The basics

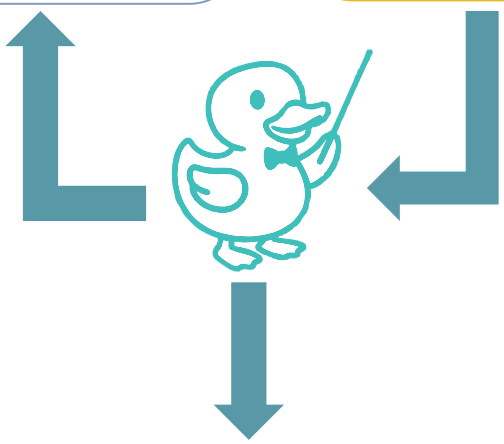
What Data Acquisition software does

Modify and **monitor** experimental settings (turning knobs)

- Voltage value
- Motor position
- Laser wavelength
- Electronics timing and pulse shape

Perform experiment **measurements** (collecting data)

- Take a 2D image
- Read a 1D graph (oscilloscopes, ...)
- Measure a number (current, pressure, ...)



Log
Settings + Data

Display live for user to overview

Save in a file for later analysis

Problems and solutions

The current situation vs the goal

Current problems

Isolated efforts
Multiple research groups trying to solve the same problems.

Fragile code
Most of the knowledge is lost when the PhD/postdoc that built the software leaves the group.

Unguided learning
The programming language (LabView) is a proprietary graphical environment. Free tutorials are scarce and aimed at industry, not research.

How conDAQtor helps

Shared progress
Mistakes and successes in one research group help others.

Documentation
Crucial knowledge is well preserved, saving future time and resources.

Platform and workshops
Curated platform with key knowledge and tailored workshops, to teach the basic LabView skills researchers need.

SOFTWARE ISSUES

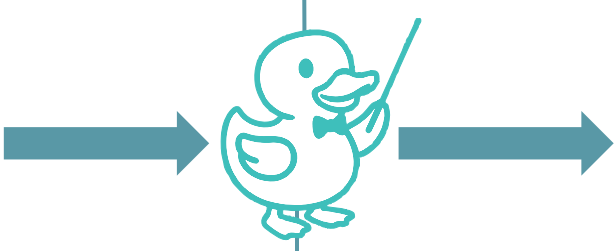
CONSEQUENCES

New science is limited
Software only allows to do the exact experiment it was originally designed for.

Inefficient lab work
Researchers spend time manually adjusting experimental settings.

Flexibility for improvements
The software can evolve together with new experimental ideas.

Automatized experiment
Researchers waste minimal time on repetitive tasks in the lab.

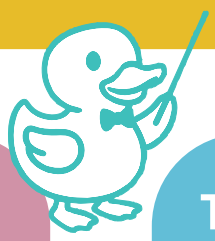


What the project offers

Flexible support

Direct support

- Short: troubleshooting, identifying what's feasible ...
- Long: collaborate on difficult modifications to the code, carefully document it, suggest improvements ...



Knowledge transfer

- Documentation of lessons learnt from working with various research groups.
- Basic examples in a collection of simple code blocks to be easily reused.

Trainings

- Online platform with tips and tricks that save hours of searching.
- Tailored workshops to teach researchers what they need, not less and not more.

A positive feedback loop: being in direct contact with researchers allows to tailor the support and resources.

What the project needs

Basic infrastructure to deliver support

One full-time position

- Research Software Engineer (32 h)

Budget for personal development

- Continuous learning in software development, project management, etc.

Visibility of the project

- Presence at conferences, contact with universities, etc.

Basic (remote) office space

- Most support can be provided remotely by having online access to the lab PC and the collaboration of a researcher on-site for specific tests.
- This saves the costs of travelling to each lab that requires support.

Minimal inexpensive hardware

- To emulate the behavior of lab equipment.
- For developing the training resources, small mockup programs, test, etc.

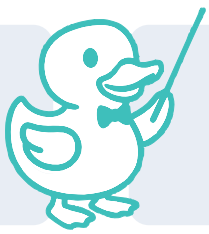
Biggest advantages

- Only one position is needed to support multiple research groups.
- Most of the work can be done remotely in a standard office setup.

What I can bring to the project

Balance of research and software

Research	Software
Experience as experimentalist	Experience with DAQ software
Fully understanding researchers' requests	Essential knowledge on LabView language
Suggesting how software can help the experiment	Ability to quickly learn new coding skills

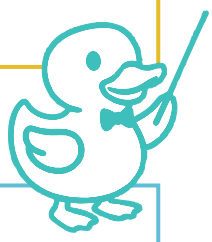


Next steps

Conversations with all parts involved

Researchers

- Conduct survey among researchers to identify more precisely how they would benefit from the project.
- Contact some PIs to ask if they would be interested in future collaborations.
- Showcase project at a stand at national physics conference (if funded).



Research Software Engineers

- Contact Digital Competence Centers, eScience Center, SURF, etc. for feedback and to discuss potential collaboration.

Funding agencies

- Contact funding agencies (e.g. NWO) for information on what calls could support the project.

The language behind most DAQs

NI LabView

LabView is a graphical programming environment owned by National Instruments (Emerson).

Most instruments manufacturers include a LabView library in their Software Development Kit.

It is widely used in industry for automatization and testing, and has become one of the most used languages for Data Acquisition software in research.