Title

Automated Fluidic Platforms for Protein Purification and Molecular Diagnostics

ISBN

9798189298295

Author

Puccinelli, Robert Russell

Publication Date

2026-09-03

Peer reviewed|Thesis/dissertation

Automated Fluidic Platforms for Protein Purification and Molecular Diagnostics

by

Robert Russell Puccinelli

DISSERTATION

Submitted in partial satisfaction of the requirements for degree of

DOCTOR OF PHILOSOPHY

in

Bioengineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO
AND
UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

DocuSigned by:
0CECE2247695423...

Joseph DeRisi

Chair

DocuSigned by:
Iain Clark
DocuSigned by:
Wallace Marshall
43941FCFA7C0447...

Iain Clark

Wallace Marshall

Committee Members

Acknowledgments

My initial foray into the field of engineering was learning how bicycles functioned. A bike repair shop in my university town had sold me tires that were the wrong size for the aged bicycle that was donated to me by some friends, and I spent an evening in my bedroom sawing off the Kevlar beads with a pocketknife. $40 was worth a lot for me while I was a student at University of California, Merced, and seeing it wasted was painful. I was determined to learn how to maintain and fix my bicycle to prevent myself from being a victim of my own ineptitude once more. That little steed served me well until it was stolen off a rack outside the police department at the UC Davis Medical Center in Sacramento a summer or two later. My first acknowledgment is to the singular bike shop in Merced for enabling me to learn how mechanical systems function.

Midway through my second year at the university I also took my first physics course and was drawn to the intuitive nature of solving physical problems. Towards the end of the semester, my lecturer, Carrie Menke, sensed that I was engaged with the introductory course for biology students and she did her best to dissuade me from studying biology and become a physics convert instead. I met her halfway as I was convinced that it wouldn't hurt to pursue degrees in both biology and physics. The remaining years of my undergraduate education are remembered positively as I was able to engage with my mostly international physics cohort who had brought humor, fellowship, and cultural insights into my daily life while also experiencing a shared suffering during our wrangling with various topics in physics. I am very grateful that Carrie Menke took the initiative to persistently nudge me in a new direction and ultimately laid the intellectual foundation for the engineering endeavors that followed.

I initially met my thesis advisor, Joe DeRisi, immediately after my undergraduate studies while I built microarray, microscopy, and microfluidic instrumentation as the first member of the Fordyce lab at Stanford University. Some decades prior, Joe DeRisi had built and documented the construction of a massive room-sized microarrayer that he had used to produce the first full genome microarrays of *Saccharomyces cerevisiae* during his own graduate studies with Patrick Brown. One of my earliest tasks was to replicate the arrayer he designed, which was my first true engineering project and the beginning of the scientific instrumentation arc of my career. The system was complex, ran on Windows 98, and required ancient PC hardware to control. It might not have been the easiest project to start with, but Joe's guidance was a valuable introduction.

Joe later launched the Chan Zuckerberg Biohub with Mark and Priscilla as the co-president of the organization. I emailed him asking if there were any positions available once I saw the launch announcement and he connected me with Rafael Gómez-Sjöberg. Knowing Joe as well as I do now, I am grateful that Joe found the time to read, respond to, and forward my email. I joined Chan Zuckerberg Biohub as the first team member of the bioengineering group where I continued to refine my instrumentation skills. Joe later converted Biohub to CLIAHUB for early pandemic SARS-CoV-2 testing where I was able to build urgently needed instrumentation for producing testing supplies that were then shipped all over the state of California. Although this instrument was simple, it was my first taste of engineering for impact and it left a defining mark on me.

I applied to the UC Berkeley-UCSF Graduate Program in Bioengineering, was accepted, and asked Joe if I could join his lab mid-pandemic. His response was "let's do it". While life during my graduate studies was unpredictably turbulent and challenging, I genuinely appreciate the stable environment and the freedom to approach and explore topics that Joe offered. I would like to thank Joe DeRisi for providing me with the unique opportunities to grow as a scientist and engineer in

both graduate school and beyond. We have known each other for more than a decade, and I am truly grateful to have benefited from his labor and for his impact on my life.

I would also like to extend my gratitude to my thesis committee members Wallace Marshall at UCSF and Iain Clark at UC Berkeley. I first met Wallace as I helped instruct a "Fab-in-Lab" course at UCSF with him alongside Joe and Rafael to teach graduate students how to build instrumentation. Wallace has always brought supportive optimism and creativity to conversations in both the course and my committee meetings. I appreciate his ability to think outside of the box and his challenges to see things from a different perspective. I similarly appreciate Iain Clark's detail-oriented approach to engineering. I learned engineering by doing rather than through formal coursework and I value the balance he offered. When I took over the UC Berkeley microfabrication laboratory as a graduate student instructor, course material that Iain shared proved to be a useful structured resource for the students while also helping me fill in some knowledge gaps regarding other microfluidic implementations that I was not experienced with. The diversity, support, and education that Wallace and Iain offered inside and outside of my committee is appreciated.

As I transitioned to microfluidic assays in my Ph.D., Paul Lum provided an organized and well managed facility as the director of the Biomolecular Nanotechnology Center at UC Berkeley while I spent weeks struggling with developing, and later perfecting, microfabrication protocols. His guidance and generosity were pivotal to the success of my microfabrication learnings. Paul also provided endless support while I taught the microfabrication laboratory course and many humorous stories regarding Hewlett-Packard in its prime and the development of the BART subway system in the San Francisco Bay.

Institutional resources at the University of California, San Francisco are also valued. Victoria Ross was the UCSF program administrator for the joint bioengineering graduate program.

Her quick responses and approachable personality helped me understand and navigate the unfamiliar realm of graduate academics. Victoria also provided opportunities for engagement when I felt disconnected from the program, and I appreciate the effort she invested in looking after my well-being. UCSF had also made resources for physical and mental fitness available, both of which were useful for my journey.

None of the engineering work described herein would have been possible without the mentorship of Rafael Gómez-Sjöberg. When we first met as I was interviewing for his team at Chan Zuckerberg Biohub, all I knew was that I thought microcontrollers were cool and I told him that I wanted to learn how to use them. Years later, I independently developed a feature rich microfluidic controller using a custom solenoid driver printed circuit board, a custom compiled Linux operating system, and an extensive software framework to address instrumentation needs in microfluidics research, as described in Chapter 2. Rafael spent years training me in the technical foundations of engineering that I have used daily as I pursue my interests in instrumentation and I hold a deep sense of gratitude and appreciation for the mentorship that he provided prior to and during my graduate studies.

Outside of my studies, I would like to thank my family. My parents have provided an endless supply of support without judgment as I have stumbled through life. My paternal grandparents have been unchanging caricatures who brought humor to my life while also never failing to ask when I will finish. My maternal grandfather was likely the source of my creativity and engineering inclinations and I valued the conversations we had. My sisters have also continued to make life interesting.

Friends from my time at University of California, Merced have also reminded me that there is more to life than graduate school. I will always appreciate the annual catch ups with Qianting

Chen during the Photonics West conference and the subsequent dim sum outings to spend time with her animated family. Annie Nguyen will always be appreciated for the unfiltered conversations and spontaneous outings. Eva Mo has my gratitude for keeping me connected to the natural world with regular backpacking and hiking trips.

From Stanford, Kara Brower and Nitin Rajan provided support as a dynamic duo with dining that is a bit too fancy for my taste and honest conversations regarding life and career. Their support as peers and academic rigor is inspirational. I would also like to thank Dan Le and Marta Zumwalt. I appreciate the friendship and thoughtfulness they have shared with me as they consistently invited me to join family outings and discuss life as a graduate student.

I would also like to thank my fellow DeRisi Lab members for the countless conversations and support over the years. My nearest neighbor, Elze Rackaityte, frequently brought much needed energy and unexpected queries while I was working at the bench. From other places, I would like to thank Ilakkiyan Jeyakumar for his kindness, cheapness, and the bond of brotherhood. Xiao Gao was a persistently engaging member of my graduate program cohort over the years with stories from his experiences that have almost no parallel to my own life. John Pak never failed to be a pillar of optimism and a source of humorous hypotheticals. As my graduate studies neared their end, conversations with Yuri Lim were insightful and contributed to my personal growth.

The path through my graduate studies was not particularly easy and I hold significant gratitude for the many people who accompanied me along the journey and shared a slice of life.

Pain + Reflection = Progress

There is no avoiding pain, especially if you're going after ambitious goals… If you can develop a reflexive reaction to psychic pain that causes you to reflect on it rather than avoid it, it will lead to your rapid learning/evolving… If you're not failing, you're not pushing your limits, and if you're not pushing your limits, you're not maximizing your potential… If you choose to push through this often painful process of personal evolution, you will naturally "ascend" to higher and higher levels. As you climb above the blizzard of things that surrounds you, you will realize that they seem bigger than they really are when you are seeing them up close; that most things in life are just "another one of those".

— Ray Dalio, *Principles: Life and Work*

Contributions

This dissertation is the result of original research outlines the development of automated fluidic control instrumentation and the exploration of autoimmune serological and multiplexed CRISPR-Cas assays. It is organized such that each chapter corresponds to an individual manuscript or a collection of similar work. Figures and partially modified text are derived from the following manuscripts:

Chapter 2:

Puccinelli RR; Sama SS; Worthington CM; Puschnik AS; Pak JE; Gómez-Sjöberg R. Open-source milligram-scale, four channel, automated protein purification system. *PLoS ONE* **19**, e0297879 (2024).

Chapter 3:

Puccinelli RR and DeRisi JL. An extensible, comprehensive microfluidic automation platform. (In Preparation).

Chapter 4:

The work described here consists of a collection unpublished experimental work pertaining to phage immunoprecipitation sequencing and CRISPR-Cas assays. A significant portion of the work conducted builds upon work previously conducted in the laboratory of Joseph DeRisi. Material support for the washing assay development was provided by Caleigh Mandel-Brehm, Colin Zamecnik, Elze Rackaityte, Grace Wang, Aditi Saxena, and Bryan Castillo-Rojas. Material support for the Luminex xMAP assay was provided by Aaron Bodansky, Monica Dayao, Charlotte Hurabielle, Elze Rackaityte, Vida Ahyong, Krista McCutcheon and Colette Caspar. Many others

contributed through numerous conversations and demonstrations. Joseph DeRisi provided funding through a private grant from Chan Zuckerberg Biohub and assisted with project administration.

Chapter 5 concludes with future directions.

PLoS ONE has open-access licenses permitting use and explicitly grant for thesis reproduction with proper citation. Permission for reuse of a figure in Chapter 4 for this thesis is permitted by Elsevier and is derived from Fozouni, P. *et al.* Amplification-free detection of SARS-CoV-2 with CRISPR-Cas13a and mobile phone microscopy. *Cell* **184**, 323-333.e9 (2021).

Automated Fluidic Platforms for Protein Purification and Molecular Diagnostics

Robert Russell Puccinelli

Abstract

The implementation of laboratory automation systems has yielded significant benefits in life science research settings due to reductions in operator induced errors and improvements in assay reproducibility; however, commercial platforms often impose substantial barriers that limit adoptability due to their prohibitive costs and black box ecosystems. In particular, current systems for protein purification and microfluidic control often lack the adaptability and extensibility required to broadly enable high throughput research. Consequently, there remains a critical need for accessible instrumentation that can bridge the gap between complex, labor intensive protocols and standardized, automated workflows. The work described here details the development of several fluidic platforms designed to facilitate laboratory automation and streamline molecular diagnostics. One system is a four-channel parallel fast protein liquid chromatography system capable of parallelized milligram scale purification of proteins, a capability that was leveraged for rapid SARS-CoV-2 protein production during the COVID-19 pandemic. Another system is a compact, low-cost microfluidic control platform that utilizes a multithreaded and modular software framework to flexibly control diverse device architectures. This work further explores methods to automate phage immunoprecipitation sequencing (PhIP-seq) protocols for the detection of autoantibodies and also explores spatially multiplexed CRISPR-Cas assays designed to bypass the spectral limitations of traditional fluorescence-based nucleic acid diagnostics. Collectively, these platforms lower the barriers to advanced instrumentation, enabling broader participation in cutting-edge biological research and accelerating the development of next-generation diagnostics.

Table of Contents

List of Figures

List of Tables

Chapter 1.
Introduction

1.1 Motivation

Preanalytical errors, many of which stem directly from manual sample processing and handling, account for the majority of total laboratory errors and fundamentally compromise assay quality before an analytical instrument can process a sample.[1] A major driver of this variability is the inherent physical inconsistency of manual pipetting. Comparative evaluations across laboratory personnel reveal substantial operator dependent imprecision.[2] Even under controlled conditions, a significant fraction of technicians fail to meet standard ISO precision thresholds, particularly when dispensing low volumes.[3] These volumetric discrepancies do not remain isolated and they propagate throughout complex workflows. In quantitative assays like LC–MS/MS, liquid transfer inaccuracies directly skew internal standard ratios and analytical precision.[4] Furthermore, slight volumetric inaccuracies introduced during serial dilutions can generate a compounding negative bias that shifts dynamic ranges and invalidates bioanalytical concentration.[5,6] In clinical diagnostics such as viral load testing, inter-laboratory variance in manual technique risks shifting quantification results enough to impact diagnostic sensitivity and patient monitoring.[7]

By replacing manual liquid transfers with precise robotic dispensing, automated platforms eliminate operator induced errors while achieving low volumetric variance[8] and significantly enhancing assay reproducibility across diverse applications. The automation of cellular assays like ELISpot has been shown to dramatically reduce inter-operator variability by ensuring uniform cell counts and reagent delivery across high-throughput immunological screens.[9] Similarly, automated liquid handling yields significantly lower intra-run and inter-operator coefficients of variation

compared to manual execution in complex isolation protocols.[10] Transitioning benchtop assays into an automated microfluidic format also yields similar benefits while also accelerating experimental throughput, streamlining complex biological sample processing, and enabling low volume, highly reproducible assays.[11–15] Automation bridges the gap between high throughput demands and data integrity by transforming error prone manual protocols into standardized and reproducible workflows.

Despite the benefits of automating sample handling, a key constraint that limits the adoption of these technologies includes the accessibility of the platforms. Traditional commercial automation systems introduce substantial financial, technical, and supply chain barriers that keep advanced workflows out of reach for many grant-dependent laboratories.[16–20] Cost conscious laboratories often consider open source platforms which have routinely demonstrated that they can be built with a total bill-of-materials that is 10 to 100 fold lower than their commercial analog.[18,21] Additionally, alternative open source platforms can often bypass vendor lock-in or the technical integration barriers that vendors intentionally implement with their black box ecosystems.[19] A cumulation of these benefits is realized in the multi-channel liquid chromatography system described in this dissertation, which was used to simultaneously purify multiple expressed SARS-CoV-2 proteins early in the COVID-19 pandemic that were subsequently distributed globally to facilitate the development of needed diagnostics. Open source platforms are critical for enabling research to be conducted cost effectively and bypass limitations that are intentionally or inadvertently introduced with constrained commercial systems.

1.2 Fluidic automation systems

Protein purification methods have advanced significantly since Antoine Fourcroy began attempting to isolate plant matter with egg whites in 1789. In addition to the development of

filtration by different biochemical properties including size, ionic charge, hydrophobicity and affinity tags in more recent decades,[22,23] state-of-the-art methods often rely on programmable fluidic platforms. Common commercial fast protein liquid chromatography (FPLC) systems include the ÄKTA series from Cytiva and the NGC series from Bio-Rad. Prices for these systems range from roughly $10,000 to $150,000 and vary in functionality between limited low pressure, single column processes to fully automated buffer preparations, multistep purification, and design-of-experiments studies. Studies and technical notes highlight that even entry level automated systems have significant benefits over gravity or spin column purification, and these systems can result in reduced protocol time while improving protein yield and reproducibility with equivalent purity.[24,25] However, these systems also introduce their own limitations. Lower cost systems like the ÄKTA *start* provide non-expandable and fixed hardware configurations that restrict which purification methods can be leveraged while also lacking support for handling multiple samples. Higher cost systems can provide more flexibility at the expense of exorbitant cost, proprietary hardware, and multi-sample purification that is limited to sequential sample processing instead of a parallelized, medium throughput configuration.

Beyond protein purification, the automation of biological and biochemical assays has also been greatly facilitated by leveraging microfluidic technologies. These fields include but are not limited to clinical diagnostics, drug discovery, bioreactors, and single cell analyses.[26] A key component of their success is derived from the physical phenomena that emerge as fluidic reactions are miniaturized to the nanoliter scale.[27] As the scale shrinks, diffusive mixing time scales drops quadratically with channel dimensions from hours in macrofluidic assays to milliseconds which can be critical for maximizing reaction yields.[28–30] Similarly, the surface to volume ratio of the reaction scales inversely with channel dimensions and allows for rapid thermal equilibrium of the

sample which benefits thermal processes such as PCR and maximizes the contact probability between functionalized surfaces and low abundance targets.[29,31–33] Additionally, smaller feature dimensions naturally lend to the conservation of precious reagents and higher density of assays to enable economical, high-throughput and multiplexed studies.[28,32,34,35]

While there are many approaches to fabricate and operate microfluidic devices, early breakthroughs derived from multilayer soft lithography techniques using molds produced via photolithography and devices composed of polydimethylsiloxane (PDMS). These techniques utilize semiconductor fabrication processes to generate 3D microstructures on planar silicon wafers that are then used as a negative mold upon which liquid PDMS would cure.[36,37] By stacking and bonding thin layers of PDMS upon one another, intersections of the molded fluidic channels produce unsupported, flexible membranes that function as fluidic valves when one layer is pressurized to seal the flow of another layer.[38–40] Photolithography processes naturally create rectangular structures, so the pressurization of these elastomeric microvalves would often act as sieves due to the fact that the deflected membrane is unable to seal the corners of the channel being sealed.[41,42] To address this challenge, mold fabrication processes were modified to include a combination of positive and negative photoresists where positive photoresists would undergo a reflow step to generate parabolic channel geometries.[41,43,44] These Quake-style valves allow for complete sealing of fluidic paths due to the absence of channel corners.

