

# A Standalone, Browser Controlled Microfluidic Automation Platform



Robert R. Puccinelli<sup>1</sup> and Joseph DeRisi<sup>2</sup>

1. UC Berkeley - UCSF Graduate Program in Bioengineering, UCSF 2. Department of Biochemistry, UCSF



## Summary

The miniaturization of assays with microfluidics holds many benefits including the potential for fully automated experiments, favorable molecular kinetics in confined volumes, and access to sample manipulation methods that are otherwise infeasible on the benchtop. However, key challenges with adopting the technology include fabrication and control of the devices. Both of these typically require specialized hardware that most labs don't have access to. The work presented here is a comprehensive microfluidic control platform that seeks to address the limitations of current hardware solutions for controlling valved, multilayer PDMS devices.

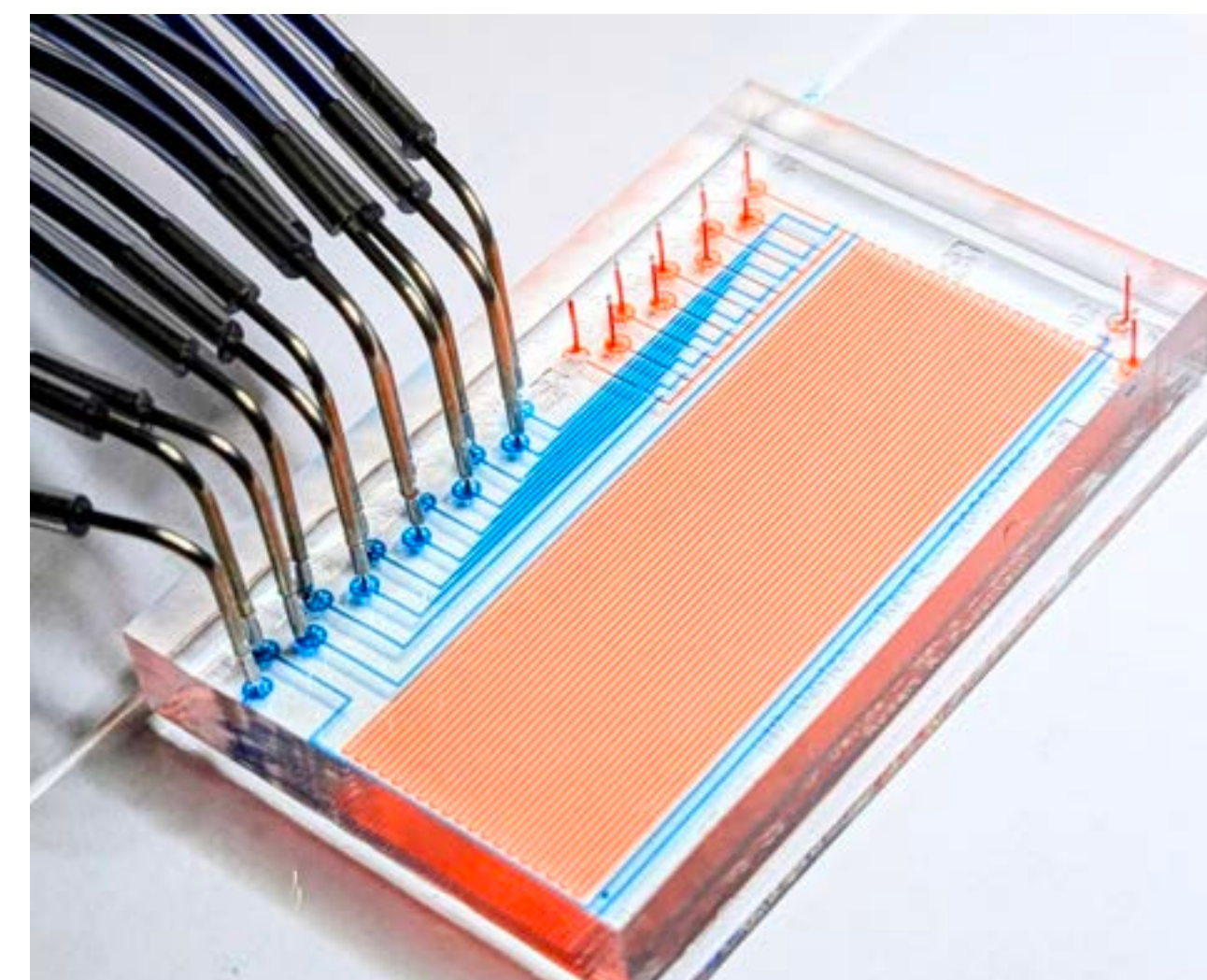
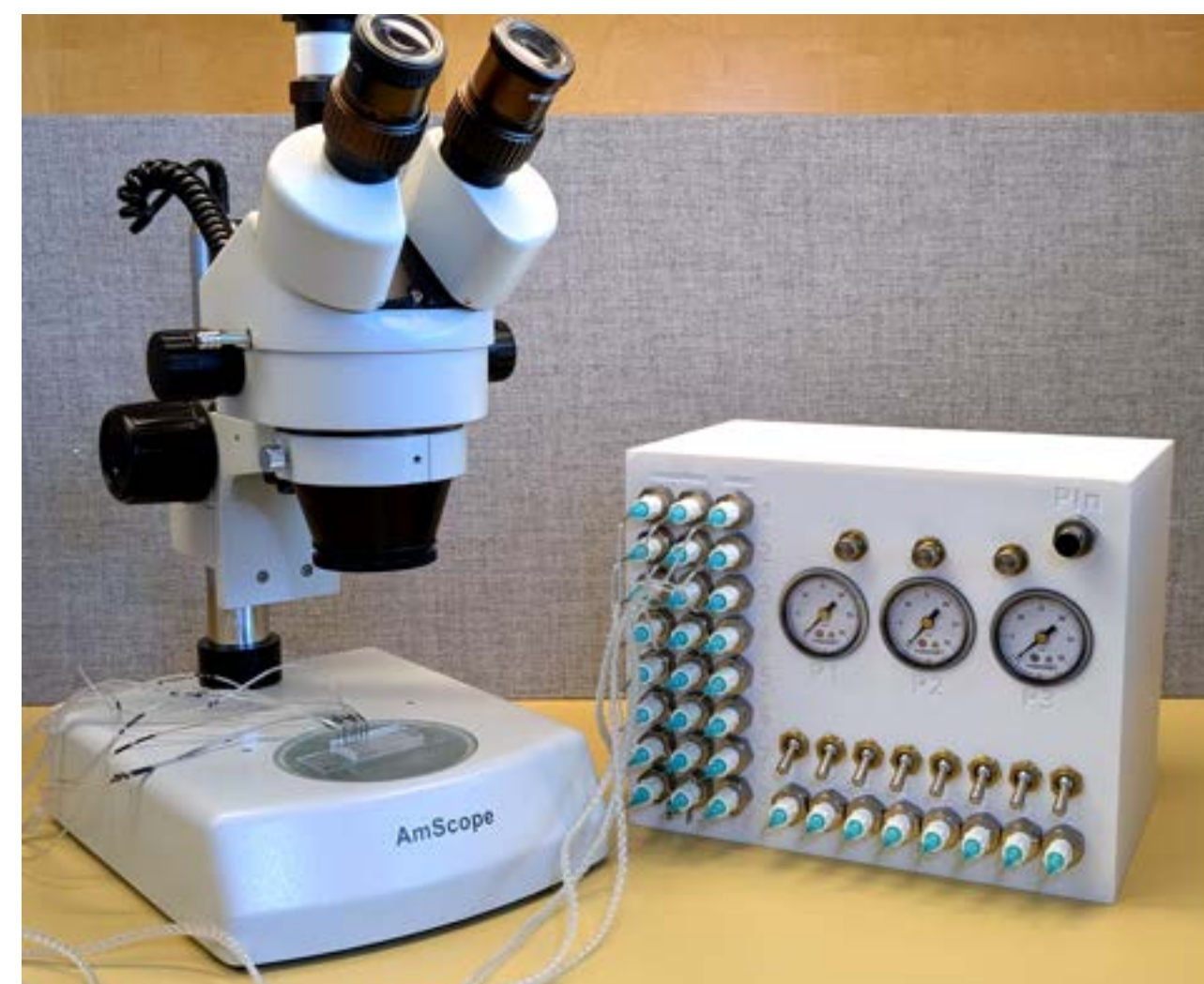


