The Journal of Open Source Software

# MAHOS: Measurement Automation Handling and Orchestration System

**Kosuke Tahara** [1]

**1** Toyota Central R&D Labs., Inc., Japan

**DOI:** 10.21105/joss.05938

**Software**

- Review
- Repository
- Archive

**Editor:** Arfon Smith

**Reviewers:**

- @sidihamady
- @aquilesC

**Submitted:** 27 July 2023
**Published:** 11 November 2023



## Summary

Experimental researchers often face challenges in building systems to automate their measurement procedures in physical science and engineering, or related areas. While simple automation could be done by a single script, systematic orchestration is required to handle complex cases. The measurement automation typically requires several pieces of programs in diverse layers: 1) low-level drivers to communicate with the instruments, 2) high-level measurement logic or analysis algorithms, and 3) graphical user interfaces (GUIs) or top-level automation scripts.

A `modular` framework is necessary to write maintainable programs for this purpose. In such a framework, the pieces of code are organized within high-level abstraction mechanisms (module, class, etc.) and the interfaces are clearly defined.

`Distributed` messaging is one of the modern approaches for building flexible computing or control systems. In this approach, individual programs run on multiple computers and they exchange the data by using a messaging mechanism. Each program works in concert with the other to realize the system's functions.

Although there are several `modular` frameworks for laboratory automation, the `distributed` messaging approach is not widely recognized and tested in this field. `MAHOS` tries to bring these two concepts together and provide a `modular` and `distributed` framework for measurement automation.

## Statement of need

`MAHOS` is a `modular` and `distributed` framework for building measurement systems in Python. Python has been a major language of choice for laboratory automation. The low-level binding libraries for instruments, such as PyVISA (Grecco et al., 2023) and PyDAQmx (Cladé, 2023), are available for Python. There are also several projects aiming to provide a comprehensive `modular` framework ranging from the instrument drivers to the GUIs (Binder et al., 2017; Feder et al., 2023; Jermain et al., 2020). While we believe that the `distributed` messaging could help create flexible and accessible systems, the existing libraries take rather centralized approaches. That is why we have developed `MAHOS` to provide both a `modular` and `distributed` framework for measurement automation.

On the `MAHOS`, several programs run as different processes and exchange the data by using a distributed messaging system. Two distributed system concepts are inspired by the ROS project (Quigley et al., 2009). First, each piece of the program is called a `node`. Second, two patterns of communication methods are provided for the nodes: Request-Reply (synchronous remote procedure call) and Publish-Subscribe (asynchronous data distribution). The ZeroMQ library is currently utilized as the messaging backend (The ZeroMQ authors, 2023).

Contrary to the ROS, the `MAHOS` library is kept rather small and simple. Creating a new

node is as easy as defining a Python class and no build process is required. The system is considered `static` reflecting the purpose of measurement automation, i.e., we assume to know all the possible nodes and messages beforehand. This assumption enables the transparent configuration of the whole system; it can be written in a single TOML configuration file.

Since the messages can be delivered across computers even with different platforms (operating systems), we can build up a multi-computer system flexibly. For example, if an instrument has a driver only for Windows but we have to use Linux for the rest of the system, we can run a driver node on the Windows host and access it from the other nodes on the Linux. All it takes to move a node from one host to another is to change a few lines in the configuration file.

High data and service accessibility are the best benefits brought to `MAHOS` by adopting the distributed messaging approach. Notable examples can be listed below.

- The data published by a measurement logic (node in 2nd layer defined in Summary) can be simultaneously A) visualized in a GUI (3rd layer) node and B) processed and analyzed in an interactive console such as IPython or Jupyter.
- It is straightforward to conduct an initial measurement using a GUI node, and then write an ad-hoc automation script to run many measurements with different parameters because the GUI node and the script are seen as equivalent clients from the measurement logic node. The GUI will visualize the data even when the measurement is started by the script.
- We can inspect the instrument's status or perform ad-hoc operations on the instrument at run time without shutting down any nodes. This is because the operation requests from measurement logic nodes and ad-hoc ones are equivalent from the perspective of the instrument driver (1st layer) node.