Despite the advancement of microfluidic methods and dearth of described assays, adoption of the technology is mostly confined to research laboratories with access to capital and instrumentation engineering resources.[28,45–47] Central to this issue is the diversity of microfluidic device implementations and assay requirements ranging from thousands of valves in high density, parallelized micro-reaction chamber architectures that operate on the scale of days to no valves at

all with millisecond timing requirements in the case of some flow-based architectures.[41,48] As a result, there isn't a single hardware solution that can be easily adopted. Laboratories often build their own ad-hoc solutions that are rarely published for their specific device or they assemble a control system for tens of thousands of dollars using commercial modules.[49] The lack of a centralized and affordable platform that broadly supports pressure-driven and flow-based devices is noticeably apparent and adoptable solutions are needed.

1.3 Serological profiling assays for autoimmune antibodies

Autoantibodies are a fundamental component to a broad set of autoimmune diseases and develop when the immune system loses self-tolerance and acquires self-targeting behavior that can irreversibly damage tissues.[50] While many autoimmune diseases may appear to be symptomatically similar, the diseases are frequently distinguished by a subset of highly specific autoantibodies and they often cannot be effectively managed until the identity of the antibody or epitope are identified.[51] Even without evident symptoms, autoantibodies may be present decades before tissue damage manifests and slowly contribute to the disease.[52,53] To counteract this effect, it is useful for having broadly screening assays for early detection of autoimmune antibodies. Earlier detection of autoantibodies before or during a flare has been shown to lead to better outcomes and lower morbidity.[50,52] However, a particular challenge is that autoantibody fingerprints for a particular disease can be highly heterogeneous and hard to identify with diagnostic assays that are not highly multiplexed.[54] As a consequence, many uncommon subtypes or rare autoimmune diseases do not have commercial diagnostics available due limited commercial incentives with addressing a small market.[55]

Technologies that have been developed to serologically profile the unknown autoantibody fingerprint of autoimmune disease require unbiased, highly multiplexed screening methods and

are typically limited to peptide and protein microarrays and phage immunoprecipitation sequencing (PhIP-seq) approaches.[56–58] With microarrays, up to 20,000 unique proteins or peptides representing the human proteome are spotted onto a functionalized planar substrate which is then incubated with diseased sera and stained with a fluorescent secondary antibody.[57,59] Due to the known spatial encoding of the epitopes, fluorescent spots can be decoded to identify which proteins or peptides are reactive with the sera. This information can then be translated to identify casual pathways and potential treatments for diseases and has been successfully used to characterize many diseases including narcolepsy type 1, type 1 diabetes, systemic lupus erythematosus and others.[57,59,60] Despite the utility, limitations of microarray based serological profiling include difficulty in expressing and printing whole proteins without losing native conformation, synthesizing and solubilizing thousands of peptides, inherent cross reactivity present in the massive sequence diversity of antibodies and peptide sequences, and limited dynamic range.[59,61,62]

PhIP-seq serves an alternative approach that similarly results in data that can be leveraged to identify casual pathways and potential treatments in autoimmune diseases. The method includes constructing a T7 bacteriophage library where each phage has been engineered to express one of more than 700,000 overlapping peptide fragments that collectively represent the human peptidome.[63,64] This library is then incubated with sera and subsequently immunoprecipitated with the use of protein A and protein G magnetic beads, which naturally sequester IgM and IgG isotype antibodies. After several washes, the enriched phage is then amplified in an *E. coli* culture and clarified before the process is repeated to achieve multiple rounds of enrichment and amplification.[63–65] After processing, the immunoprecipitated epitopes are then characterized by sequencing the peptide inserts present in the enriched phage. This approach has been used to characterize a variety of diseases including paraneoplastic encephalitis, autoimmune

polyendocrine syndrome, and rapid-onset obesity with hypothalamic disfunction, hypoventilation, and autonomic dysregulation (ROHHAD) among others.[66–68] Limitations of this platform include the use of a labor intensive protocol that spans several days, the inclusion of many steps that may be implemented differently by different operators, resource intensiveness, and a noisy background with highly diverse sequences that can make it challenging to small signals.[56,65]

1.4 Multiplexed diagnostics with Luminex xMAP microspheres and CRISPR-Cas endonucleases

While microarrays and PhIP-seq are excellent exploratory tools for characterizing the autoantibody profile of autoimmune diseases, the depth of cross reactivity and relatively long assay times are not ideal for a confirmatory diagnostic where quick and confident results could change the treatment path for patients. The Luminex xMAP platform is a spectrally multiplexed, bead based assay that can support the detection of up to 500 analytes simultaneously in several hours and is used routinely for serological profiling.[69] Many commercial panels exist and have been used for the detection of antibodies including but not limited to anti-HLA antibodies in transplant labs to assess rejection risk, anti-cytokine antibodies for immune system dysregulation, and various disease specific autoimmune panels.[70,71] Autoantibody detection panels are also developed in research laboratories to address unmet needs such as lupus nephritis.[72] Given the success of implementing multiplexed autoimmune diagnostics with the xMAP platform, it may be a viable approach for translating signatures discovered through exploratory PhIP-seq methods into a rapid diagnostic and partially address the challenge of the lack of diagnostics for rare disease and subtypes. A prior study that investigated translating epitopes identified via PhIP-seq in multiple sclerosis patients onto Luminex beads indicates that this may be a viable path with further optimization for heterogenous diseases.[73]

Similar to how spectrally encoded microspheres can translate exploratory autoimmune signatures into a diagnostic, multiplexed CRISPR-Cas assays may provide a similar function for infectious disease surveillance platforms.[74–76] Clinically, many infectious diseases can be screened for using qPCR panels; however, each of these panels are frequently limited to roughly a dozen pathogens which limits the depth of the screen.[77] While convenient, screens with limited depth may either miss pathogens completely or incompletely characterize the sample and miss coinfections or resistance markers.[78] As a result, additional screens would need to be performed or ineffective treatments would be administered. Higher throughput screens previously included microarray hybridization methods but have since transitioned to unbiased sequencing technologies to broadly characterize samples.[78,79] Sequencing approaches are powerful and software platforms such as CZ ID readily parse sequencing data directly to provide metagenomic pathogen and antimicrobial resistance detection.[80,81] Drawbacks with these methods however include cost, required technical expertise required for sample processing, and assay time.[79]

Assays that have leveraged several subtypes of CRISPR-Cas endonucleases have recently become an alternative method to PCR for detecting nucleic acid signatures in a sample.[82,83] These endonucleases are dynamically programmed to detect for the presence of a sequence by providing a complementary sequence with a CRISPR RNA to form ribonucleoprotein (RNP) complexes. Once an RNP complex hybridizes to its target sequence in a sample, the conformation of the endonuclease changes and it becomes catalytically active and begins to cleave neighboring nucleic acids. Collateral cleave assay often includes excess nucleic acid fluorophore quencher pairs so that hybridization to target sequences results in cleaving the reporters to generate a fluorescent signal. There are numerous demonstrations of this technology working in practice and was particularly commonly used to develop SARS-CoV-2 diagnostics during the COVID-19 pandemic.[74,84,85]

Multiple studies have detailed methods to enable multiplexing of these endonuclease reactions to detect multiple target sequences in the same sample. In orthogonal Cas multiplexing strategies such as SHERLOCK v2 and HOLMES v2, difference species of Cas endonucleases are mixed into the same reaction and leverage the differential selectivity of reporters by the enzymes.[82,86,87] In the case of SHERLOCK, 4 target sequences were able to be probed for simultaneously by using LwaCas13a, PsaCas13b, AsCas12a, and Csm6 in a 30 minute assay. Each of these RNP complexes have a different preference for a collateral cleave reporter, so the inclusion of 4 unique reporters with different fluorophore-quencher pairs was used to spectrally encode the multiplexed signal. In toe-hold mediated strand displacement (TMSD) methods, multiplexed signals can be generated by using a single Cas enzyme with various TMSD constructs consisting of weakly hybridized fluorophore-quencher pairs that are spectrally multiplexed over a period of 60 minutes.[88] Once the Cas RNP complex hybridizes to a target sequence, it can generate an 8 base pair overhang that is subsequently used to displace the quencher strand. Since each target sequence can have a different overhang, the depth of multiplexing is primarily limited by spectral overlap of the reporters. The CARMEN approach was shown to simultaneously screen 8 samples for 169 targets.[75] Multiplexing at this scale is possible with the use of RNP complex and sample droplets uniquely stained with fluorescent dyes to encode for the contents. Droplets are colocalized on a microwell array, imaged, merged, and fluorescent intensity from the Cas catalytic activity is measured to quantify the interaction between samples and RNPs.

While multiplexing with spectrally differentiable reporters does allow for multiplexing of Cas reactions, those approaches are fundamentally limited by the spectral overlap of the signals and are unlikely to exceed the constraints of traditional qPCR methods. CARMEN does not encounter these limits due to the use of a single reporter and ratiometric spectral encoding of

sample and RNP complex solutions, but it does face other constraints. In particular, the fine spectral encoding of the droplets requires expensive image acquisition and fluidic hardware, the assay requires a minimum of 8 hours to complete due to the extensive imaging steps, and a PCR preamplification of samples is necessary to ensure that the sample concentration exceeds the minimum to ensure that every sample droplet has at least 1 target, if present in the sample. While many Cas assays can achieve results quicker than qPCR in samples with picomolar nucleic acid concentrations,[89] they are fundamentally limited by depth of multiplexing. The CARMEN assay does achieve a practical scale of multiplexing that sits between qPCR and sequencing approaches, but it is limited by the need for specialized and expensive equipment and an assay time that is not significantly better than sequencing methods.

1.5 Unaddressed needs in automation infrastructure and diagnostics

There have been significant advancements in the automation of biological assays in recent decades through the use of pipetting robots, specialized fluidic systems such as FPLCs for purifying proteins, and the development of microfluidic platforms; however, these approaches, as previously discussed, are not always practical to adopt in research settings despite clear benefits in error reduction, standardization, higher quality data, and higher throughput assays.

Even though commercial FPLC systems are regularly utilized in scientific research and and multiple variations are available to address different needs, these systems typically offer limited extensibility, which is readily apparent in entry level systems, and require the use of proprietary components, which constrain repairability and force laboratories to be dependent on long term product support.[90] Additionally, more expensive options allow for multiple samples to be processed sequentially, but no systems allow for parallelized processing to improve throughput.

While protein purification approaches have room for innovation, the dependence on commercial systems ultimately limits the flexibility and capability of these methods.

Hardware constraints are additionally present in the field of microfluidics. The diverse set of microfluidic implementations and requirements leaves many laboratories designing their own custom control equipment for their specific assays. While this approach can work, it is not easy for laboratories with limited engineering resources to replicate these platforms.[47] Alternative commercially available hardware can often be pieced together with various modules, but this approach is also expensive and may make laboratories interested in exploring microfluidics apprehensive regarding the start-up costs.[49] Consequentially, even though there are many research groups that can benefit from microfluidic assays that have been previously developed and characterized by others, barriers prevent them from adopting the technology.

Furthermore, while exploratory diagnostics are powerful methods for large scale screening, there are unaddressed gaps that can improve the efficiency of the processes. PhIP-seq allows for the screening of autoantibodies against a library of more than 700,000 unique peptides. While the method is effective for charactering the autoantibody signatures for autoimmune diseases, it is a laborious, multi-day protocol that is susceptible to moderate variance due to noisy background signals and differences in operator techniques.[56] Automation of the assay could allow for more consistent results by introducing standardized processes, cleaner signals with more thorough washes in less time, and a simplified implementation of new methods such as gradient elution to assess how autoantibody kinetics correlate severity of disease.

Similarly, while exploratory methods such as PhIP-seq and sequencing are useful for broadly characterizing diseases, they may not be best suited for confirmatory diagnostics when a commercial alternative is not available. These exploratory methods take many hours to days to

complete which can cause delays in identifying causal factors of an autoimmune flare or infectious disease. Delays in diagnosis can result in worsening prognoses as effective treatment courses cannot be implemented in a timely manner.[50,52] Methods such as multiplexing with Luminex xMAP microsphere have shown that they allow for quick serological screens for autoantibodies on the order of hours instead of days.[72] Translating autoimmune epitopes previously identified with PhIP-seq onto an xMAP format could be an effective bridge for enabling PhIP-seq results to translate into a confirmatory diagnostic for rare autoimmune diseases.[73]

There is also room for improvement in rapid nucleic acid diagnostics. While qPCR panels typically have low multiplexing depth and sequencing approaches have technical and long assay times, CRISPR-Cas enzyme may be able to serve as an intermediate solution with high multiplexing and quick assay times. Current CRISPR-Cas assays are either quick but limited in multiplexing power due to spectral overlap of reporters or they have multiplexing depth in the hundreds but require significant time and equipment to process.[75,87] An intermediate assay design with fast assay times and high multiplexing could address the challenges of qPCR and sequencing approaches for the identification of infectious diseases.

1.6 Overview of dissertation

This dissertation seeks to address the aforementioned gaps with a collection of designs for fluidic instrumentation and an exploration of assay implementations. An open source FPLC system was developed to expand the functionality and flexibility of current protein purification methods and enable parallelized purification of proteins for higher throughput. Similarly, a key barrier with the adoption of microfluidics lies with the lack of extensible, low cost, and easy-to-build solutions. One such system is described here. Fluidic systems such as these form the cornerstone of automating biological assays. In addition, multiple assays are explored including the automation

of washing steps in the PhIP-seq protocol with macrofluidic and microfluidic systems, the translation of autoimmune epitopes previously identified with PhIP-seq methods into a Luminex xMAP assay, and the development of a spatially multiplexed CRISPR-Cas assay to overcome the limits of spectral overlap of fluorophores and spectral decoding of droplet reactions.

The dissertation is structured as follows:

- **Chapter 2**: This chapter describes the development of a parallelized FPLC system that allows for milligram scale purification of 4 samples simultaneously. This system was successfully used to purify SARS-CoV-2 early in the COVID-19 pandemic. These proteins were subsequently distributed globally to facilitate the development of rapid diagnostics.

- **Chapter 3**: This chapter describes the development of an extensible, compact, and low cost microfluidic control platform that supports both multilayer, pressure driven microfluidic devices and devices that require syringe pumps for constant flow. The software architecture underlying the platform was developed to flexibly support a variety of hardware implementations and facilitate remote operation with a model-view-controller architecture and a Flask-SocketIO interface.

- **Chapter 4**: This chapter presents unpublished work and explores various approaches for automating PhIP-seq washing methods, translating autoimmune epitopes discovered with PhIP-seq into a confirmatory diagnostic, and alternative methods to implement a multiplexed CRISPR-Cas diagnostic without spectral constraints of existing methods.

- **Chapter 5**: This chapter provides closing remarks outlining the future direction of instrumentation accessibility in research settings and the continued development of serological and infectious disease diagnostics.

Chapter 2.
Open-source milligram-scale, four channel, automated protein purification system

2.1 Abstract

Liquid chromatography purification of multiple recombinant proteins, in parallel, could catalyze research and discovery if the processes are fast and approach the robustness of traditional, "one-protein-at-a-time" purification. Here, we report an automated, four channel chromatography platform that we have designed and validated for parallelized protein purification at milligram scales. The device can purify up to four proteins (each with its own single column), has inputs for up to eight buffers or solvents that can be directed to any of the four columns via a network of software-driven valves, and includes an automated fraction collector with ten positions for 1.5 or 5.0 mL collection tubes and four positions for 50 mL collection tubes for each column output. The control software can be accessed either via Python scripting, giving users full access to all steps of the purification process, or via a simple-to-navigate touch screen graphical user interface that does not require knowledge of the command line or any programming language. Using our instrument, we report milligram-scale, parallelized, single-column purification of a panel of mammalian cell expressed coronavirus (SARS-CoV-2, HCoV-229E, HCoV-OC43, HCoV-229E) trimeric Spike and monomeric Receptor Binding Domain (RBD) antigens, and monoclonal antibodies targeting SARS-CoV-2 Spike (S) and Influenza Hemagglutinin (HA). We include a detailed hardware build guide, and have made the controlling software open source, to allow others to build and customize their own protein purifier systems.

2.2 Introduction

The purification of recombinant proteins is essential for both basic research and drug discovery and development[91]. Applications ranging from the determination of protein structures, the screening of drug targets, and the production of biologic therapeutics all have been enabled by robust protein purification. The use of liquid chromatography to separate complex mixtures of proteins based on differing affinity and/or partition between a mobile (or aqueous) phase and a stationary phase (or resin) is often necessary for the development of reproducible purification processes at larger, preparative scales[91]. Commercial liquid chromatography systems, including the ÄKTA platforms, are mature and wide-spread and share essential features that at a minimum include the use of valves to select from several aqueous phases or buffers, the ability to drive consistent flow through a selected flow path or column, as well as the ability to recover the eluted and purified proteins.