Figure 1. Microfluidic control platform described in this work.

Figure 2. An example multilayer PDMS microfluidic device.

## Challenges

Equipment for pressure driven microfluidic devices are often encumbered by some of the following limitations:

1. Consist of a fragmented collection of hardware
2. Require proficiency in programming languages
3. Contain a fragile dependency on outdated software and drivers resulting in aged PCs sitting around the lab
4. Require a significant space allocation
5. Provide limited logging utilities to ensure that experiments are documented for reproducibility

## Description of Hardware

The microfluidic platform described here is designed to be a modern solution for controlling most Quake valve devices.

1. Has support for 24 solenoid valves and 8 reagents
2. Designed to be remotely controlled via web browser
3. Designed to be modular and extensible
4. Scriptable with a human readable language
5. Logging capabilities for documentation of experiments
6. Occupies less than 1 sqft and costs less than \$1500
7. Does not require a bulky, dedicated PC
8. Developed with a custom OS to ensure software stability and reproducibility of builds

## Pneumatic Design

The pneumatic architecture was engineered for versatility, supporting moderately complex multilayer PDMS devices. The system provides three independently regulated pressure lines (P1–P3) managing 24 solenoid valves and 8 manual toggle switches. P1 and P2 are directed to solenoid valves to facilitate advanced features such as on-chip peristaltic micropumps that require different operating pressures, while P3 is dedicated to reagent pressurization and delivery.