The transportation overhead could be the downside of networked messaging if one deals with very large data such as high-resolution images produced at a high rate. However, this overhead can be reduced significantly by running the relevant nodes as threads in a single process and using intra-process transportation. This switching can be performed by only modifying and adding several lines in the configuration file thanks to ZeroMQ providing both networked (e.g. TCP) and intra-process transportation.

Along with the base framework above, `MAHOS` currently comes with a confocal microscope and optically detected magnetic resonance (ODMR) measurement implementations for research of solid-state color centers such as the Nitrogen-Vacancy (NV) center in diamond (Gruber et al., 1997; Jelezko et al., 2004; Wrachtrup & Jelezko, 2006). The covered functionalities are similar to that of the qudi project (Binder et al., 2017).

## Acknowledgements

We acknowledge Prof. Mutsuko Hatano and Prof. Takayuki Iwasaki for their support in developing the predecessor project at Tokyo Institute of Technology. We are also grateful to Wataru Naruki, Kosuke Mizuno, and Ryota Kitagawa for contributing to and maintaining it. We thank Katsuhiro Kutsuki and Taishi Kimura for the opportunity and support to start this project.

## References

Binder, J. M., Stark, A., Tomek, N., Scheuer, J., Frank, F., Jahnke, K. D., Müller, C., Schmitt, S., Metsch, M. H., Unden, T., Gehring, T., Huck, A., Andersen, U. L., Rogers, L. J., & Jelezko, F. (2017). Qudi: A modular python suite for experiment control and data processing. *SoftwareX*, *6*, 85–90. https://doi.org/10.1016/j.softx.2017.02.001

Cladé, P. (2023). PyDAQmx. In *GitHub repository*. GitHub. https://github.com/clade/PyDAQmx

Feder, J., Bourassa, A., mtsolmn, bensoloway, jamendez99, mendezUC, & Jones, A. (2023). *nspyre-org/nspyre* (Version v0.6.1). Zenodo. https://doi.org/10.5281/zenodo.8396120

Grecco, H. E., Dartiailh, M. C., Thalhammer-Thurner, G., Bronger, T., & Bauer, F. (2023). PyVISA: the Python instrumentation package. *Journal of Open Source Software*, *8*(84), 5304. https://doi.org/10.21105/joss.05304

Gruber, A., Dräbenstedt, A., Tietz, C., Fleury, L., Wrachtrup, J., & Borczyskowski, C. von. (1997). Scanning Confocal Optical Microscopy and Magnetic Resonance on Single Defect Centers. *Science*, *276*(5321), 2012–2014. https://doi.org/10.1126/science.276.5321.2012

Jelezko, F., Gaebel, T., Popa, I., Gruber, A., & Wrachtrup, J. (2004). Observation of Coherent Oscillations in a Single Electron Spin. *Physical Review Letters*, *92*(7), 076401–076401. https://doi.org/10.1103/PhysRevLett.92.076401

Jermain, C., minhhai, jmittelstaedt, dennisfeng2, StePhanino, unclemoe, Rowlands, G., Girard, H.-L., Schippers, C., Schneider, M., chweiser, Buchner, C., Spirito, D., Feinstein, B., nowacklab-user, Tiger, S., Vaillant, G. A., Boxtel, T. van, Chan, M.-A., … chrische-xx. (2020). *PyMeasure* (Version v0.8). Zenodo. https://doi.org/10.5281/zenodo.3732545

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. (2009). ROS: an open-source Robot Operating System. *IEEE International Conference on Robotics and Automation Workshop on Open Source Software*.

The ZeroMQ authors. (2023). *ZeroMQ: An open-source universal messaging library*. https://zeromq.org/

Wrachtrup, J., & Jelezko, F. (2006). Processing Quantum Information in Diamond. *Journal of Physics: Condensed Matter*, *18*(21), S807. https://doi.org/10.1088/0953-8984/18/21/S08