While modern liquid chromatography instruments excel at the serial purification of a single protein at a time[92,93(p2)], the parallel purification of multiple proteins at once is less developed and non-standardized. At smaller analytical scales (micrograms), parallel purification of up to 96 proteins at once has been accomplished through the use of liquid handlers to drive aqueous flow through resin-packed pipet tips, or in batch mode using resin slurry or functionalized magnetic beads[94]. At a large preparative scale (milligrams to grams), the only commercially available system that we know of that is capable of purifying multiple proteins at once is the Protein Maker. Unfortunately, other commercial instruments that could run purifications in parallel at those scales, like the ÄKTAxpress and the CombiFlash OptiX 10 have been discontinued. Moreover, there are currently no open-source automated and extensible, parallel protein purification platforms described in the literature that can operate at the milligram scale. Given the rapid discovery of

new protein sequences, particularly antigen[95] and antibody[96,97] sequences, recombinantly producing and characterizing proteins at higher throughput and at multiple scales in parallel will be of critical importance to researchers moving forward.

Here we describe the development and characterization of an open-source, milligram-scale, four-channel, automated protein purification system. Each channel of the instrument can 1) select from up to 8 buffer choices via a shared rotary valve; 2) select from multiple flow path options via a series of independently controlled solenoid valves; 3) stably drive isocratic flow using peristaltic pumps, and 4) collect flow through and eluate samples for downstream analyses, using an in-line automated fraction collector. The instrument is controlled by an onboard Raspberry Pi single-board computer connected to various peripherals via an I2C bus[98]. The feature set of the system can be extended by adding new devices to the bus. To accommodate users with various levels of technical experience, the system has two interfaces: a touchscreen for the execution of standard purification protocols and a Python API for advanced programmatic control. We validated our instrument by purifying a panel of coronavirus Spike antigens (SARS-CoV-2, HCoV-229E, HCoV-OC43, HCoV-229E) and monoclonal antibodies, expressed in mammalian cells, under several parallelized processes, and establish that these purified proteins are biochemically indistinguishable from those purified using traditional, serial instrumentation. This work is open-source and includes a complete build guide, allowing customization to enable a variety of additional parallelized purification processes.

2.3 Materials and Methods

We include as supplementary information a detailed build guide and a bill of materials to allow replication of the purifier. The following links contain additional information about the construction of the device:

- Detailed CAD model of the whole instrument (available under the CERN Open Hardware License Version 2 - Weakly Reciprocal):

  https://cad.onshape.com/documents/768143c17dda5be636f2c7b2/v/b03159dbe387f6c073f53195/e/caf0bd7f2f19e0ffc6ed1810

- Software (available under the 3-clause BSD open-source license):

  https://github.com/czbiohub-sf/automated-protein-purifier

2.3.1 Hardware Design

As shown in the fluidic diagram in Figure 2-1A, each of the four affinity columns has a dedicated load input and access to shared buffer inputs. While specific buffers are selected by a motorized, 8-position rotary valve, all other fluidic path selections are done via solenoid valves. These solenoid valves allow for selecting between load and buffer input reagents, and they also route the flow to waste when washing a column or when flushing the fluidic system to load new reagents. To minimize the pressure drop across the resin, which can result in the formation of bubbles from buffer outgassing, a backpressure regulator has been placed after each column. Flow from both load and buffer inputs is driven by peristaltic pumps. The rotary valve, peristaltic pump, motorized fraction collector, and all solenoid valves are controlled by a Raspberry Pi computer. All wetted surfaces were selected to be chemically and biologically inert to facilitate sterilization and ensure compatibility with standard purification reagents. As detailed in Figures 2-1, all major components are mounted to the front panel of the machine, except for the fraction collector and the touchscreen. The front panel, fraction collector, and touchscreen are mounted on a chassis made from aluminum extrusions.

2.3.1.1 Chassis

The machine chassis is constructed with standard 20 mm × 20 mm and 20 mm × 40 mm T-slot aluminum extrusions. The chassis is 37 cm deep, 65 cm tall, and 57 cm wide, and serves to

support a front panel where most of the fluidic and electronic components are mounted, a top shelf to hold the sample and buffer bottles, and the fraction collector at the bottom. The top shelf is fitted with a polycarbonate tray that provides spill containment for the buffer bottles.

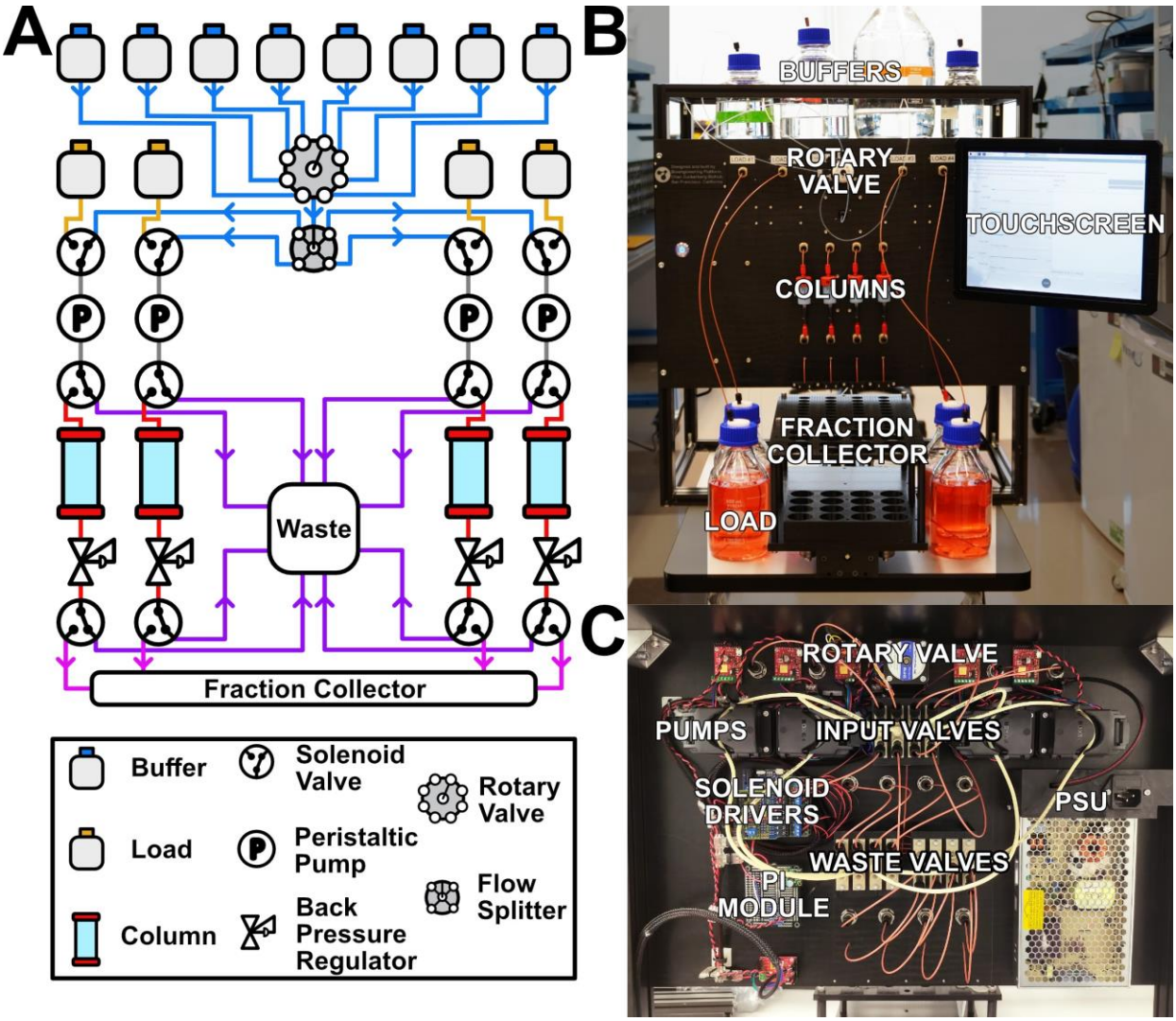


Figure 2-1. Protein purifier instrument.
A) Flow path diagram. Flow paths are colored by buffers (blue), load (orange), pre- and post-pump valves (gray), pre-column waste (purple), though column (red), post-column waste (purple), and fraction collector (pink). B) Front view of the front panel. The four-column configuration shown includes four 5 mL HisTrap™ Excel columns, four 500 mL loads, and a fraction collector capable of holding 50 mL, 5 mL, and 1mL centrifuge tubes. Buffer bottles have been partially cropped for clarity. C) Back view of the front panel, showing all the major components.

2.3.1.2 Component Panel

All fluidic components, except for back pressure valves, and all electronic components, except for the touchscreen monitor, are mounted directly to a front panel made of a 5mm thick laser-cut acetal sheet. Acetal was selected for the panel primarily for its strength, good chemical resistance to most reagents commonly used in laboratories (especially to ethanol, isopropanol and bleach, commonly used to clean and disinfect) and for its compatibility with laser cutting. Acrylic is not a good material for the panel because it is brittle and degrades when exposed to ethanol and isopropanol.

2.3.2 Electronic Design

The electronics for the protein purification system (Figure 2-2) allow for programmatic control of the flow paths and flow rates used by each fluidic channel. Please note that all voltages mentioned below are DC, unless explicitly marked as AC.

2.3.2.1 Single Board Computer

A Raspberry Pi 3B+ was used as a single board computer to run all of the automation in the machine. It was selected due to its low cost, small size, widespread availability of accessories, and sufficient features for this project, including plenty of general-purpose input/output (GPIO) pins for hardware control, integrated WiFi, HDMI, USB and Ethernet ports. All peripherals are controlled by the Raspberry Pi via an I2C bus. However, the I2C bus on the Raspberry Pi operates at 3.3 V, while the I2C bus on the peripherals operates at 5 V. To interface with the peripherals and receive power from the 24 V power supply, a custom circuit board was built and mounted on the general-purpose input/output (GPIO) pins of the Pi (**Appendix A**). This circuit includes bidirectional logic level shifters for operating the I2C bus at different voltages and a step-down

voltage regulator to power the Raspberry Pi and the solenoid valve controllers with 5 V. A detailed schematic of this circuit can be found in the associated build guide. The system is running on Debian 10 installed on a 16 GB SD card.

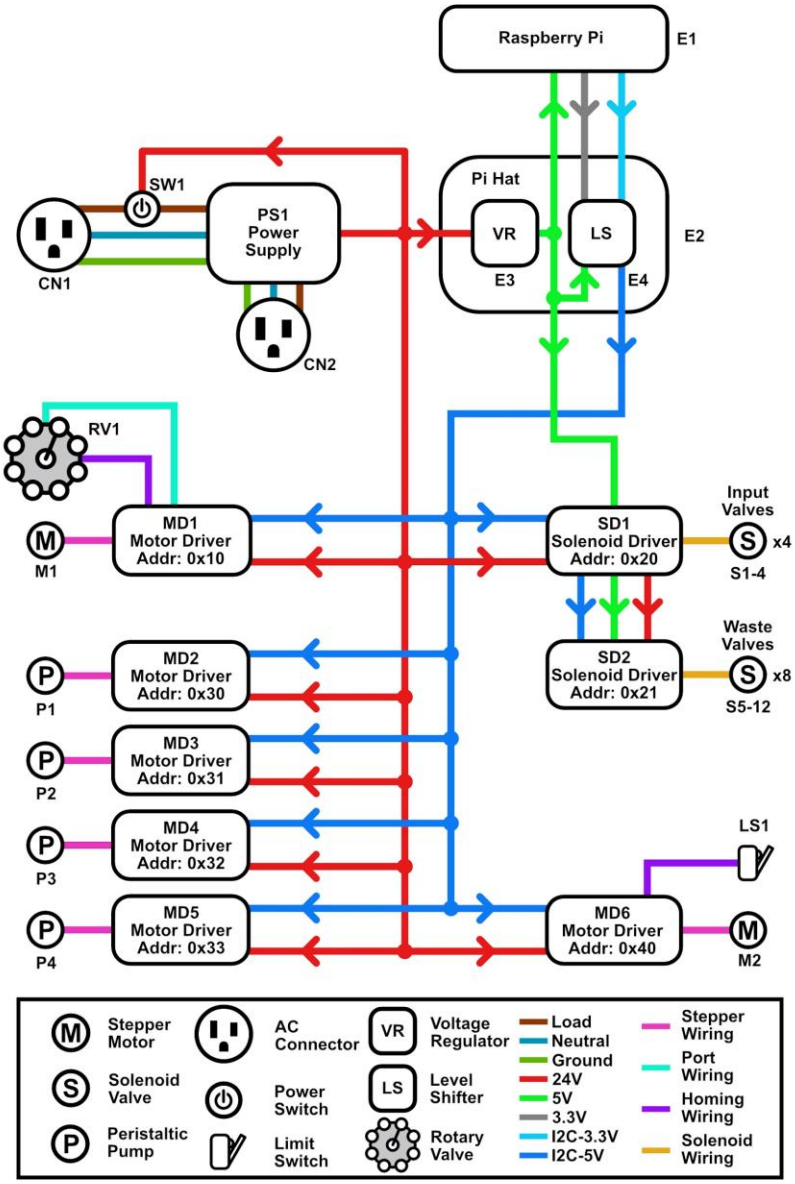


Figure 2-2. High level electronics diagram.
The protein purifier software is executed on a Raspberry Pi, which interfaces with all hardware peripherals through an I2C bus. While most peripherals are powered by a 24 V power supply, a secondary 5 V source generated by a voltage regulator on the Pi hat is used to power both the Raspberry Pi and the solenoid valve controller. The custom hat also uses a level shifter to convert the 3.3 V I2C signal of the Raspberry Pi into a 5 V signal for the attached peripherals.

2.3.2.2 Power Supply

The power supply takes 120 VAC or 240 VAC (selectable by a switch) as input and outputs 24 V, which is routed to all electronic peripherals (peristaltic pumps, rotary valve, fraction collector, solenoid valve controllers), and to the 5 V step-down voltage regulator on the Raspberry Pi hat. An illuminated main power switch is used to pass the 120/240 VAC input to the power supply and to a power socket for the touch screen. For safety, there is a 2.5 A fuse on the 120/240 VAC input. The 5 V step-down voltage regulator on the custom circuit board mounted to the Raspberry Pi converts the 24 V supply to 5 V, which is then passed to the power pins of the Raspberry Pi, the peripheral side of the bidirectional level shifter, and the IOREF input of the solenoid valve drivers. The 3.3 V output of the Raspberry Pi is passed to the Pi-side of the bidirectional level shifter.

2.3.2.3 I2C Bus

All electronic peripherals included in this design are capable of communicating over the I2C bus in the Raspberry Pi, which allows up to 128 devices to sit on the same bus as long as they are each given a unique address. This approach both simplifies the required wiring and the process of replacing or adding modules. While the Raspberry Pi 3B+ technically has hardware I2C capabilities, there is an error in how the silicon was designed and it is unstable when used with devices that support clock stretching[99]. There is evidence that suggests this error is still present on newer revisions, including the Raspberry Pi 4[100]. To work around this issue, the I2C communication protocol was implemented via software with a GPIO overlay. Two arbitrary GPIO pins were assigned to serve as the I2C data and clock lines and were routed to the Raspberry Pi-side bidirectional level shifter.

2.3.2.4 Solenoid Valves

Each fluidic channel requires three 2/3 way solenoid valves for configuring the flow path. The first valve is used for selecting between buffer and sample, the second is used for purging the pre-column path to waste, and the third is used for directing buffers passing through the column to either the fraction collector or to waste. The valve material was selected to be PEEK in order to be compatible with a broad range of chemicals, buffers, and biological materials that may be run through the system. The solenoid valves are controlled by stackable 8-channel relay driver boards that are connected to the Raspberry Pi via the I2C bus.

2.3.2.5 Peristaltic Pumps

Each channel has a 6-roller peristaltic pump that is placed between the solenoid valve that controls the sample input and the solenoid valve that purges the pre-column path to waste. A peristaltic pump is ideal in this system because it offers smooth flow with minimal pulsing, a well-defined and controllable flow rate, and a closed fluidic path that prevents buffers from passively leaking by gravity through the system when not in use. The pump comes with an integrated stepper motor that is connected to a motor driver module that is in turn connected to the Raspberry Pi via the I2C bus. When used with the recommended 1mm inner diameter tubing, the pump can achieve a minimum flow rate of 4×10^{-6} mL/min and a maximum flow rate of 21 mL/min. The flow rate range can be adjusted by using tubing with a different inner diameter.

2.3.2.6 Rotary Valve

The 8-to-1 rotary valve is used to select one of eight possible input buffers to be routed to all channels on the system through a 1-to-4 splitter. The valve material is PTFE, which has high chemical compatibility and low friction. The valve has integrated home and port position sensors

and is driven by a NEMA 18 stepper motor. The sensors and the stepper motor are connected to a motor driver module that is connected to the Raspberry Pi via the I2C bus.

2.3.2.7 Fraction Collector

The output of the column can be redirected to either waste or an automated fraction collector that is composed of a carriage with a holder for fraction containers mounted onto a linear actuator. Given that different sized fractions are desirable for columns of different volumes, the carriage was designed to have an interchangeable fraction holder (top plate) that can be designed to hold many different arrangements of containers. We have designed two holders, with one version holding ten rows of 1.5 mL Eppendorf tubes and the other ten rows of 5 mL Eppendorf tubes. Both also have four rows of 50 mL of Falcon tubes to collect flow through. The linear actuator is attached to the machine chassis and powered by a NEMA 18 stepper motor that is connected to a motor driver module, which is in turn connected to the Raspberry Pi via the I2C bus.