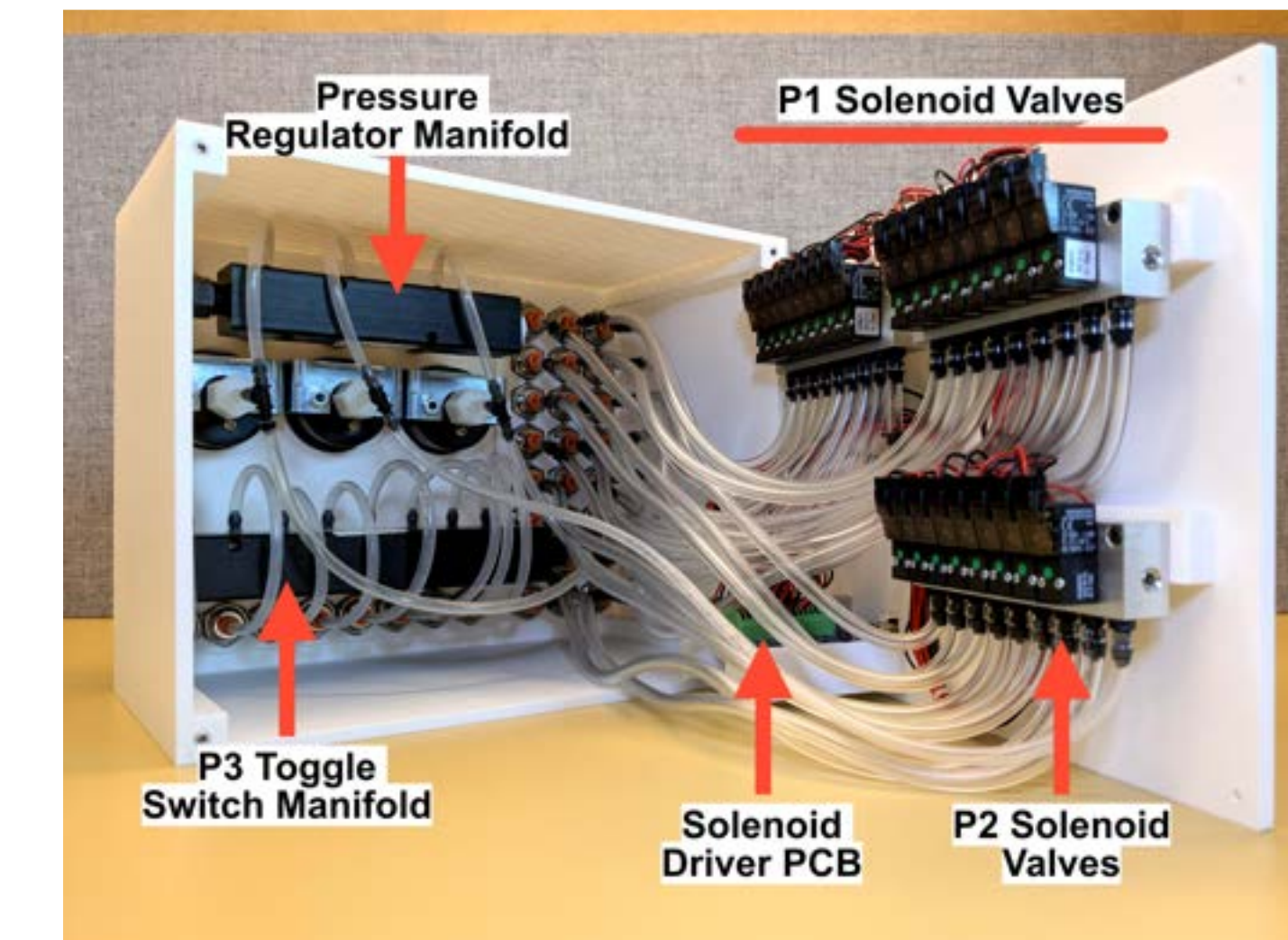
