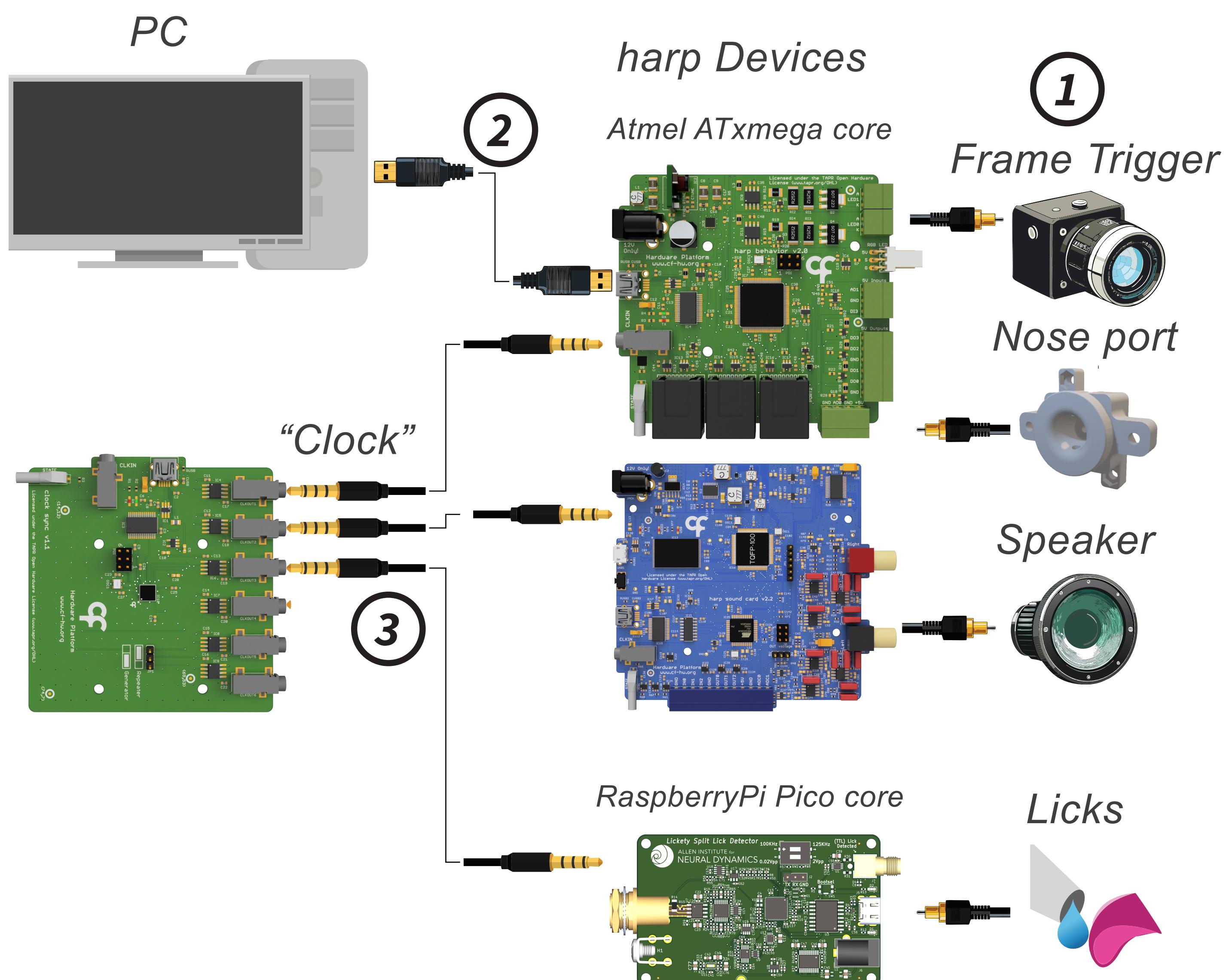


## SUMMARY

## A standardized hardware solution for:

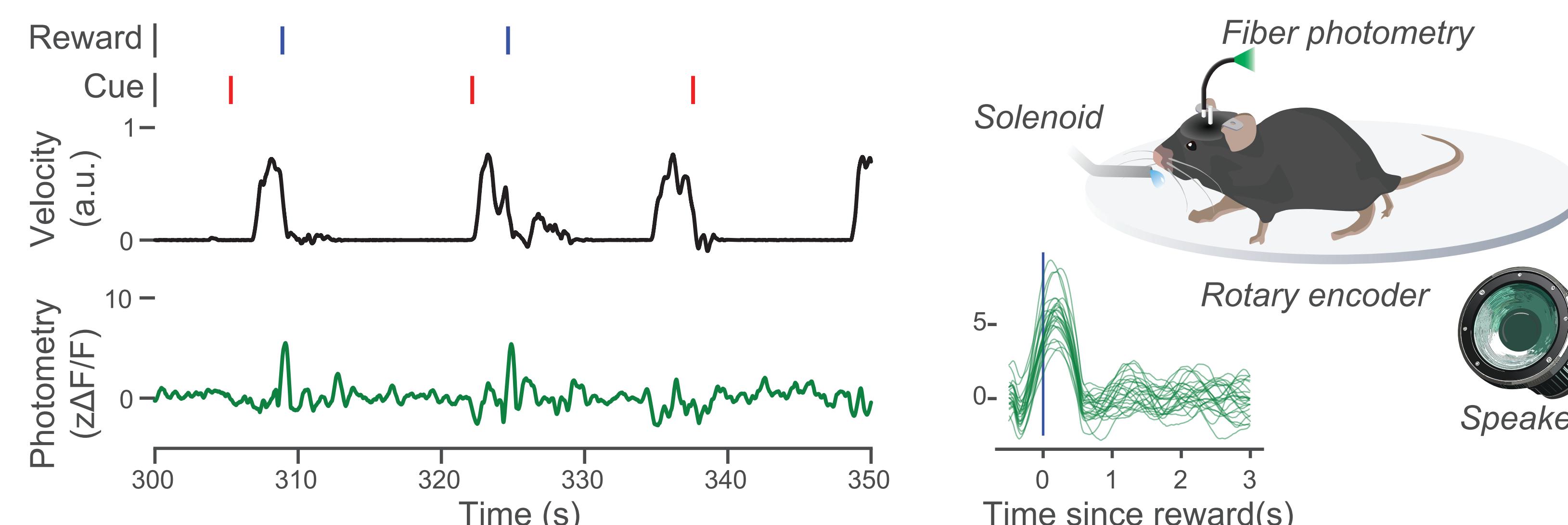
- Automatic sub-millisecond synchronization among all devices
- A binary protocol for communication between devices and PC
- An ever-growing family of devices
- Seamless integration with Bonsai for defining experimental logic