2.3.2.8 Touchscreen Display

The touchscreen display provides a basic interface for generating and running purification protocols on the machine. The display is connected to the HDMI port on the Raspberry Pi and the touch capabilities are managed via USB. While the touchscreen interface does allow for basic modifications of protocol parameters, more advanced features are only accessible through the Python programmatic interface.

2.3.3 Software Design

2.3.3.1 Structure

The software developed here was written in Python 3 and organized as a client-server model (Figure 2-3) that abstracts low-level peripheral operation of the machine from the operator and allows for remote control. The server-side software includes various hardware drivers and middleware that manages the peripherals and communication with the client. The client-side software includes middleware for managing communication with the server and high-level instructions that can be translated into operations by the server. While the server-side software is always executed as a service on the Raspberry Pi in the instrument, the client-side software can be run on an ad-hoc basis either remotely or locally through a Python interpreter or a script executed from the command line. We have also designed a graphical user interface (GUI) that makes it very easy to run the system with predefined protocols without any knowledge of Python programming. However, the current GUI design can only be run locally, using the touchscreen of the system. As implemented, only a single client can interface with the server until control is relinquished by the client or a disconnect is forced. This means that when the GUI is running remote clients cannot connect to the machine.

If errors occur, system logs can be recorded at different levels of verbosity on both the client and server to debug potential hardware or communication faults. This feature is particularly useful when modifying the system to include new features or different hardware.

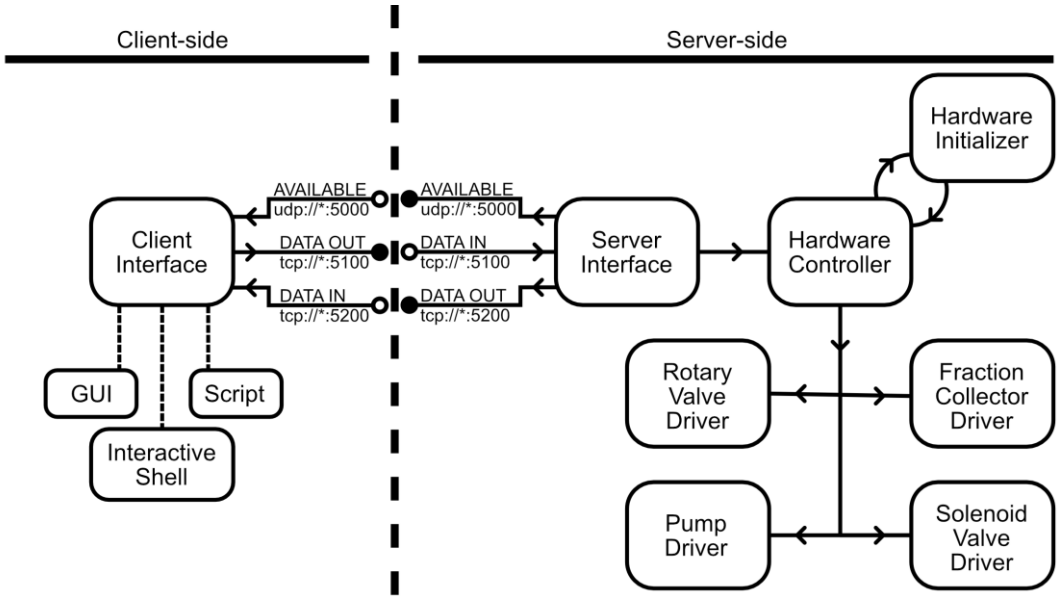


Figure 2-3. System software diagram.

The software used by the system follows a client-server model. While the client interface can be generated by a GUI, interactive shell, or a script, the TCP data connections only allow for one client to interface with the system at a time. When there isn't an active connection, the server will periodically broadcast its status on the UDP port. Once a client has connected, all data will flow to the server's device interface, which is responsible for parsing commands to be executed and providing system updates to the client. After connecting, the client will select a hardware configuration to be used by the system, which is then forwarded to the hardware controller and subsequently processed by the hardware initializer. Afterwards, additional commands can be routed to the peripherals through the hardware controller.

2.3.3.2 Server Software

Each type of peripheral in the instrument is managed through a driver class that defines how low-level commands are issued from the Raspberry Pi to the hardware device. All these driver classes are controlled by a hardware controller class, which facilitates communication with peripherals by defining a subset of operationally significant commands. This driver-controller implementation allows the higher-level functionality of the software to be preserved if alternative peripherals are used. To accommodate alternative hardware peripherals for the existing functionality such as a different stepper motor controller board, one could write a new driver to control the peripheral without modifying the higher-level code. Extending the feature set of the system, however, would require adding additional functionality to the higher-level code.

Given that each machine may have a different set of peripherals or configurations that can be changed between use, such as employing 1mL or 5mL resin columns, the connected client states what configuration should be used by the hardware controller prior to operation. Once the configuration is identified, the controller instantiates the hardware classes accordingly and preserves the configuration until the client disconnects or it is manually reset.

To allow for local and remote clients, there is a server communication interface that is perpetually run as a Linux service and performs three functions: 1) it broadcasts server availability to potential clients when not in use, 2) it receives commands from the client that are processed into instructions for the hardware controller, and 3) it reports to the client data resulting from both receiving and completing the command provided. This communication occurs over separate TCP sockets on the Raspberry Pi, which allows for local client operation on the machine and remote operation over a network.

2.3.3.3 Client Software

The client software includes middleware for managing communication with the server and high-level command wrappers for facilitating human-readable scripts. The communication middleware consists of a set of commands for connecting to and disconnecting from the server, validating server responses, and issuing peripheral commands. While the operator will need to provide the TCP address of the server and identify the name of the hardware configuration to be used, most other operations are fully defined by the command wrappers, which simplifies the scripting process. A GUI is also available for operators who are not comfortable with programming in Python and are not interested in advanced features. The GUI uses a few parameters such as the name of the configuration to be used and the volume of sample to be loaded into the columns in

order to automatically generate a script that is run locally on the system. The GUI also allows operators to pause, hold on a step, or terminate the run using buttons.

2.3.4 Characterization

2.3.4.1 Evaluation of long-term flow rate stability

Phosphate Buffered Saline (PBS) was pumped through the buffer and load flow paths through one channel for a duration of 120 minutes and measured by weighing the accumulated output volume on a Mettler Toledo digital balance. The flow was paused every 10 minutes to allow the scale to acquire a stable reading, which was then transmitted to a PC using RSKey software (A&D Company) and an RS-232 cable. This process was repeated with both 1 mL and 5 mL IMAC columns at flow rates of one column volume per minute. Due to the 320 g mass limit of the scale, the accumulated volume in the 5 mL runs was disposed of every 40 minutes to stay within the limits of the scale.

2.3.4.2 Evaluation of fraction volume consistency

The flow rates of all buffer flow paths were coarsely calibrated in parallel by flowing PBS through a 1 mL or 5 mL HisTrap excel (Cytiva)Immobilized Metal Affinity Chromatography (IMAC)column on each channel at an approximate flow rate of one column volume per minute. Following calibration, measurements were then taken by dispensing PBS from either buffer or load flow paths into the fraction collector through 4 channels simultaneously for one minute. This process was repeated with both 1 mL and 5 mL IMAC columns at flow rates of one column volume per minute and 30 measurements were taken for each condition.

2.3.4.3 Generation of stable Expi293 cells for the expression of soluble SARS-CoV-2 Spike (S) and Receptor Binding Domain (RBD)

cDNAs for soluble SARS-CoV-2 S and RBD (containing C-terminal His-tag)[101] were cloned into EcoRV-cut plenti-CMV-Puro-DEST (Addgene, #17452, gift from Eric Campeau & Paul Kaufman) using NEBuilder HiFi DNA Assembly Master Mix (NEB). Lentivirus was produced in HEK293FT by co-transfection of cDNA containing pLenti plasmids together with pCMV-dR8.2 dvpr (Addgene, #8455, gift from Bob Weinberg), pCMV-VSV-G (Addgene, #8454, gift from Bob Weinberg) and pAdVAntage (Promega) using FugeneHD (Promega). Supernatants were collected 48 hr post-transfection and filtered using a 0.45 μm syringe filter. For the transduction of Expi293 cells, five 6-well plates with 2 million cells per well were spin-infected with lentivirus diluted 1:2.5 in fresh Expi293 expression media under the following conditions: 2 hr, 33°C, 1000 g. Cells were subsequently pooled together and cultured in 30 mL Expi293 expression media on a shaking incubator. 24 h post-transduction puromycin was added at 2 μg/mL and cells were selected for 7 days.

2.3.4.4 Expression of coronavirus S and RBD, and monoclonal antibodies

For stable expression of SARS-CoV-2 S and RBD, stable suspension cells were maintained in Expi293 media (Thermo Fisher Scientific) at 37°C, 140 RPM (25 mm shaking diameter), 8% CO_2, and 80% humidity. All suspension cells were cultured in Thomson Optimum Growth Flasks. Flasks were seeded at a density of ~0.3×10^6 live cells/mL at either 0.8 L scale (1.6 L flask) or 1.2 L scale (2.8 L flask). Cells were then grown for 6 days and harvested at a density of 8×10^6 to 10×10^6 live cells/mL at >90% viability. Harvested cell cultures were centrifuged at 4,000 g for 30 min, followed by filtration of the supernatant through a 0.45 μm NalGene Rapid Flow filter unit (Thermo Fisher Scientific). Filtered supernatants were adjusted to pH 7.4 using 1 M NaOH.

Expression titers (purified protein per L of culture) for SARS-CoV-2 S and RBD from stable cell lines were ~3 mg/L and ~20 mg/L, respectively.

For transient expression of coronavirus S, RBD, and monoclonal antibodies expression plasmids were transiently transfected into suspension Expi293 cells following protocols as previously described[102]. For antibodies, the amount of plasmid to be transfected was split equally between the corresponding pair of heavy and light chain plasmids. Three days after transfection, cell cultures at a density of 3 to 5×10^6 live cells/mL and ~50-75% viability were harvested, clarified, and adjusted to pH 7.4 as described above. For monoclonal antibodies, plasmids separately encoding for the heavy chain and light chains were transfected at 200 mL scale (500 mL flask), grown for 5-6 days (viability = ~98%), and harvested, clarified and adjusted to pH 7.4 as described above. Serum (rabbit) was purchased from a commercial vendor (Sigma, #R4505) and filtered through a 0.45 μm syringe filter prior to purification. Expression titers (purified protein per L of culture) for SARS-CoV-2 S and RBD from transiently transfected cells were similar to that of stable cell lines, ~1-3 mg/L and ~20 mg/L, respectively. Nevertheless, the low viability of cells transiently expressing SARS-CoV-2 S resulted in clogging of the filtration units and more time needed to clarify these supernatants.

2.3.4.5 Parallelized affinity chromatography purification of S, RBD, and antibodies

For S and RBD, four HisTrap Excel columns (column volume = 1 mL or 5 mL) were equilibrated in 10 column volumes (CV) of 20 mM sodium phosphate, 500 mM NaCl pH 7.4. Following parallelized loading (4 × 50 CV for CV = 1 mL; 4 × 200 CV for CV = 5 mL), each column was washed with 60 CV of (20 mM sodium phosphate, 500 mM NaCl, 20 mM imidazole, pH 7.4) and eluted with (20 mM sodium phosphate, 500 mM NaCl, 500 mM imidazole, pH 7.4) into eight 1 CV fractions. For antibodies, four 1 mL HisTrap Protein A HP (Cytiva) were

equilibrated in 10 CV of 20 mM sodium phosphate pH 7.0, loaded in parallel (4 × 10 CV), washed in 30 CV of 20 mM sodium phosphate pH 7.0, and each column was eluted with 0.1 M citric acid pH 3.0 into five 0.85 CV fractions, each fraction containing 0.15 CV of 1 M Tris pH 9.0. All steps were performed at a flow rate calibrated to 1 CV/min at the beginning of the run. Purified protein fractions (8 CV for HisTrap Excel, 5 CV for HisTran Protein A HP) were then buffer exchanged into PBS using either Amicon Ultra-15 centrifugal filter units (Millipore Sigma, 100 kDa cutoff for S, 30 kDa for antibodies, 10 kDa for RBD) or 10DG desalting columns (BioRad). S and RBD proteins, optionally supplemented with 10% glycerol, were filtered and stored at -80°C.

2.3.4.6 Size exclusion multi-angle scattering (SEC-MALS) analysis of purified S proteins

Purified S proteins were analyzed on a SRT SEC-1000 column (5 μm, 4.6x300 mm, Sepax) at 0.35 mL/min in PBS running buffer. The SRT SEC-1000 column and SEC-MALS system, which includes a 1260 Infinity II HPLC (Agilent), a miniDAWN TREOS II MALS detector (Wyatt) and a Optilab T-rEX refractive index detector (Wyatt), were equilibrated in PBS at 0.1 mL/min for at least 24 hr prior to analysis. Molecular weights were determined using the Astra software package.

2.4 Results

2.4.1 Flow rates of a single channel are stable over time

A liquid chromatography instrument should be able to deliver stable flow through all user-selected flow paths and columns for the duration of a purification protocol. To measure flow rate stability, we pumped PBS over 1 mL and 5 mL IMAC columns using two different flow paths (load and buffer) (Figure 2-4). Flow path specific differences in flow rate were small and flow rates were stable for all flow paths and column volumes tested, drifting less than 2 % over 120

minutes (Figure 2-4). Overall, we show that our instrument can deliver stable flow rates over time across different column sizes and using different flow paths.

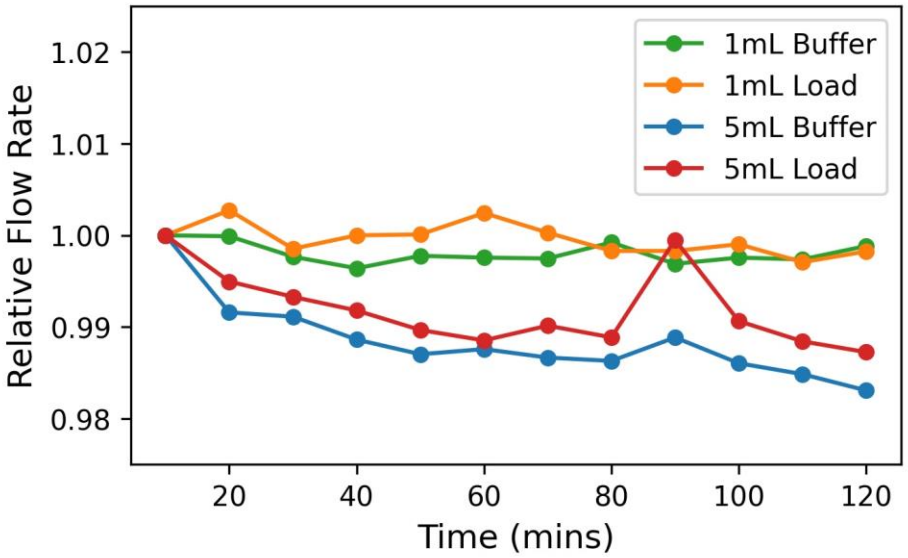


Figure 2-4. Single-column flow rate stability.
Relative flow rates over time through 1 mL and 5mL HisTrap™ Excel columns are shown. PBS was pumped through each column using the buffer and load flow paths for a duration of 120 minutes and measured every 10 minutes.

2.4.2 Fraction volumes collected in parallel are consistent and accurate

To determine if our instrument can consistently dispense fractions of equivalent volumes across all channels, we pumped PBS using the buffer and load flow paths, through 1 mL and 5 mL columns, and measured the fraction volumes collected by the fraction collector (Figure 2-5). The mean fraction volumes for the 1 mL columns showed little difference and ranged from 0.99 +/- 0.01 column volume (CV) to 1.02 +/- 0.02 CV for the buffer flow path and from 1.00 +/- 0.01 CV to 1.01 +/- 0.01 CV for the load flow path. For the 5 mL columns the mean fraction volume ranged from 0.96 +/- 0.01 CV to 01.00 +/- 0.01 CV for the buffer flow path, and from 0.92 +/- 0.01 CV to 0.98 +/- 0.00 CV for the load flow path. Comparing the flow rates between the 1 mL and 5 mL columns, there is a decrease in accuracy, but not precision for the fraction volumes when using the

larger 5 mL columns. Both column types are run at a similar linear flow velocity (~150 cm/h), thus the decrease in accuracy is likely not due to differences in linear flow velocity or column back pressure. This effect may be an artifact related to the elasticity of the peristaltic pump tubing being compressed at a higher frequency and the difference in hydrostatic pressure between the load and buffer bottles, which are placed at two different heights. Nevertheless, fractions that would be eluted off the 5 mL column, using the buffer flow path, exhibit a small error of up to <4%. Taken together, we have established that our instrument can deliver consistent and accurate fraction volumes that are necessary for parallelized protein purification.

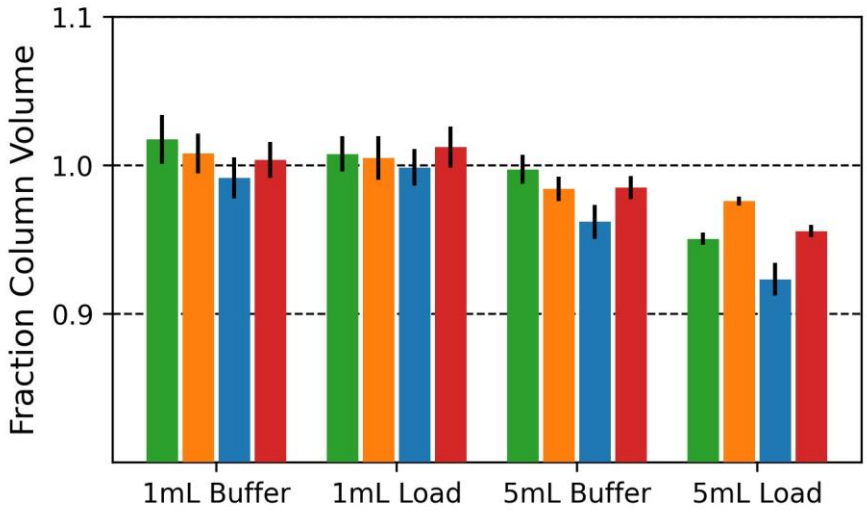


Figure 2-5. Parallelized elution profiles.
Fractions in units of column volumes collected over four 1 mL and 5mL HisTrap™ Excel columns in parallel are shown. PBS was pumped through the columns using both buffer and load flow paths, as indicated in the x-axis labels. Each color corresponds to one channel in each of the conditions.