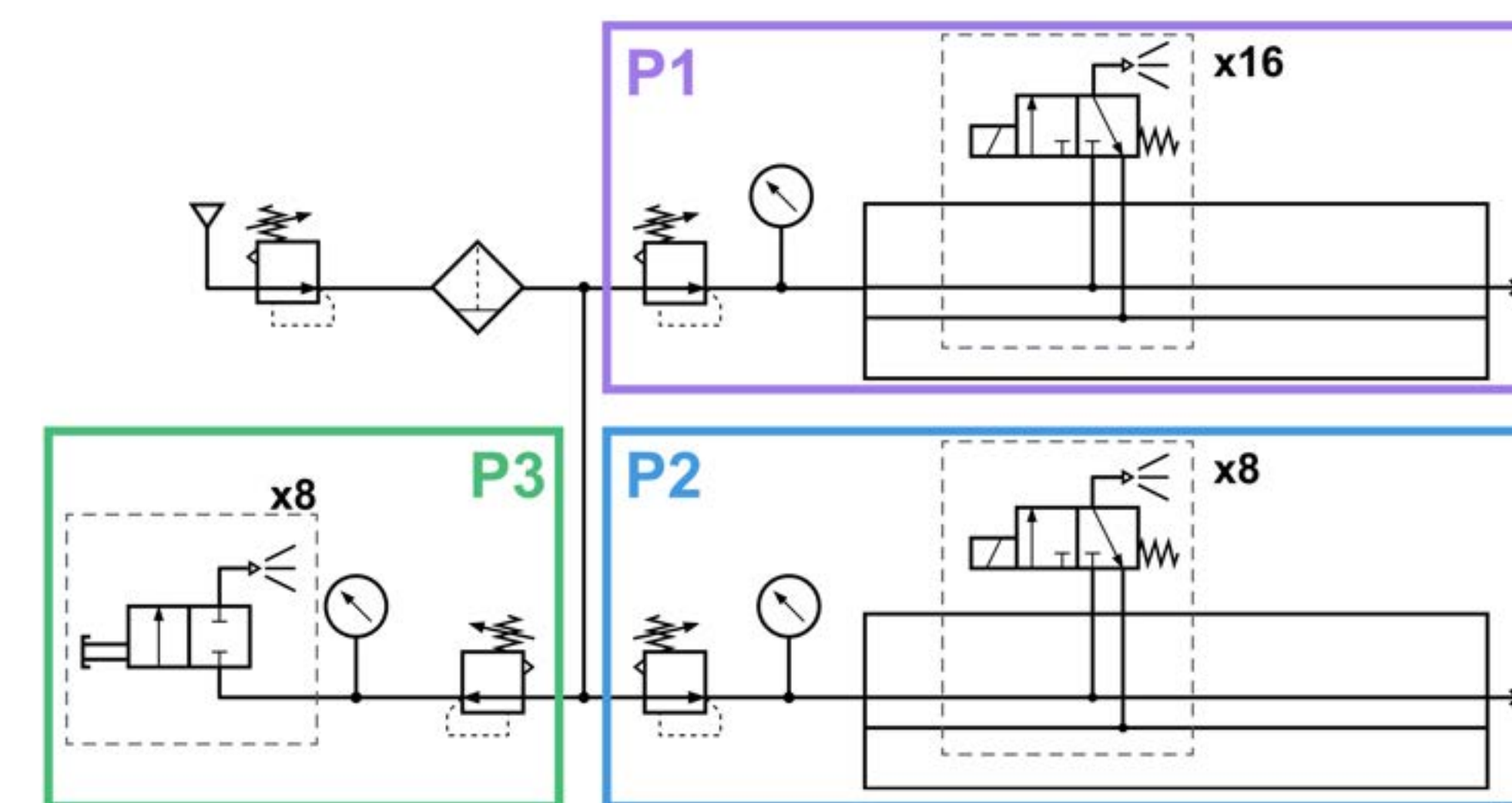