- Easy data ingestion and processing

## Assemble your experiment

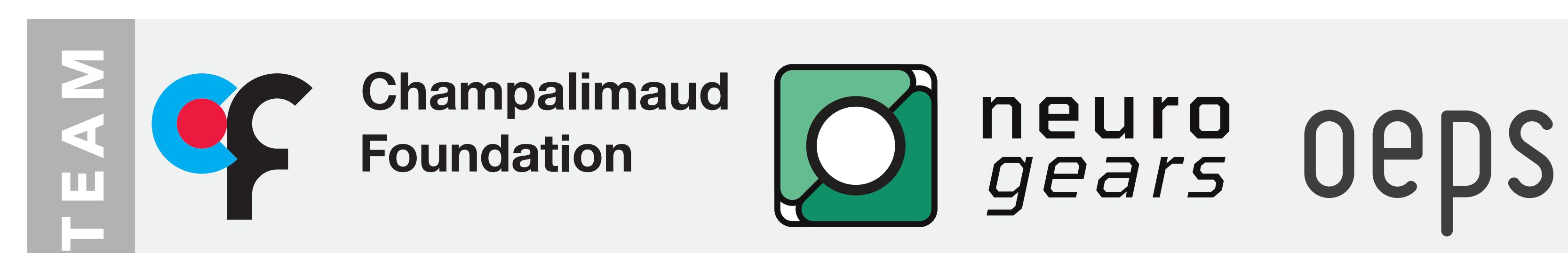


▲ The harp ecosystem. 1) Devices interface with peripherals to measure and control the environment. Time critical operations are handled by the device (2) and relayed to the PC via USB. 3) All devices receive a common "clock" signal that ensures synchronization at acquisition time.