2.4.3 Negligible inter-channel variability in purity and yield of parallel purified SARS-CoV-2 Spike RBD

To assess channel specific differences in protein purity and yield, we performed 4-channel IMAC purification of SARS-CoV-2 Spike RBD from a single mammalian cell expression source. The cell culture harvest was minimally processed prior to loading (see Characterization),

and we observed stable flow over the duration of the purification process. Capture of RBD to the IMAC column is near complete for all channels (Figure 2-6A), and the elution profiles for all channels as measured by reducing SDS-PAGE are very similar (Figure 2-6A). Quantification of the eluted protein, by Bradford analysis, shows very good inter-channel agreement in yield for both the 1 mL (0.37 +/- 0.02 mg and 0.44 +/- 0.06 mg for fractions 2 and 3, respectively) (Figure 2-4A) and 5 mL (0.64 +/- 0.06 mg and 0.12 +/- 0.01 mg for fractions 2 and 3, respectively) IMAC processes. We therefore show that our instrument can perform near-identical purification processes in parallel, with no appreciable channel-specific differences in purity or yield observed.

2.4.4 Purification of diverse serum polyclonal and recombinant monoclonal antibodies in parallel

To determine if different samples can be effectively purified in parallel using our instrument, we took three recombinantly expressed monoclonal antibodies (CR3022, 65C6 and H5M9[103–105]) and one rabbit serum sample, and purified each using a 1 mL MabSelect Sure column (Figure 2-6B). The clarified cell culture harvests and serum exhibited a large range of starting protein titers (highest for rabbit serum, lowest for H5M9 mAb), and no large deviation from the targeted flow rate of 1 mL/min was observed during the purification process, suggesting that there was no adversely large increase in column backpressure. For each sample, following identical loading and wash volumes, a 1-2 CV elution volume containing >95% of total eluted protein at high purity was observed, with each fraction having similar relative amounts of heavy and light chain as estimated by reducing SDS-PAGE (Figure 2-6B). Nevertheless, the elution profile of each sample is distinct; the recombinant human mAbs have a maximum at fraction 3, while the rabbit serum polyclonal Abs are more readily eluted from the column having a maximum at fraction 2. As our clarified harvests and serum were minimally processed prior to purification,

these distinct elution profiles may be due to intrinsic differences in affinity between rabbit and human IgGs for MabSelect Sure (a Protein A derived matrix), and/or due to non-standardized initial loading conditions. In conclusion, we show that our instrument can parallelize the purification of samples under a wide range of titers, and that different sample-specific elution profiles can be readily accommodated through fractionation.

2.4.5 Parallel purification of coronavirus Spike antigens show expected biophysical properties

Beyond yield and purity, the quality (or activity) of purified proteins from a parallel process should match with those purified using "one-protein-at-a time" processes and instrumentation. Instrument-specific factors that could negatively impact protein quality could conceivably include, but are not limited to, temperature and pressure fluctuations, non-specific interactions (to tubing), carryover of residual clean-in-place (CIP) reagents used during column regeneration and/or leakage of instrument materials into the purified eluate.

To assess if there were unforeseen instrument-specific effects on protein stability, we measured the SEC-MALS profiles of four coronavirus Spike antigens that were purified in parallel. Following IMAC purification and desalting, all Spike antigens were pure and of the correct molecular weight as shown by reducing SDS-PAGE. Light scattering (blue trace, Figure 2-6C-F) shows sample-specific differences, with each sample having a major peak (>~95% of total mass fraction by RI) that is trimeric (red curve) and two spike samples (HCoV-OC43 and HCoV-2293) showing a smaller magnitude light scattering peak, corresponding to <~5% mass fraction of the larger peak. Functional assays using our parallel purified SARS-CoV-2 Spike[102,106–112] also show that the antigen behaves similarly to other reported SARS-CoV-2 Spike protein lots, prepared using

commercial instrumentation. In conclusion, we show that proteins purified using our instrument are of the correct oligomeric state with no adverse instrument-specific differences in function.

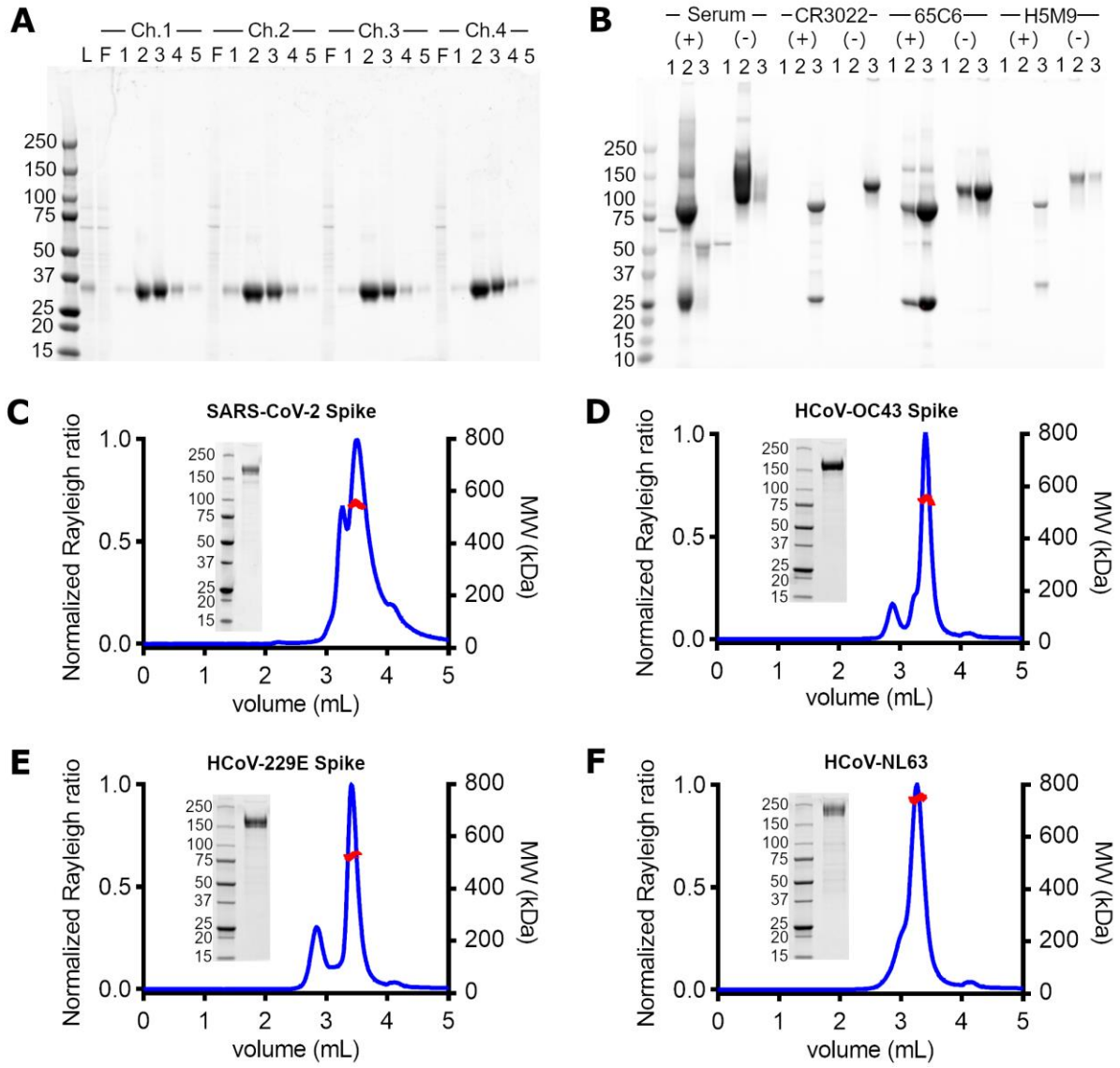


Figure 2-6. Purity and oligomer analyses of parallel purified coronavirus Spike antigens and antibodies.

A) SARS-CoV-2 Spike RBD, 1 mL HisTrap Excel purification. 10 µL of Load (L) (identical for each channel), flowthrough (F) and elution fractions (1-5) were analyzed by reducing, Coomassie-stained SDS-PAGE. B) mAbs (Anti SARS-CoV-2 Spike and anti Influenza Hemagglutinin) and pAb (healthy serum), 1 mL HiTrap Protein A purification. 10 µL of elution fractions (1-3) were analyzed by Coomassie-stained SDS-PAGE in the presence (+) and in the absence (-) of reducing agent. C-F) Coronavirus Spike antigens, 5 mL HisTrap Excel purification. Parallel purified Coronavirus spike antigens, desalted and concentrated offline, were analyzed by reducing SDS-PAGE (5 µg protein loaded) and analytical SEC-MALS. The calculated molecular weights of the major peak are shown in red.

2.5 Discussion

This work has developed an open-source, automated protein purification system that utilizes up to four-channels for milligram scale purification. While a few multi-channel liquid chromatography systems at this scale have been made commercially available, most have been discontinued and none have been made open source. Given the open-source nature of the system, others are free to replicate and customize their machine as needed to meet the changing demands of their work.

We have shown that the system is effective at purification and on par with commercial alternatives with the added benefit of parallelizable purification. We have demonstrated that different flow paths and column sizes had stable and accurate flow rates with consistent precision. Similarly, each of the parallelized channels yielded near-identical purity and yields of SARS-CoV-2 Spike RBD when a sample was split and run through each column simultaneously. The system was also shown to be capable of accommodating a diverse set of samples when various recombinantly expressed monoclonal antibodies and rabbit serum were processed in parallel. Furthermore, when four coronavirus spike proteins were purified and analyzed, there were no discernible differences in form or function between those prepared by commercial instrumentation and the system presented here.

While our platform's open-source design and implementation has several strengths for parallelized purification as outlined above, there are a couple of caveats that should be noted. First, the use of peristaltic pumps to drive flow, while cost effective, can result in a reduction in flow if the column back pressure increases during sample loading. Care must be taken to ensure that samples are filtered prior to loading to minimize the chances of clogging or aggregation that would increase column back pressure. In cases where flow rate is significantly reduced (e.g., during the

loading of a viscous sample), we have implemented a hold step to ensure that each step of the purification can be manually held to completion by the user. Second, to keep costs low, our platform does not currently include an inline A280 detector to monitor, in real time, the elution of proteins off each column. Therefore, an initial pilot purification, where purity and yield are determined offline from the eluted fractions, is useful to define suitable parameters that can be used for the subsequent purification of multiple proteins in parallel.

Even though these cost-saving omissions did not impede our purification of coronavirus Spike proteins and antibodies, other researchers may require additional modules or functionality for their purification workflows. The instrument is an open platform that permits researchers to make these customizations through its modular and extensible software architecture and accessible and reconfigurable hardware implementation. For UV detection, this would involve inserting a sensor into the flow path, writing a software class to interface with the hardware, and applying the class methods within a script. For preventing back pressure buildup, a flow rate sensor would be inserted into the flow path and a new pump class with a closed-loop feedback algorithm would replace the currently implemented open-loop class. Other desired extensions would follow similar implementation methods.

The goal of this system was to improve purification throughput via parallelization and automation while providing accessibility to a broad group of scientists. While a touchscreen interface is present to assist non-technical users with basic experiments, a Python programming interface is also made available for complex or custom operations. Similarly, different scientists may require different features, so the system was designed to be modifiable to support a broad range of use cases. To date, this system has played an important role in many of our studies focused on SARS-CoV-2 and is being continually augmented to support new capabilities such as serial

column purification and gradient elutions in addition to exploring new methods such as automated immunopurification and phage display technologies.

As mentioned in the Materials and Methods section, to allow replication of the device, a bill of materials and a build guide are available as supplementary information (S1, S2). Links to the CAD model and the software repository are also provided in that section. The total cost of materials to build the device is approximately US$14,000. Assembly takes approximately 30 hours and requires a soldering iron (for electronics), a 3D printer, a laser cutter, a hand-held drill, and some other basic tools. Laboratories that do not have access to a 3D printer or laser cutter can make use of myriad companies that today offer 3D printing and laser cutting services at very affordable prices. Given the lack of equivalent commercial instruments, we envision that this work might be of great interest to a large community of researchers that routinely purify proteins in milligram scales, and to the open science community in general.

2.6 Supplementary Figure

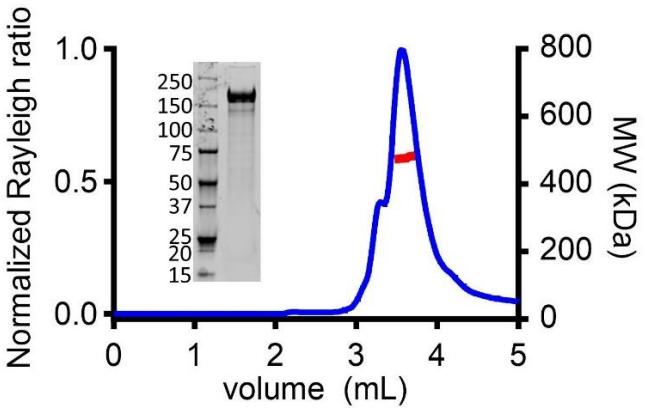


Figure 2-7: Purification of SARS-CoV-2 Spike using a commercial instrument.
SARS-CoV-2 Spike was purified using a 5 mL HisTrap Excel column on a commercial chromatography instrument (ÄKTA Pure), desalted and concentrated offline, and analyzed by reducing SDS-PAGE (2.5 µg protein loaded) and analytical SEC-MALS. The calculated molecular weight of the major peak is shown in red.

Chapter 3.
An extensible, comprehensive microfluidic control platform

3.1 Abstract

Several decades of literature detailing the development of microfluidic devices have demonstrated the benefits of the technology in enabling automated, miniaturized assays and accessing methods and processes that are not feasible on the macrofluidic scale. Despite the volume of work that has been described to date, the technology still faces hurdles in adoption, particularly in regard to control instrumentation. While academic and commercial options exist, these systems are often limited by financial, technical, or extensibility constraints and are difficult to implement in settings without hardware or software engineering support. The work herein describes a compact, low cost control platform that is designed for reproducibility, extensibility, and ease-of-use. The system features 24 solenoid valves pressurized by 2 independent regulators, 8 pressurized reagent ports, a universally deployable Python 3 application named plfluidic, software support for syringe pumps, a comprehensive browser-accessible user interface with an integrated scripting language for automation, an optional and recommended compilable operating system image, and a custom USB solenoid driver. The overhead of the user interface introduces a delay that is equivalent to valve actuation time, but the user interface may be bypassed for direct control of the solenoid driver with the included Python driver for sub-millisecond execution times. In addition multipressure and multimodal fluidic support, the system can be integrated into existing experimental infrastructure with API of the HTTP and WebSocket interface if not directly controlled by USB. The control platform addresses many of the limitations of existing systems and is expected to improve the accessibility of microfluidics in both research and teaching laboratories.

3.2 Introduction

The miniaturization of biochemical assays onto microfluidic platforms yields significant advantages including reduced reagent consumption, enhanced reaction kinetics within confined volumes, high-throughput automation, and access to sample manipulation methods that are not feasible on the macroscale.[26,113,114] As a result, a continually increasing volume of work has been published over the past few decades to describe novel microfluidic assays that leverage these phenomena for addressing biological and chemical challenges.[115] Despite the development of numerous methods, microfluidic technologies largely remain confined to engineering laboratories that possess the technical skills required to overcome bottlenecks in device fabrication and operation.[116,117]

Recent advancements in fabrication methods, such as rapid prototyping with 3D printing and laser etching, have improved the accessibility of device fabrication processes. Benchtop device fabrication is currently feasible and largely mitigates the capital-intensive challenges of traditional, cleanroom-dependent photolithography methods.[115,118–120] However, the challenges of device operation remain largely unaddressed and the practicality of existing control instrumentation remains to be a hinderance for broader adoption. Key barriers to entry include cost, ease of assembly, ease of use, and ability to flexibly adapt to new applications.

Commercial vendors of instrumentation for microfluidic research, such as Fluigent and Elveflow, offer well-engineered modules that can be pieced together to form a control system. While they offer ease of assembly and can be extended to new applications with additional modules, the costs easily fare into the tens of thousands of USD per setup and are dependent on end users programming their own interfaces to control the hardware. These barriers to entry may

be too high for laboratories interested in assessing the feasibility of applying the technology in their studies.

Academic control instrumentation is also well described in the literature but has limitations of its own. Brower *et al.* introduced a pneumatic control platform for multilayer devices which supported 48 control lines over 3 regulated pressure circuits, 18 pressurized reagent lines and remote-control capability.[49] While capable of supporting a broad range of multilayer devices leveraging elastomeric microvalves, the primary limitations of this system include a large physical footprint (0.74 m^2, 8 ft^2), a cost of more than $10,000 USD in 2017, a variety of custom 3D printed and machined parts, a remote interface that required a disruptible, always-on connection to execute experiments, and the lack of extensibility to support both constant flow and constant pressure modes of device operation.