Figure 4. The internal organization of the system is compact and servicable

## Solenoid Driver PCB

The solenoid driver circuit board is a custom solution that allows for advanced diagnostics to facilitate recovery from errors and improve system reliability. The PCB includes support for 24 solenoid valves that can be driven between 5-24V at 4A total and includes a standard USB-C interface. It uses spring clamp wire connectors for stable connections and has extensive on-board diagnostics including open-load detection, current valve states, etc.

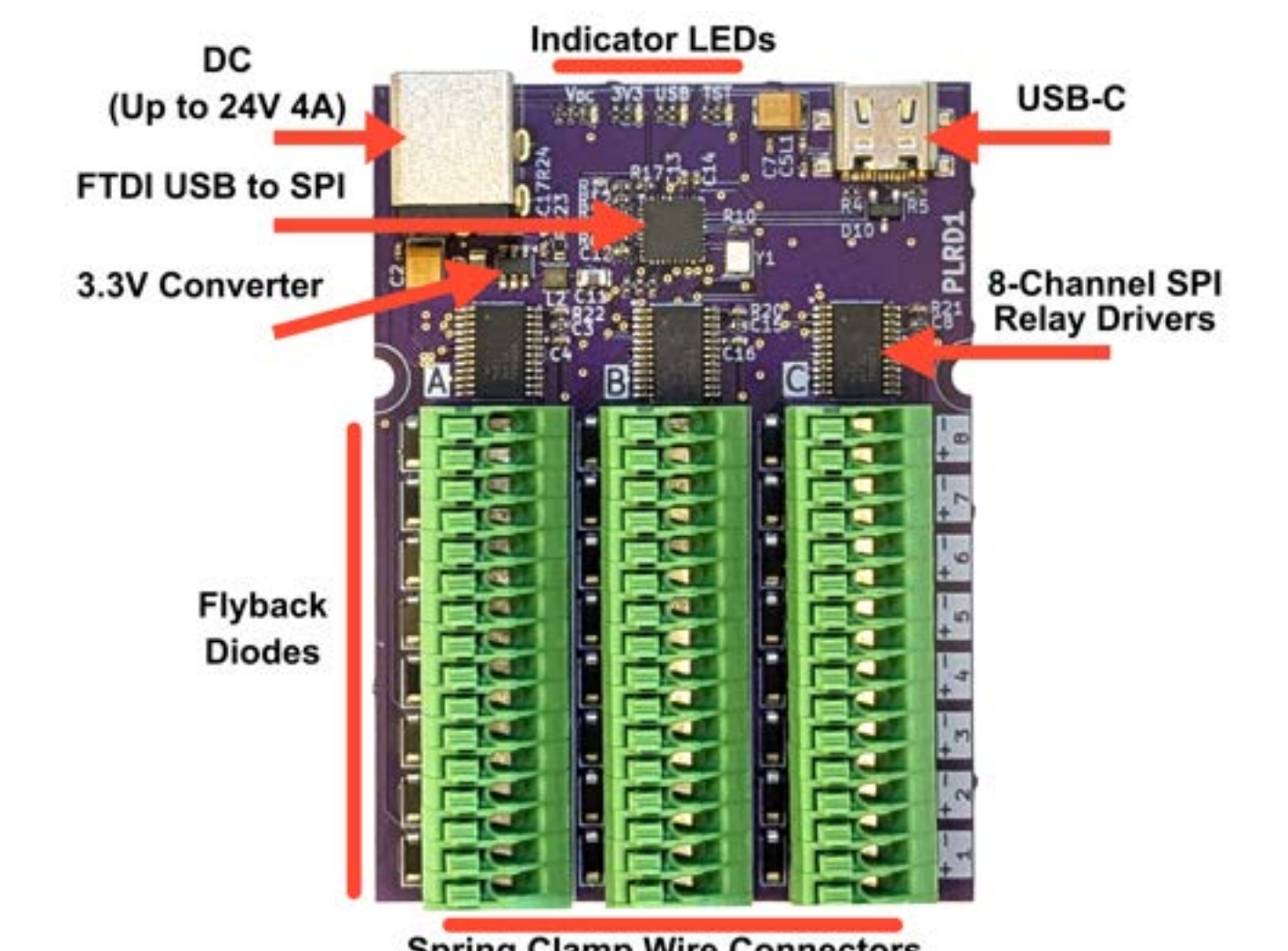


Figure 5. The PLRD1 solenoid driver is a custom credit card sized solution

## Software Architecture

The software was developed with a multithreaded model-view-controller framework to allow for a separation of concerns while being modular and extensible. The application is built on the Flask-SocketIO Python library to allow for real time, bi-directional data streaming between the user's web browser and the application. The system creates a new thread on demand to execute scripts that are designed with a human readable syntax. These scripts are executed on the controller's local hardware and interruptible to allow for adjustments. The system also includes experimental and configuration logging to ensure that experiments are documented and reproducible.