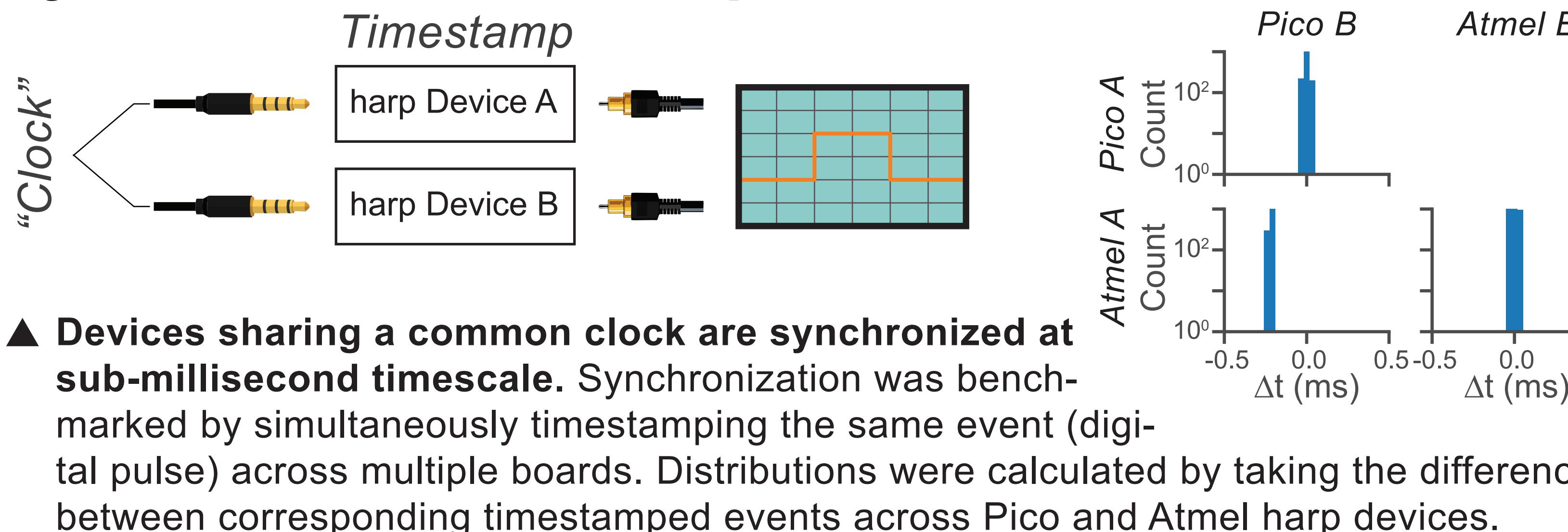
## Accelerate experimental design



▲ Experiment using harp. Mice were required to run on a treadmill (black) and wait for an auditory cue (red), after which reward was probabilistically delivered (blue). In parallel calcium activity from mid-brain dopaminergic neurons was recorded using fiber photometry. Task logic and data acquisition were implemented in Bonsai. Experiment and data collection by Joaquim Silva & Margarida Pinto, Champalimaud Foundation.