Other control instrumentation do have their own merits but are limited in that they are either developed for a narrowly defined application[121–126], limited in extensibility[127–130], or serve as a prototype and require further engineering to implement[131–136] if not already rendered obsolete[137]. Many of these systems also require end users to program their own control interfaces, are loosely assembled with electrical breadboards, are either purely constant pressure or constant flow systems, and do not provide logging utilities for experimental reproducibility.

To address the limitations of existing commercial and custom systems, this work introduces a microfluidic control platform (MCP) designed for multi-pressure and multimodal microfluidic automation. Constructed for approximately $2,500 USD with a physical footprint of less than 0.1 m^2 (1 ft^2), the instrument offers a compact and cost-effective alternative without compromising operational capabilities (Figure 3-1). The hardware design features 24 solenoid valves managed by two independent pressure regulators and includes 8 pressurized reagent ports, which makes this

system suitable for complex fluidic configurations such as actuating elastomeric valves in multi-height flow channels. Furthermore, the system architecture is designed to be extensible and includes support external flow control modalities including commercial syringe pumps. Hardware drivers for a custom solenoid driver printed circuit board (PCB), generic FT245R-based relay driver boards, and New Era syringe pumps are provided to provide end users flexibility in configuration. Build instructions, an operational guide, and bill-of-materials can be found in the Appendices along with PCB design files and source code as listed in the "Data availability" section.

The foundation of this platform is a custom Python module, *plfluidic*, that enables an extensive set of features and flexibility. The *plfluidic* module implements a multithreaded, model-view-controller (MVC) framework that communicates with the user via standard HTTP and WebSocket protocols. This design decision allows for a browser based user interface (UI) for manual or scripted device operation while simultaneously serving as an application programming interface (API) for seamless integration with external laboratory infrastructure, such as automated microscopy scripts. While not required, new microfluidic devices would benefit from creating a new hardware configuration file and modifying the HTML UI to assign labels to individual valves or pumps. Several examples are provided to facilitate use and adoption of the MCP.

While controlling the hardware with a remote UI is a convenient quality-of-life feature for operators, remote connections are also subject to timing inconsistencies and network disconnections that are not amenable for managing a controlled microfluidic experiment. The system here addresses the reliability challenge by transmitting manual operations over the network while also executing user-defined scripts with a threaded finite state machine on the local controller hardware. Experiments that require sub-millisecond control of hardware can bypass the overhead of the UI and control the system directly via USB and the provided Python drivers. While any

computer can control the platform, a custom, easy-to-compile Raspberry Pi Linux image has also been developed to facilitate system reproducibility and mitigate the challenges of software dependencies. Finally, the system also includes extensive logging to facilitate device troubleshooting and replication of experimental results.

The MCP described herein is comprehensive and extensible. It is designed to be simple to build and easy to use while solving some of the most vexing aspects of implementing microfluidic experiments. While the system can be integrated into complex multi-instrument experimental setups, it is also designed to serve as a standalone entry point into microfluidics for research and teaching laboratories without being technically or financially demanding.

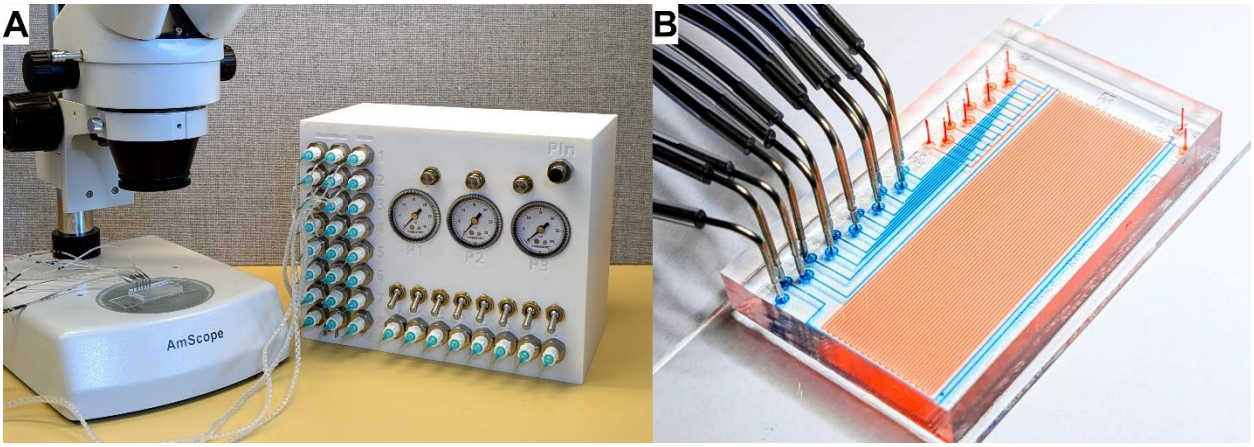


Figure 3-1. MCP hardware and a multilayer microfluidic device.
(A) The MCP with a benchtop stereoscope and a PDMS microfluidic device for reference. The controller natively supports devices with up to 24 control lines regulated at two pressures and 8 pressurized reagent ports. (B) A representative multilayer PDMS microfluidic device designed for miniaturized immunoprecipitation experiments with 12 elastomeric valves. The red dye indicates the flow layer where serially introduced reagents react in the on-chip assay. The blue dye indicates the control layer which contains elastomeric valves that manage reagent flow and are actuated by the pressurized solenoids of the microfluidic controller.

3.3 Experimental

3.3.1 Microfluidic Mold Design and Fabrication

A multilayer microfluidic device was designed with the PCB Editor utility in KiCad 9.0 with flow and control layers being exported independently as Gerber files. The microfluidic Quake valves were designed to have a push-up geometry, so the flow layer Gerber files for negative and positive photoresist were subsequently scaled up 1.5% to counteract polydimethylsiloxane (PDMS) shrinkage during the layer alignment process with a custom Python 3 utility. Gerber files were plotted onto mylar film by Fine Line Imaging at 12k DPI.

Microfluidic molds were prepared using the cleanrooms and photolithography resources available at the Biomolecular Nanotechnology Center at UC Berkeley. 4 inch silicon wafers (University Wafers) were rinsed with water, IPA, and acetone before being dehydrated at 105 °C for 10 minutes. The wafers were prepared with a 5 μm thick adhesion layer with SU-8 3005 (Kayaku Advanced Materials) according to the technical data sheet provided by the manufacturer. The control layer was prepared with 10 μm thick features with SU-8 2010 (Kayaku Advanced Materials) according to the technical data sheet provided by the manufacturer.

Rounded photoresist features for the valves on the flow layer were prepared with AZ 40XT-11D (Merck Performance Materials). The photoresist was spin-coated to a thickness of 50 μm by spreading at 500RPM for 10 seconds and spinning at 1400 RPM for 20 seconds. The wafer was then soft baked by ramping a hotplate from 60 °C to 120 °C over 5 minutes and holding at 120 °C for an additional 3 minutes. The wafer was then exposed at 450 mJ/cm^2 with an OAI Model 200 mask aligner and baked at 80 °C for 20 seconds before being ramped and held at 105 °C for 120 seconds. The photoresist was then developed with AZ 300 MIF (Merck Performance Materials)

puddles for 1 minute before briefly shaking and transferring to a new dish. This puddle development process was repeated for a total of 4 minutes. Features were then reflowed overnight by ramping a hotplate from 60 °C to 190 °C at 10 °C/hour, holding at 190 °C for 2 hours, and then ramping down to 40 °C at 30 °C/hour. Negative photoresist features were aligned to the positive photoresist and prepared to a 50 μm thickness with SU-8 3035 (Kayaku Advanced Materials) according to the technical data sheet provided by the manufacturer.

Following mold fabrication, all wafers were silanized with trimethylchlorosilane (MilliporeSigma) by placing a drop in an aluminum boat, evacuating a vacuum chamber with the wafers and TMCS for 60 seconds, and holding the chamber in a depressurized state for 10 minutes.

3.3.2 Multilayer PDMS device fabrication

Multilayer PDMS devices were prepared with RTV 615 (Momentive). The flow layer was mixed at a 5:1 base to crosslinker ratio, degassed for 45 minutes, poured to a thickness of 5 mm, degassed for an additional 20 minutes before being partially cured at 80 °C for 30 minutes. The control layer was mixed at a 20:1 base to crosslinker ratio, degassed for 45 minutes, spin coated by spreading at 500 RPM for 10 seconds, ramping to 2500 RPM over 15 seconds and holding for an additional 60 seconds. The wafer sat at room temperature for 10 minutes to relax streaks that formed during the spin coating. Afterwards, the control layer was transferred to an 80 °C oven for 30 minutes.

Once the flow and control layers had cooled to room temperature, the flow layer PDMS was peeled from the wafer and ports were punched out with an Accu-punch MP10 with a TEP-011 core and CR0250175N23T1 cutting edge (Syneo). The devices were then cut out, cleaned with Scotch tape (3M), aligned to the control layer and baked at 80 °C for an additional 90 minutes.

Afterwards, the devices were excised from the wafer, control ports were punched out, devices were cleaned with tape, and then devices and pre-cleaned glass slides (Thermo Fisher Scientific) were oxygen plasma treated with a PDC-001 oven (Harrick) at high power and at 650 mTorr for 30 seconds. Devices were then bonded to glass slides and incubated at 40 °C for 10 hours.

3.3.3 Valve actuation validation

Control lines were filled with water and pressurized at 10 PSI while the flow layer was filled with blue food dye as a contrast agent for imaging. A valve was imaged with a 10x objective and 4x4 binning for 2 ms with Micro-Manager 1.4 and an Andor iXon camera in an open and closed state using 10 PSI sourced from the microfluidic controller. The controller using custom OS was then scripted to actuate the valves at a frequency of 2 Hz for a total of 50 cycles to validate pneumatic behavior. 2 Hz was the selected frequency because it was empirically observed to be close to the maximum operating frequency for the geometry of the device and the applied pressure. The camera was set to image this dataset with 8x8 binning, a ROI of 16x16 pixels, the minimum exposure time available, and an internal trigger to achieve an average of 300 frames per second. Many images were found to contain duplicate timestamps in their metadata and were removed from the dataset so that each timestamp was represented by the first image available. The mean intensity of each frame was computed and the timepoints at which the intensity reached half-maximum were extracted. The mean edge-to-edge time and timing jitter were computed. Timing jitter was defined as the standard deviation of the edge-to-edge measurements. The process of cyclically actuating valves was also observed using 8-channel FT245R-based relay control boards (1982AY-USB, SainSmart) to validate the feasibility of an off-the-shelf alternative for the custom solenoid driver described herein; however, measurements were not recorded.

3.3.4 Solenoid driver timing analysis

The electrical performance of the system was assessed by bridging an output of the custom solenoid control board with a 1 kΩ resistor and measuring the voltage with a Siglent SDS804X HD oscilloscope at 5 MSa/s. The output was toggled at a frequency of 20 Hz for 5 s. This process was repeated using the control software that had been installed on a Raspberry Pi using a RaspiOS Trixie-lite image released on 2026-04-21, a Raspberry Pi with a custom optimized OS, and a Windows 11 laptop with an AMD Ryzen 7 Pro 4750U processor. The measurements were repeated once more by using only the custom Python 3 hardware drivers in a script to estimate the timing overhead of the control application software in each of the three hardware systems previously described. The oscilloscope for these burst measurements was set to acquire at 2 GSa/s. For each configuration, the mean edge-to-edge time, timing jitter, and maximum edge separation were measured.

3.3.5 Microfluidic control platform (MCP)

The MCP described here occupies a volume of roughly 7.6L (0.27 ft^3), costs $2500 to produce, and contains an extensive set of features that enable it for broad use with its modular design. The following subsections describe the fundamental design principles and behavior of each subunit. A complete build guide is available in **Appendix C**, a guide for operation is available in **Appendix D**, a bill-of-materials is available in **Appendix E**, PCB design documents, mechanical enclosure files for 3D printing, and custom code can be found in the associated Zenodo repository as listed in the “Data availability” section.

3.3.5.1 Pneumatic circuit

An external pressure source that is regulated and filtered is fed into the system where it is split into three units with each unit being independently regulated up to 200 kPa (30 PSI). The first unit consists of 16 independently addressable 3/2-way solenoid valves (Pneumadyne), the second unit consists of 8 independently addressable 3/2-way solenoid valves (Pneumadyne), and the third unit consists of 8 2-way manual toggle valves. Pneumadyne 10 mm valves have since been discontinued and support for equivalent SMC S070 series 3/2-way solenoid valves is provided in the supplementary information. Figure 3-2 provides a view of the pneumatic hardware and Figure C-2 presents a schematic for the pneumatic circuit. This multiunit configuration allows for downstream applications where multiple pressures may be required such as a device that operates pneumatic microvalves and on-chip micropumps. The surface area between a micropump membrane is typically significantly larger than that of a microvalve and necessitates the use of different ideal actuation pressures to ensure responsive actuation. The third unit is designed to uniformly pressurize input reagent reservoirs to drive fluid flow into the microfluidic device.

3.3.5.2 Enclosure

The control system is enclosed in a compact and portable shell that has a 370 cm^2 (0.4 ft^2) footprint and a heigh of 20 cm (8 in). It is divided into two 3D printed parts: the main body and the rear panel. The front face of the main body serves as a physical interface for managing the pressures of the three independent pneumatic units and the manual toggle valves while also providing output connections for all solenoid and manual valves. The inner cavity of the main body is used for routing and enclosing the pneumatic tubing and manifolds. The rear panel is used to support the solenoid valves and control board so that they remain accessible and maintainable.

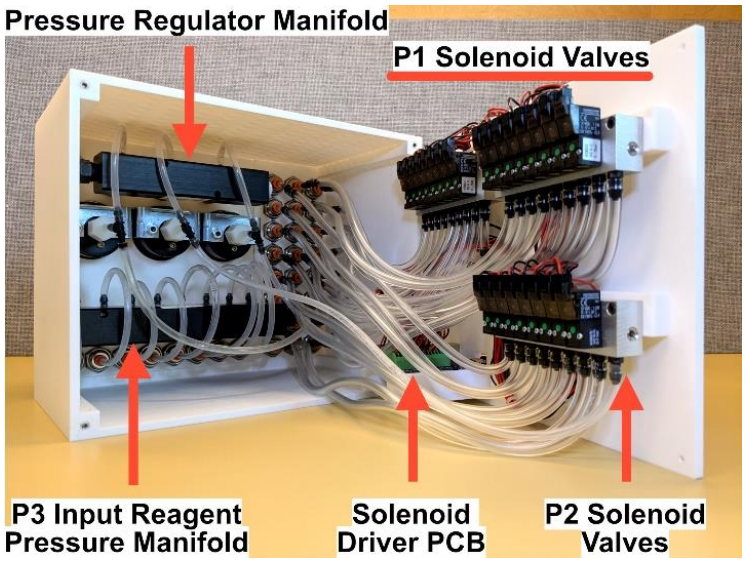


Figure 3-2. Internal View of the MCP.
The enclosure compactly contains the core pneumatic and electronic control systems of the microfluidic controller while also facilitating serviceability. House air is diverted to three separate pressure regulators which are subsequently fed to a set of 16 solenoid valves, a set of 8 solenoid valves, and a manifold that pressurized reagent reservoirs with 8 manually actuated toggle switches. The solenoid driver PCB can be seen behind the solenoid valve tubing in this view.

3.3.5.3 Solenoid driver PCB

The solenoid driver control board serves to translate instructions received from a computer over USB into solenoid valve operations. A custom printed circuit board (PCB) was developed to bridge PC communication with solenoids using (3) DRV81008 load drivers and an FTDI 4222H USB to SPI adapter (Figure 3-3). The custom PCB can control up to 24 solenoid valves with a voltage input ranging from 5V to 24V with a total maximum power rating of 96W or 170 mA per valve. This approach was used to enable feedback and diagnostics while also providing flexibility for solenoid selection. The standard USB-C interface also makes this system platform independent and allows for flexibility in deployment. The primary function of this board in controlling solenoids is not unique and can be substituted with alternative hardware, such as the Arduino and shield described by White and Streets[132] or a generic off-the-shelf relay controller. Using a different

solenoid driver will require writing a basic software driver in Python so that the microfluidic software can interface with the hardware of choice.

3.3.5.4 Raspberry Pi embedded computer

The computer (1) hosts a control program that executes scripts in the background while also managing a user interface (UI), (2) interfaces the control program with the solenoid driver board, and (3) provides network accessibility for a remote connection while also allowing for direct control with a monitor, keyboard and mouse.

Upon booting, a Raspberry Pi 4B hosts the UI as a Python-based Flask application with WebSocket support at port 5454 of the device. The application automatically generates a comprehensive log file of all operations that the operator or a script has performed allowing debugging of failed experiments and documentation of successful runs. New log files are generated at the start of each day and are retained for 30 days. Guidance for retrieving the logs is found in the operation guide.

For remote deployment, a high-speed mesh virtual private network (Tailscale) is optional and recommended. This network layer establishes secure, direct peer-to-peer connections without requiring local network administration privileges. Guidance for automatic registration of the MCP onto the virtual private network when using the provided custom Linux image is provided in the build guide.