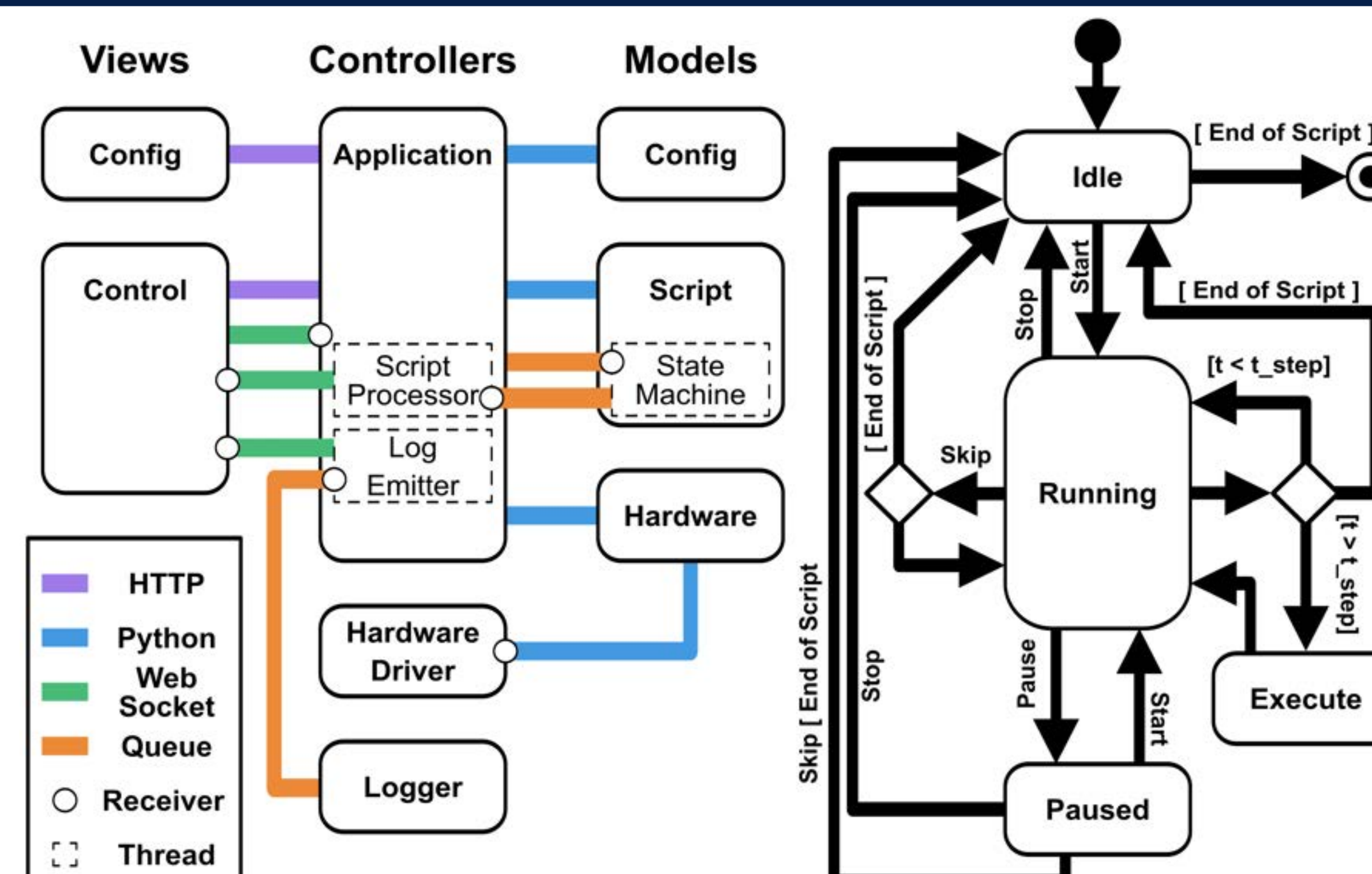


Figure 6. The system leverages a modular model-view-controller framework and a state machine for executing interactive scripts

## Custom OS

This platform can be controlled by any PC, but a custom Linux OS was developed for a Raspberry Pi 4 with Yocto to ensure long term software stability and reproducibility as drivers and systems are deprecated over time. The custom OS also builds in Tailscale so that the machine utilizes secure remote access with zero network configuration required. The OS takes 20 seconds to boot into a kiosk view.