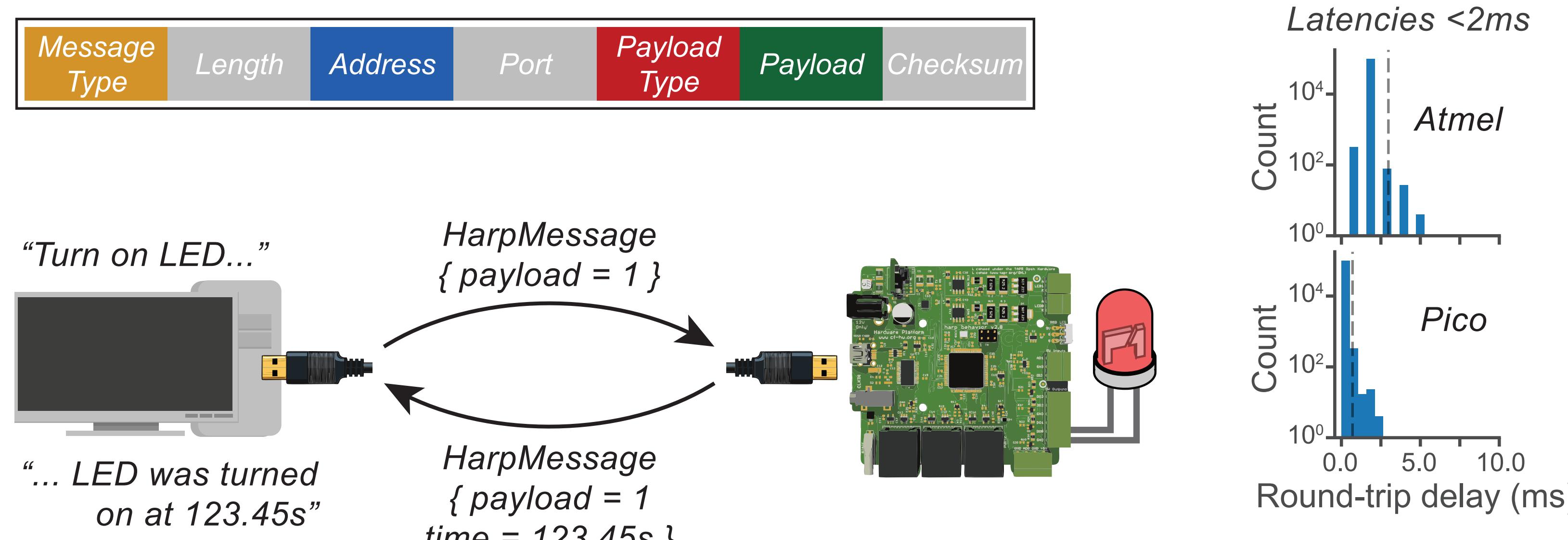


## Synchronization at acquisition time



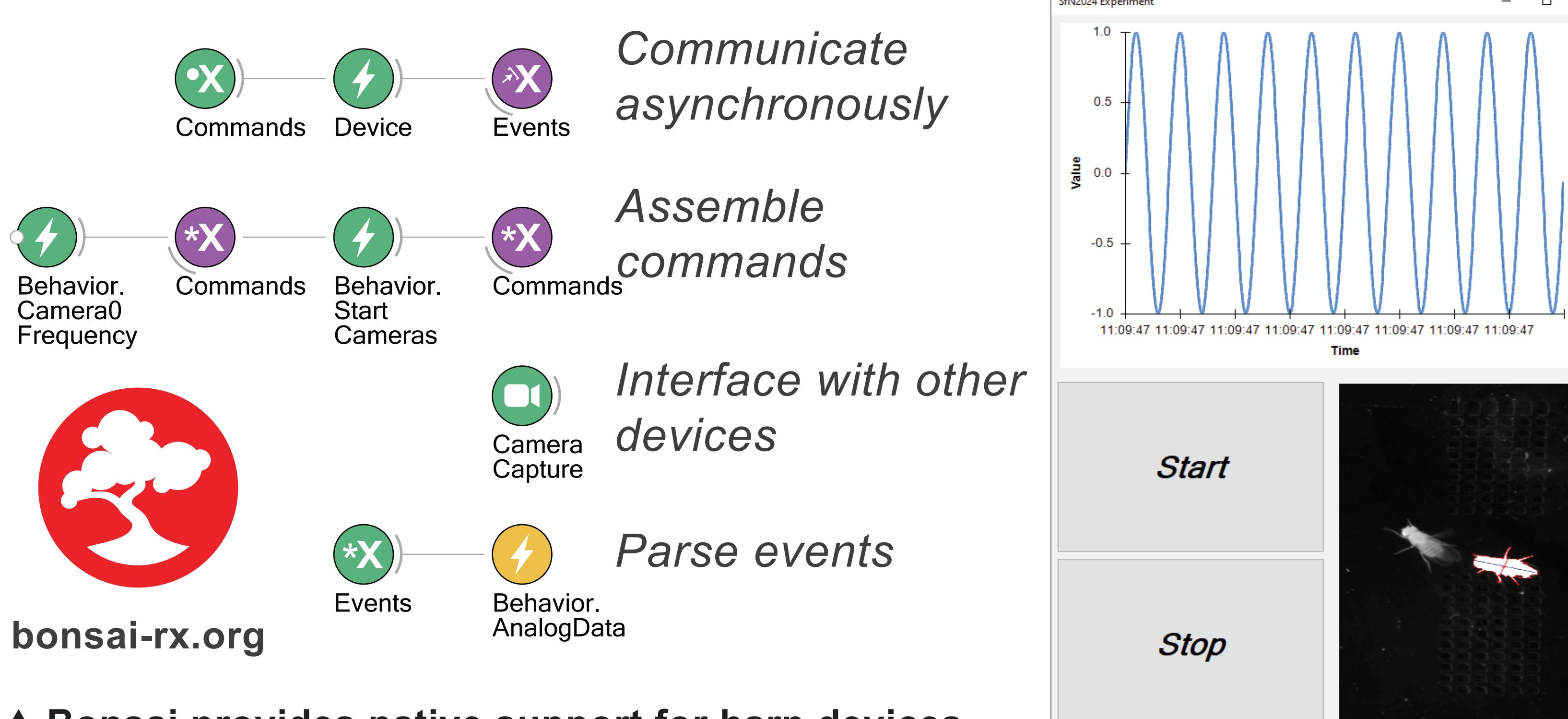
▲ Devices sharing a common clock are synchronized at sub-millisecond timescale. Synchronization was benchmarked by simultaneously timestamping the same event (digital pulse) across multiple boards. Distributions were calculated by taking the difference between corresponding timestamped events across Pico and Atmel harp devices.

