio, A. Arduino: A low-cost multipurpose lab equipment. *Behav Res* **44**, 305–313 (2012).
2. Pearce, J. M. Building Research Equipment with Free, Open-Source Hardware. *Science* **337**, 1303–1304 (2012).
3. Koenka, I. J., Sáiz, J. & Hauser, P. C. Instrumentino: An open-source modular Python framework for controlling Arduino based experimental instruments. *Computer Physics Communications* **185**, 2724–2729 (2014).
4. Koenka, I. J., Sáiz, J. & Hauser, P. C. Instrumentino: An Open-Source Software for Scientific Instruments. (2015). doi:info:doi/10.2533/chimia.2015.172
5. Rossum, G. *Python Reference Manual*. (CWI (Centre for Mathematics and Computer Science), 1995).
6. Koenka, I. J. Instrumentino, <https://github.com/yoelk/Instrumentino>.