## User Interface

The user interface consists of HTML pages for a configuration and a control view. The configuration view allows for hardware configurations to be created, modified, and loaded. The control view is a custom page that allows for scripting and device operation while streaming data including device logs, states, and expected step and run

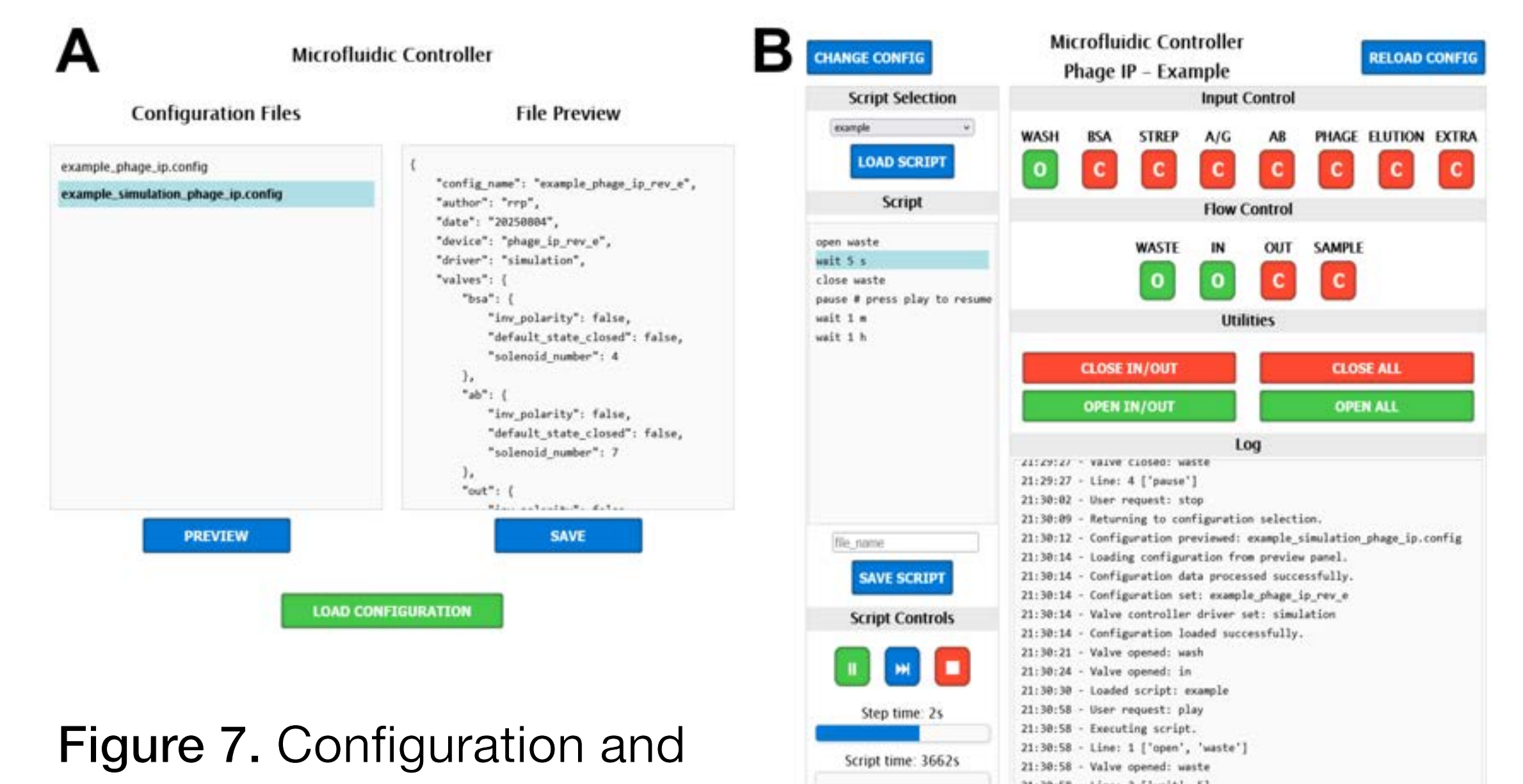


Figure 7. Configuration and control user interfaces



System Documentation Website