## A standard communication protocol



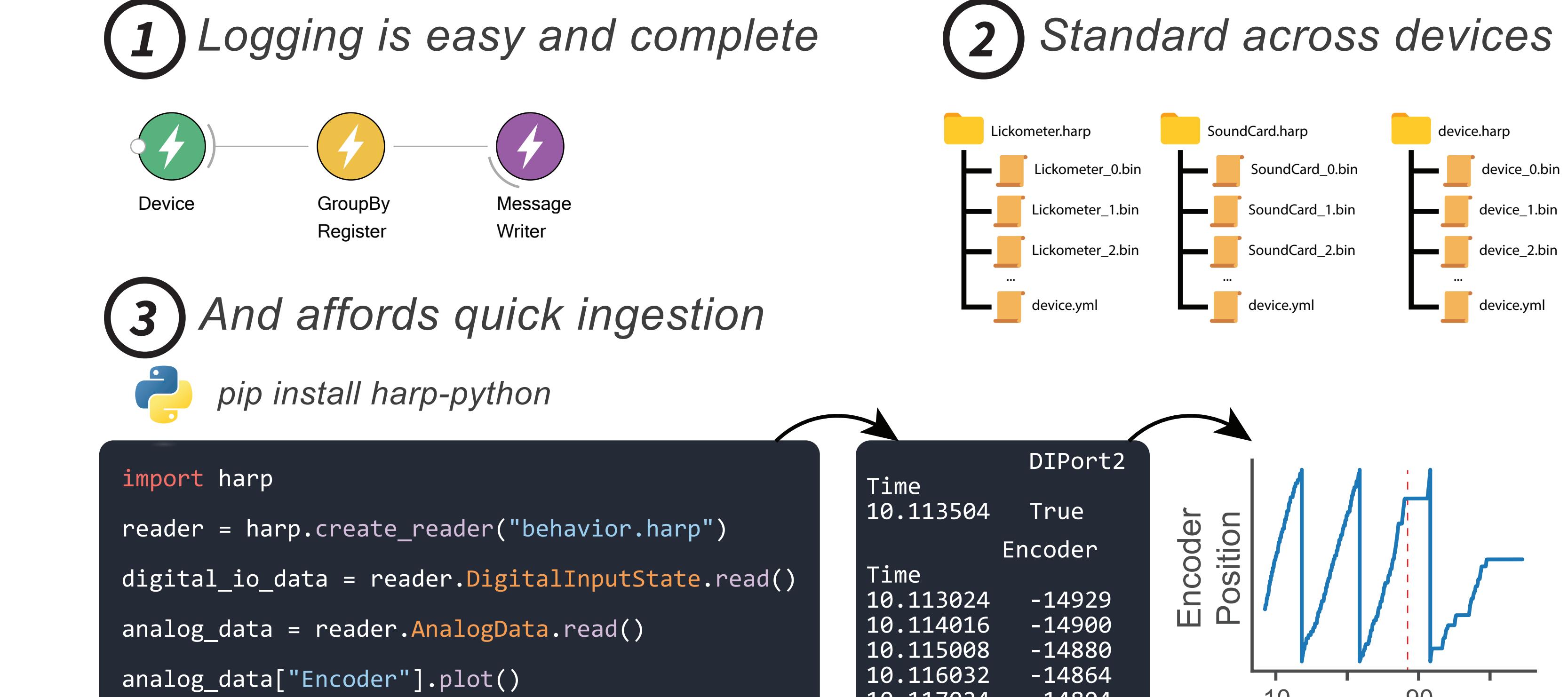
▲ Protocol designed with closed-loop experiments in mind. Devices communicate with the PC via harp messages. This request-reply pattern ensures that all time critical operations are handled and timestamped by the device. Complex task structure can be handled by the PC but still achieves latencies compatible with closed-loop behavior experiments (<2ms).