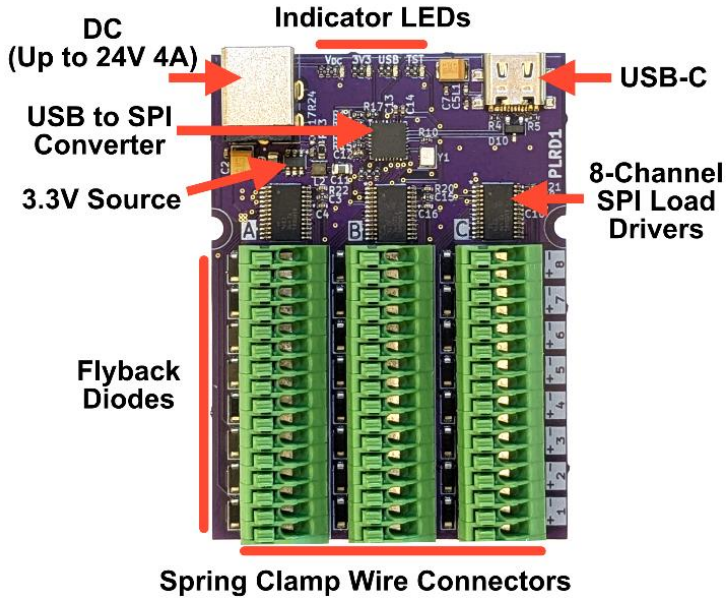


Figure 3-3. Optional custom solenoid driver PCB with callouts for core components.
The custom PCB enables the control of 24 solenoid valves via USB-C in a small credit card sized form factor. The board converts USB instructions into SPI commands with an FTDI 4222H to drive 3 independent TI DRV81008 load drivers. The board can be flexibly powered between 5V and 24V to support a variety of solenoid valves with a maximum amperage output of 170 mA per solenoid valve. This driver can be substituted with generic off-the-shelf relay driver boards at the expense of compactness, error detection utilities, and higher cost.

3.3.5.5 Custom operating software

To eliminate software dependency conflicts and combat the reproducibility challenges inherent in research hardware, a custom operating system (OS) was developed and compiled for the Raspberry Pi 4B using the Yocto Project build system and kas configuration manager. The resulting minimal OS image encapsulates the entire application, hardware drivers, mesh-VPN, system configuration, and implements a dedicated kiosk interface for headless or local operation. Compiling the OS via Yocto results in an exact replica of the software environment used in this study. Furthermore, it allows wireless network and VPN credentials to be preconfigured to enable immediate remote connectivity on first boot. While the custom OS is optional, it significantly streamlines software installation and platform development. The build process took 270 minutes

using 6 cores of a Ryzen 3700X (AMD) and 32 GB of DDR3 memory (Crucial). Refer to the build guide and associated Zenodo repository for guidance on building the image.

3.3.5.6 Software architecture

The Python 3 module developed here, *plfluidic*, serves a Flask-SocketIO application that only contains pure Python dependencies except for an installable FTDI USB for the custom solenoid driver PCB. This implementation allows for the application to be deployed on any computer that can run Python and install USB drivers. The *plfluidic* server was designed with a multithreaded model-view-controller (MVC) framework to separate concerns, facilitate modularity, and ensure real-time responsiveness (Figure 3-4). Within this framework, the views are HTML pages that are served to the user via a web browser. The controllers act as a translation medium between the models and views while also interfacing with external resources such as hardware drivers or logging handlers. The models store system parameters and process inputs. Extending system functionality to interface with different hardware such as relay controllers or syringe pumps is detailed in the supplementary build guide listed in “Data availability”.

The views interact with the application controller through HTTP requests and WebSocket connections. While designed for manual interaction, the standard use of HTTP and WebSockets also allow for other programs to interface with the controller as an API. This enables broad integration such as coupling automated microscopy programs with the microfluidic controller. This network-based approach also allows multiple clients to simultaneously connect and monitor the system state.

The primary application controller serves HTML pages, processes HTML and WebSocket requests, passes information to models for processing, and spawns threads to handle concurrent

communication and parallel execution of experimental scripts. Secondary controllers are used to translate model commands into hardware actions and to facilitate global system logging.

The models define and capture the state of the system, manage the formatting and processing of inputs from users or scripts, and issue directions to hardware drivers. To allow concurrent processing of automated scripts and user actions, the script model hosts a state machine that executes within a dedicated thread. This thread is terminated if the script completes or if the user stops the execution. The finite state machine for script execution is described in Figure D-7.

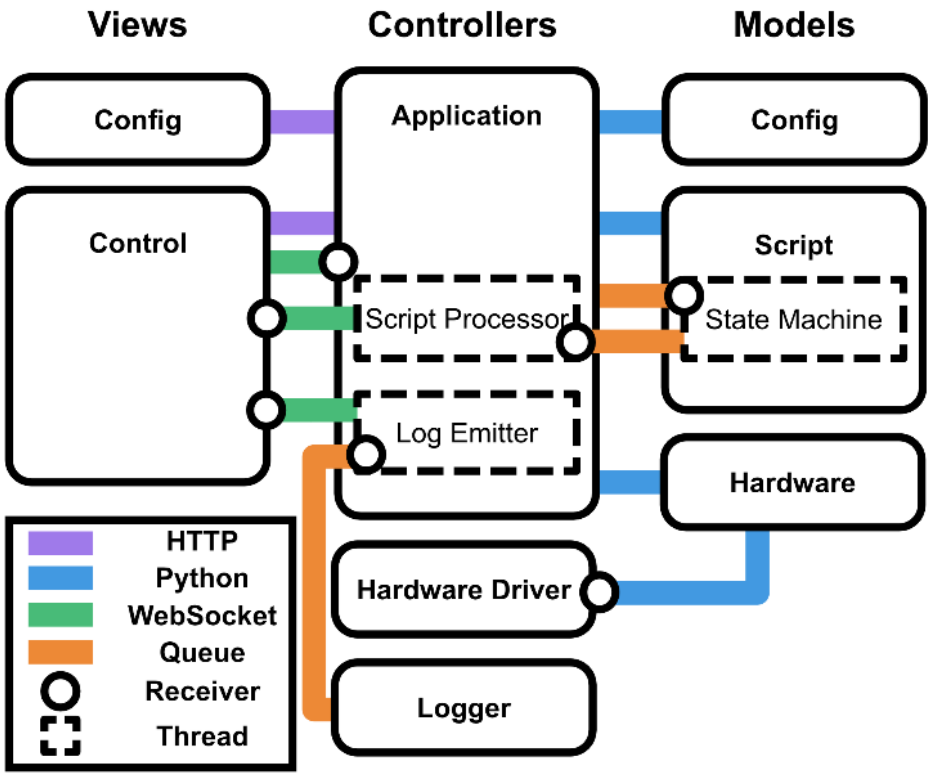


Figure 3-4. A simplified view of the model-view-controller framework implemented in the plfluidic Python 3 module.

The modular framework provides a flexible foundation that enables the deployment of different control views and hardware configurations on a single platform. This allows the system to seamlessly adapt to different environments and demands while still leveraging the underlying core utilities.

3.3.5.7 User interface

The UI consists primarily of configuration and control HTML pages that are served within a standard web browser (Figure 3-5). The configuration view is used to create or select a hardware configuration file corresponding to a specific microfluidic device design. Within this view, users can assign distinct names to individual solenoids, define initial hardware states, and select which HTML page to serve for the control interface. Configurations can be edited to make ephemeral modifications or be saved as new files.

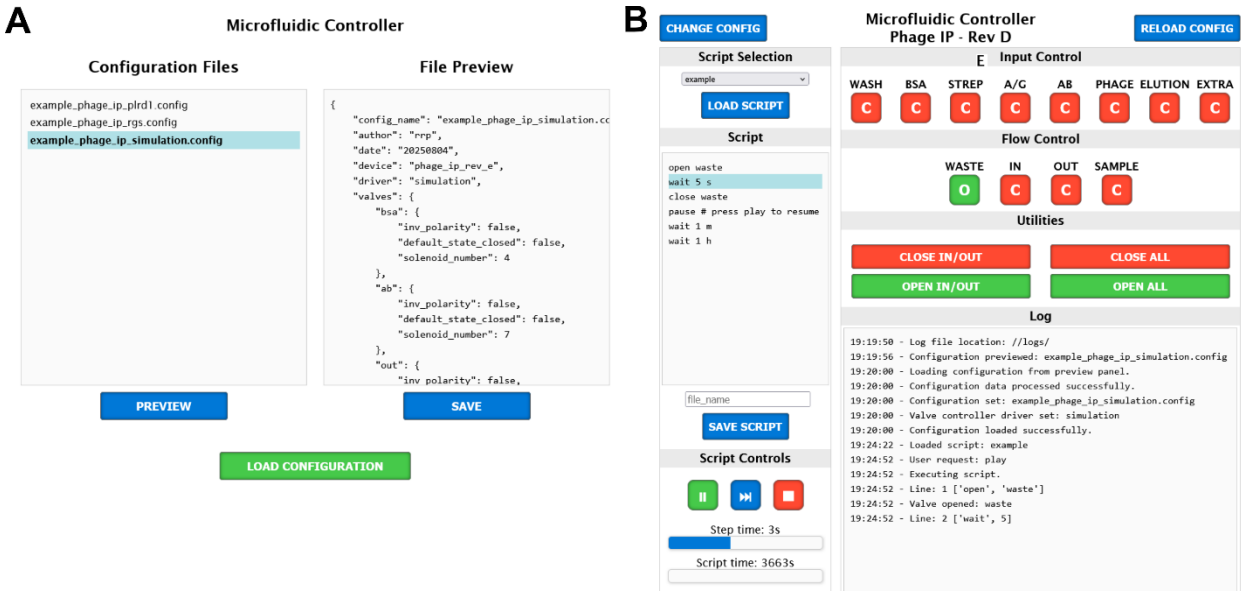


Figure 3-5. Configuration view (A) and control view (B) of the user interface as rendered in a Firefox web browser.

The configuration view provides an interface for viewing, editing, saving, and loading configuration files for the microfluidic controller. Configuration files define how hardware resources are allocated and what control view to load. The control view allows for manual control of hardware resources, the creation and execution of automated scripts, and a log stream. Manual controls are locked while a script is being executed, but they can be accessed once a script is paused or stopped.

The control view enables manual hardware actuation as well as the creation, saving, and execution of automation scripts with a human readable language. Additionally, it features countdown timers for the current step and entire script to estimate time of completion and it presents a real-time stream of system logs. Creating new control interfaces for different

microfluidic devices requires creating a copy of an existing HTML file and updating variable names to match the associated configuration file. The operation guide details this approach.

3.4 Results and discussion

3.4.1 Valve actuation validation

The pressurized and depressurized state of the elastomeric valve were clearly discernible as seen in Figure 3-6. While the valve was pressurized, residual dye could still be observed in the flow layer and may be attributed to hydrophobic binding to the PDMS or molecular trapping of the dye in the porous PDMS. When the valve was scripted cyclically actuate 50 times with a period of 500 ms, microscopy measurements found that the edge-to-edge timing of 291.6 ms and a timing jitter of 14.4 ms. This large deviation from the expected edge-to-edge timing of 250 ms may be partly attributable to challenges encountered in acquiring small ROI images at a high frame rate. Of the 96973 images that were captured, only 7287 had unique timestamps and were used in the analysis. The system was also observed to have buffering issues during acquisition which may have affected the timestamps captured in the metadata. While the system effectively actuates elastomeric valves, timing performance of the system is better captured in the following solenoid driver timing analysis section. When the custom solenoid driver board was substituted with generic commercially available FT245R-based relay controllers, the system was also observed to effectively actuate elastomeric valves. Caveats of using these boards include uncontrollable valve actuation upon USB connection and slower valve response times as the on-board relays and the solenoid valves have additive energizing and deenergizing times on the order of tens of milliseconds.

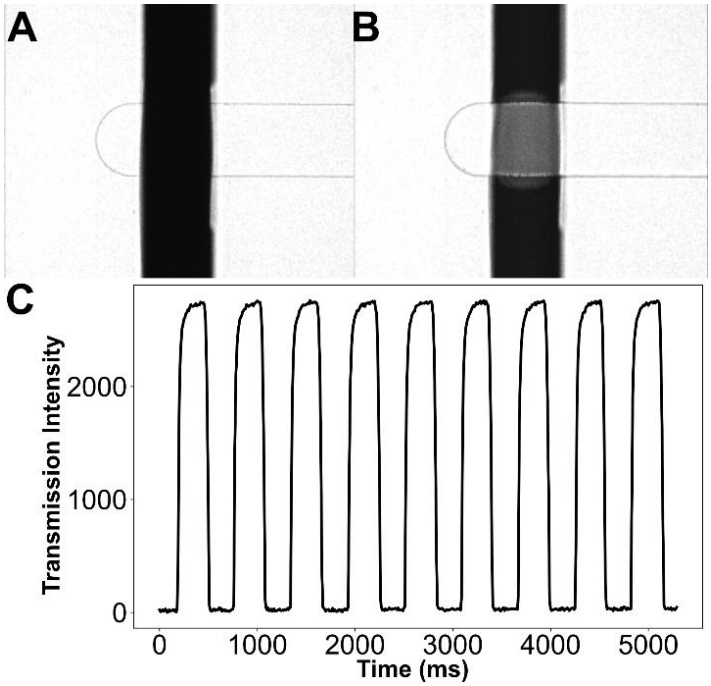


Figure 3-6. Representative data of elastomeric valve actuation.
Subfigures (A) and (B) demonstrate observable changes in opacity of a flow channel loaded with food dye when the overlapping control channel is in a depressurized and pressurized state, respectively. Subfigure (C) depicts the average light transmission intensity of the valve as it is cyclically actuated for 5 seconds at a rate of 2 Hz.

3.4.2 Solenoid driver timing analysis

When the system was scripted to toggle a solenoid at 20 Hz using the provided user interface, the Raspberry Pi with a default OS, Raspberry Pi with a custom OS, and Windows 11 systems exhibited similar overall behavior with minor operational variations (Figure 3-7A). The system with a default OS had a mean step time of 55.20 ms with a jitter of 1.33 ms and a maximum edge separation of 58.15 ms. The system with a custom had a mean step time of 58.77 ms with a jitter of 0.60 ms and a maximum edge separation of 60.67 ms. The Windows 11 system had a mean step time of 58.97 ms with a jitter of 3.91 ms and a maximum edge separation of 66.40 ms.

When the systems were reconfigured to utilize only the custom Python driver to toggle the state of the output state of the custom solenoid driver PCB as quickly as possible, more pronounced

distinctions between the systems appeared (Figure 3-7B). Under these burst conditions, the system with a default image had a mean step time of 249.2 μs with a jitter of 95.3 μs and a maximum edge separation of 354.9 μs. The system with a custom image had a mean step time of 324.3 μs with a jitter of 101.4 μs and a maximum edge separation of 518.8 μs. The Windows 11 system had a mean step time of 440.8 μs with a jitter of 135.9 μs and a maximum edge separation of 1186.4 μs.

The experimental results align closely with the expected architectural behaviors of the evaluated operating systems. The default Raspberry Pi OS leverages four processor cores to execute the Python application alongside a moderately lightweight set of background processes. While this resource availability yields rapid script execution, background process competition introduces intermittent latency, as evidenced by the bimodal distribution of step timings in Figure 3-7A.

In contrast, the custom, minimal OS dedicates two specific cores exclusively to the Python application. This isolation successfully yields a tighter, monomodal step-timing distribution with a jitter of 0.60 ms vs 1.33 ms. However, the nominal execution speed was slower, which is likely a consequence of fewer available cores or a lack of advanced scheduler optimizations present in the minimal Linux image. Furthermore, the higher variability observed in the custom OS burst datasets occurred because the raw driver scripts ran across the remaining two non-dedicated cores.

Finally, while the Windows 11 system possesses significantly greater raw computational power, it simultaneously runs intensive background architecture competing for CPU cycles. This environmental overhead resulted in slower mean

execution times and the highest timing variability across both testing modes. In the burst datasets, the observed bimodal distributions are attributed to High-Speed USB 2.0 communication

protocols. High-Speed USB is typically polled at 8000 Hz and, if a command is not actively queued in the buffer precisely when polled, an inherent propagation delay of up to 125 µs is introduced.

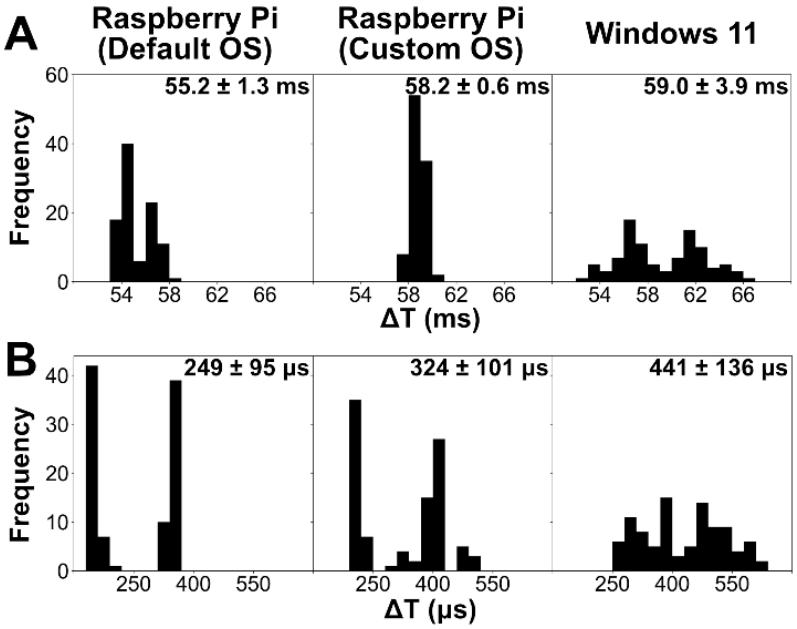


Figure 3-7. Distributions of edge-to-edge timings of an output on the solenoid driver PCB under different modes of control.
Subfigure A presents edge-to-edge timing data for a script that toggles the output every 50 ms using the provided UI while subfigure B presents edge-to-edge timing data for the output as it is toggled without delay using the included Python driver communicating directly with the solenoid driver PCB without passing through the provided UI.