## Custom device interfaces for Bonsai



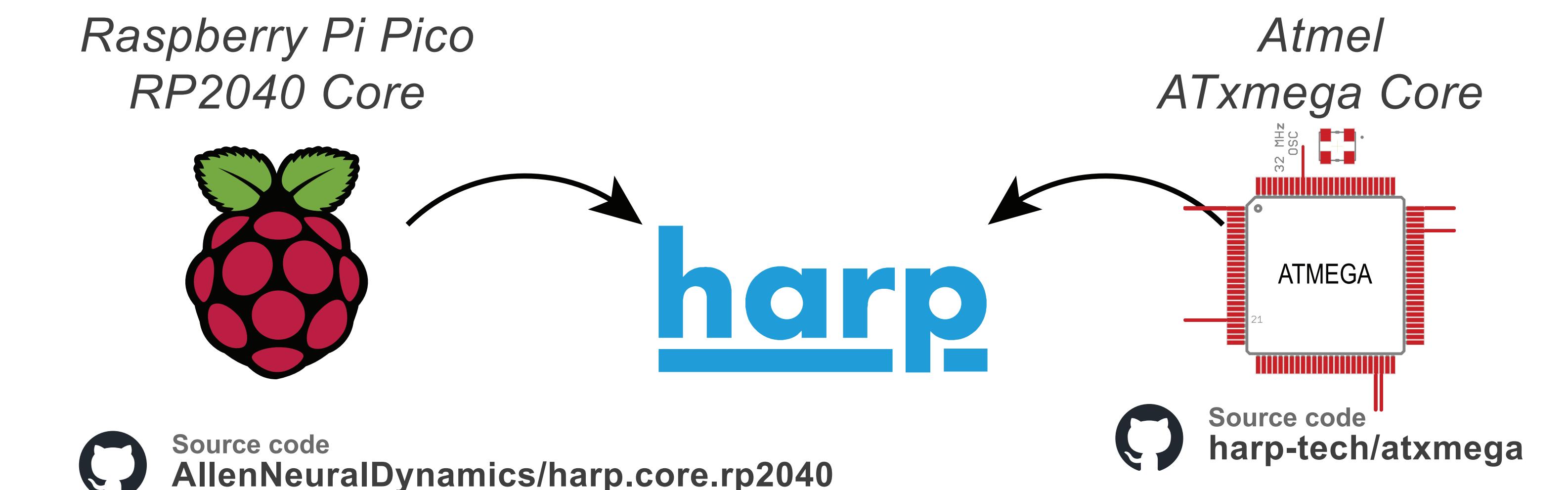
▲ Bonsai provides native support for harp devices. The current implementation supports efficient asynchronous communication between the PC and the devices. This framework is well-suited for online control and manipulation of data from harp devices. By adopting Bonsai, users can also leverage already existing hardware interfaces (e.g. cameras), as well as online processing algorithms (e.g. filtering and pose estimation) and visualization (right).

## Standardize logging and easy data ingestion



▲ Every device message is acquired and saved automatically using existing Bonsai operators (1) in a standard format (2). Leveraging the standard, the harp-python package can be used to quickly load the data for analysis and visualization.

## Hardware templates for new devices



▲ Two device templates are currently provided for accelerating the design of new devices.

## Automatic generation of Bonsai interfaces



▲ A configuration file for each device (device.yml) keeps track of all registers and other metadata. The reflex-generator tool uses this device specification to generate C# code that implements a bespoke interface for the device in Bonsai.

Chat with us and see demos! @ Booth 1753

**harp** Website [harp-tech.org](http://harp-tech.org)



Documentation [harp-tech.org/articles/about](http://harp-tech.org/articles/about)



Community and Q&A [github.com/orgs/harp-tech/discussions](https://github.com/orgs/harp-tech/discussions)

## POSTER