3.4.3 Application examples

The modularity and breadth of the MCP described here allows for significant operational flexibility. The system can be utilized as-is for the operation of pneumatically actuated microfluidic devices, extended to support other fluidic modalities such as constant flow with syringe pumps, or integrated into complex, multi-instrument workflows.

3.4.3.1 A multi-pressure pneumatic controller

Nobs and Maerkl previously introduced a microfluidic chemostat array with 120 chambers for long-term single cell analysis of *S. pombe* to track over 100,000 divisions in a single

experiment.[138] Their device architecture required 18 control lines and 6 reagent inputs. While most valves hermetically seal flow channels with a height of 16 μm, two valves are used to gently compress the cells within shallower 4 μm chambers. The disparity between two feature heights necessitated a dual control pressure setup to secure the sample while also sealing the flow channels.

The platform developed in this work natively accommodates such requirements by providing 24 control lines that are managed by two independent pressure regulators. These pressure rails divided into groups of 16 and 8 solenoid valves. In addition, the system has a capacity to pressurize 8 input reagents without modification. For assays that require more than 8 input reagents, external manifolds can be seamlessly coupled to the existing pressurized reagent ports.

3.4.3.2 A multimodal microfluidic control platform

To study cellular dynamics, Quintard *et al.* developed a microfluidic system designed to stimulate a single pancreatic islet cell with glucose and subsequently collect the secreted insulin.[139] This application required both constant pressure control for actuating elastomeric valves and constant flow control to maintain physiological shear conditions and precise volumetric delivery rates.

The extensible design of the MCP described here has the capacity to support multimodal devices without further development. In addition to the solenoid valves built into the system, support for syringe pumps has also been developed. The system includes drivers for New Era syringe pumps and an example configuration and control view for multimodal operation. The system can be extended to support other syringe pumps as detailed in the build guide and requires writing a Python wrapper for the communication protocol of the syringe pump using the existing base class. Coordinated control of multimodal fluidic systems is of particular importance given the

observations made by Zhou *et al*.[140] In their work, they demonstrated that synchronized control over pressure and flow sources is critical to minimizing the non-trivial impact of transient flow perturbations on sensitive cell cultures.

3.4.3.3 Integration into existing infrastructure

Many microfluidic experiments are comprised of a combination of measurement and device control hardware that are tightly coordinated.[117] In cases where timing is critical, such as the microfluidic droplet sorting using integrated bilayer valves developed by Chen *et al*.[141], the platform can be implemented by bypassing the web server entirely. Sub-millisecond communication with the solenoid control board can be achieved via USB by using the included Python 3 drivers. If timing latencies are not a limiting experimental parameter, the web server can be deployed either locally or remotely, which renders the system fully controllable and programmable via the standard HTTP and WebSocket interfaces.

3.4.4 Limitations of the platform

While the microfluidic control platform described here is fully capable of controlling elastomeric valved microfluidic devices, certain areas remain open for further development depending on the target application. Currently, the integrated scripting language is basic and does not natively support parallel valve operations even though the underlying drivers possess the flexibility to do so. In many microfluidic experiments, a latency of less than 10 ms between sequentially closing valves will not meaningfully impact experimental outcomes. However, for time critical applications where this delay is prohibitive, the solenoid control PCB can be operated directly via USB and the provided Python driver, or the control program can be modified to allow for concurrent valve operation. Time sensitive applications should also consider the step latencies

characterized here to operate valves at specific frequencies. The scripting language also lacks advanced control structures such as macros or loops, which could simplify complex or repetitive device operations.

Regarding timing optimizations and speed, the software architecture would benefit from being organized as a multiprocess application rather than a multithreaded framework. In the current multithreaded system, threads share a Python global interpreter lock. This restricts many processes to a single core while the interpreter rapidly switching between threads to simulate concurrency. A multiprocess application spawns independent interpreters for each process and enables parallel execution on multiple cores. This is expected to significantly reduce the step latency observed in the UI.

Similarly, deterministic execution could be achieved with the custom OS image if it is compiled with realtime Linux patch (RT_PREEMPT) to allow for the control program to interrupt low-priority background processes to ensure the program executes with guaranteed sub-millisecond latency. To contextualize the timing limitations, the solenoid valve driver board can reliably drive 3 solenoids for an on-chip micropump at 25 Hz with the UI or 650 Hz using the Python driver directly on a Raspberry Pi 4B with a default OS image.

Even though the solenoid driver PCB can achieve drive frequencies up to 650 Hz, the functional operation frequency may be fundamentally limited by the geometry of device features and the physical limitations of the solenoids used. The Pneumadyne 10 mm valves have minimum deenergizing time of 10 ms, which makes a response frequency of more than 50 Hz infeasible. Pneumadyne solenoid valves have also been discontinued and were used in this instrument because they were accessible at time of development. This should not be considered a hinderance given that the custom solenoid driver PCB was designed to for flexibility and can drive solenoid valves

between 5V and 24V with a power draw of 4W per valve. The SMC S070 series solenoid valves are a recommended alternative with energizing times that are at least twice as quick as the Pneumadyne valves used here while being lower wattage with the same flow capacity.

Finally, the system architecture relies on a custom solenoid control board. While this allows for a very small form factor system with on-board diagnostics, it introduces manufacturing barriers that may make it challenging for other labs to produce. As an effort to improve accessibility, a driver for a generic relay control board was developed and lightly tested to validate functionality. Utilizing these off-the-shelf boards function as intended, though implementing them will require adjusting the design of the 3D printed enclosure. In addition, the commercial board tested here briefly actuates valves uncontrollably upon USB connection, which may impact some devices if connected before the controller is powered.

3.5 Conclusions

The work presented here describes a compact, low-cost microfluidic control platform designed to improve the accessibility automated microfluidic devices in research laboratories. The feature rich, custom *plfluidic* Python module serves as the foundation of the system and integrates a multithreaded model-view-controller (MVC) software framework with either custom or commercial off-the-shelf hardware. The MCP bridges the gap between cost-prohibitive commercial platforms and fragile, application-specific control systems while being simple to build and operate. The platform introduces multi-pressure capability with 24 control lines regulated by two independent pressure rails and multimodal compatibility with syringe pump driver support within a physical footprint of less than 0.1 m^2 (1 ft^2) and an approximate build cost of $2,500 USD. It also includes extensive logging to facilitate device troubleshooting and experimental reproducibility. This implementation significantly lowers the financial and technical barriers to

advanced microfluidic device control and may serve as an entry point to microfluidics in research and teaching laboratories where ease-of-use and experimental documentation are important criteria.

Quantitative characterization of the execution timing performance demonstrated its capability to meet diverse experimental demands. When operated via the browser user interface, the platform demonstrated consistent behavior across multiple computer architectures and can reliably operate a 3 solenoid on-chip micropump at 20 Hz. An optional, custom compiled and reproducible Linux environment was also developed for a Raspberry Pi 4B and observed to reduce the step timing jitter down to 0.60 ms. For high-throughput or time critical applications, such as microfluidic droplet sorting or rapid micropumping, bypassing the UI and communicating directly with the underlying Python driver achieved sub-millisecond execution, enabling an upper limit drive frequency of on-chip micropumps at 650 Hz with a standard operating system. The functional maximum pumping frequency is fundamentally limited by the device geometry and the solenoid valves used. Literature-backed case studies further validated that the hardware natively satisfies the constraints of challenging dual-pressure assays, can adapt to synchronized, multimodal setups, and can integrate into existing hardware infrastructure pipelines like automated microscopy.

While the current framework relies on a basic scripting interpreter and shared thread resources, well defined optimization pathways exist to adapt the system for deterministic, sub-millisecond workflows via multiprocessing architectures and real-time kernel patches. By offering a series of hardware drivers, the system is also compatible with widely available FT245R-based relay driver boards and New Era series syringe pumps in addition to the custom solenoid driver. These drivers may also serve as an adaptable, accessible foundation to extend support to other hardware. Ultimately, this platform simplifies the deployment of sophisticated microfluidic

automation, enabling broader replication and improved experimental reproducibility across the interdisciplinary scientific community.

3.8 Data availability

The data and design files supporting this study are available through both supplementary material and public repositories.

Python application software, Yocto build files, KiCAD PCB files, PCB characterization datasets, and processed valve image data The files associated with these resources can be accessed through a Zenodo record at https://doi.org/10.5281/zenodo.22140981. Alternatively, it can be accessed on GitHub at https://github.com/robertpuccinelli/plfluidic-controller-resources.

Build guide, operation instructions, bill-of-materials These documents are provided in Appendices C, D, and E.

Valve actuation images This raw data can be made available upon request and has not been uploaded due to file size constraints. The processed image data is included in the repository listed above.

Chapter 4.
Exploration of automated assays for immunoprecipitation and disease diagnostics

4.1 Introduction

The following chapter is a collection of unpublished experimental work that outlines the development of methods for automating the immunoprecipitation aspect of the PhIP-seq protocol while also touching on other diagnostic systems. Section 4.2 begins with an exploration of macrofluidic prototype instrumentation for automating the phage immunoprecipitation washing protocol before discussing the development of a microfluidic platform in more detail. Section 4.3 touches on the feasibility of using a Luminex bead-based assay to screen for autoimmune antibodies against a panel of epitopes that have been previously validated in other experiments before discussing work that explored the feasibility of a multiplexed nucleic acid detection platform using spatially arrayed CRISPR-Cas endonucleases.

4.2 Continuous flow approaches for phage immunoprecipitation

Displaying peptide fragments on the surface of bacteriophages has led to the development of interesting and exciting approaches for detecting the presence of antibodies targeting specific protein fragments via immunoprecipitation and sequencing. The phage immunoprecipitation sequencing approach (PhIP-seq) has been effectively utilized to detect novel autoantibodies in both rare diseases without commercially available diagnostics as in the case with ROHHAD and as a method to discriminate between many symptomatically similar but molecularly distinct cases of autoimmune encephalitis.

Despite the exploratory and diagnostic utility of the technology, the approach is manually intensive to implement and error prone over the course of the multi-day protocol. Implementing semi-automated methods to facilitate the washing steps of the assay would be beneficial in that it would 1) standardize the sample handling steps and reduce variability between plates and experiments, 2) reduce nonspecific interactions and improve the signal-to-noise ratio with more thorough washing techniques, 3) allow for method extensibility to explore new biochemical information such as gradient elution to assess binding kinetics of autoantibodies and 4) reduce the hands-on time of the assay that would otherwise be spent on low cognitive load tasks.

Assays that process many samples in parallel often leverage commercially available automated pipetting robotic systems or benchtop plate washers to improve throughput and obtain consistency. While these are often effective tools, phage immunoprecipitation has some challenges at scale that make them difficult to implement. Each plate is washed five times per round of enrichment and amplification. A typical three round experiment would consume a minimum of 2880 tips for washing a single plate and multiple plates are washed in each experiment. The volume of tips used makes the use of robotic pipettors financially impractical.

Similarly, plate washers also have limitations that make them unsuitable for this specific application. These systems typically have a unit for shaking the plate and pulling down magnetic beads and a separate unit for dispensing and aspirating wash buffer. The unit that is used for aspiration is typically rinsed with wash buffer between sample plates, but it is never thoroughly cleaned or replaced. The PhIP-seq method leverages exponential amplification of phage and the risk of bead carryover early in the protocol or phage carryover late in the protocol poses sample integrity concerns when handling multiple sample plates.

Several macrofluidic and microfluidic continuous flow approaches were developed and tested to assess their feasibility in enabling semiautomated sample processing. While traditional approaches wash sample plates by repeatedly dispensing and aspirating a fixed volume in a well, implementing a system with continuous flow allows for optimization of flow rate and flow volume to maximize signal-to-noise while possibly reducing assay time and the number of materials consumed. Of the systems explored, most macrofluidic systems could be eliminated as practical approaches through empirical observation without quantitative support, and they are mentioned to maintain a record of work. A microfluidic approach remains to be a promising candidate for automated PhIP washing at the expense of being a low throughput process and requires further development to effectively implement.

4.3.1 Macrofluidic magnet array

While a plate washer pulls down magnetic beads, exchanges a fixed volume of wash buffer and resuspends beads, an inverted system where beads are pulled out of a plate and transferred to a washing platform would allow a continuous washing modality. The system explored here consisted of a handheld magnet array, a disposable sheath to interface the magnet array with the sample, and a sample washing platform. The magnet array consisted of 96 2” long 1/8” diameter axially aligned N52 magnets that were pressed into a 3D printed tool with a sheath ejection mechanism as shown in Figure 4-1. The sheath had a wall thickness of 0.8 mm and tip at each magnet had a shallow concave geometry that was used to guide the beads towards settling in the center of the magnet and reduce the risk of beads shearing off the sheath during the wash. Edge effect of the magnets were observed to cause beads preferentially settling away from the center of the flat face of the magnet, which is why the concave design was implemented. Both the sheath

and the 3D printed tool were composed of PLA for an initial non-biological assessment of bead capture and retention.

The process consisted of mounting a sheath onto the magnet array, lowering it into a deep well plate with 1 mL of PBS and 20 uL of Protein A coated Dynabeads, and then transferring the tool to the washing platform to be washed with RIPA buffer. After the beads had been washed, they would be transferred to a new plate where the sheath would subsequently be ejected so that the beads could be resuspended in PBS. The washing platform was controlled by a Masterflex peristaltic pump where the inlet and outlet were both actively pumped. The wash platform was designed to pump buffer into the base of each well where it would then flow out the top and collected into a drainage system. While this design poses a risk of cross contamination at the top of the sheath, this risk is mitigated in practice as the top of the sheath does not contact the subsequent resuspension buffer. Instead, this design focused on prioritizing the effectiveness of bead washing by placing them in close proximity to the buffer inlet.

Empirical observations indicated that the design was limited in its effectiveness in collecting from the well plate and retaining beads during the wash. Primary limitations included 1) failure to capture beads that had settled to the bottom of the well plate, 2) significant aggregation of beads to the side of the roughly textured sheath rather than the concave tip, and 3) loss of beads on the side of the magnet by mechanical shear when transferred to the washing platform. In total, it was estimated that more than 10% of beads were lost during the functional assessment study. In the PhIP-seq assay, many significant diagnostic markers of disease occupy less than 1% of the total sequencing reads and a loss of 10% of the total sample would likely lead to many false negatives. Given that where patient sera is typically only available in small quantities and the PhIP-seq takes multiple days to process the sample, a loss of this magnitude would be unacceptable.

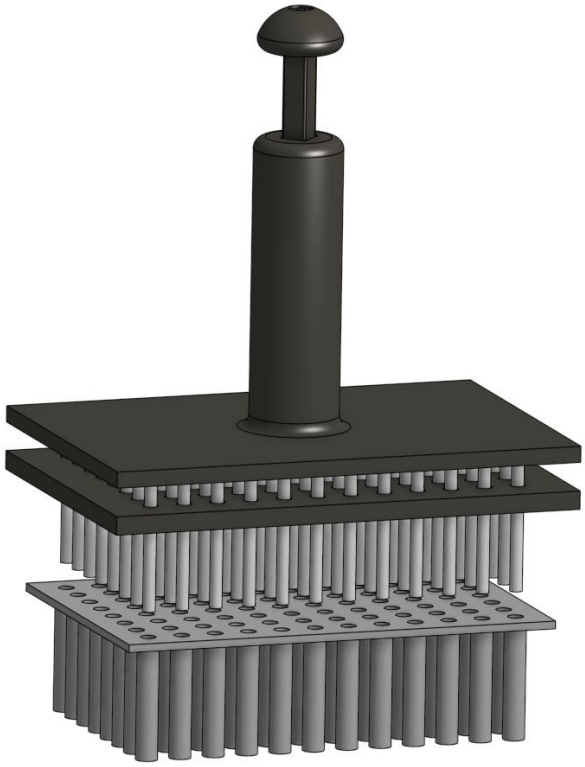


Figure 4-1. Magnet array for magnetic bead washing.
The magnet array tool uses a disposable sheath that sits over the magnets during the assay and can be ejected with the built-in plunger.

Despite the limitations, it is clear in retrospect these are engineering challenges that are addressable and a practical solution may still exist in this magnet array approach. Some of the beads could not be pulled out of the plate once they had settled to the bottom of the well. This is mostly due to the use of a 2” long magnet in wells that are 2” deep. The magnets did not extend to the bottom of the wells because several millimeters of the length were used to press them into the 3D printed tool for retention purposes. Specialty magnets of a longer length could have been ordered from manufacturers to ensure that the magnet reached the bottom of the well. Similarly, the aggregation of beads to the side of the sheath could have been counter acted by modifying the protocol and spinning down the sample plates with beads so that they would pellet to the bottom. In combination with a longer magnet, sample retrieval should have been trivial. If there were no beads on the side of the magnet, then loss of beads from mechanical shear into the washing

platform would also be nonexistent. However, minimizing contact with the sidewalls of the washing platform would still be recommended to minimize risks of contamination. Incorporating guide rods into the washing platform and linear bearings into the 3D printed tool would have allowed for controlled lateral positioning of the magnet arrays in the washing platform. The repeatable positioning of this approach would have eliminated the risk of side wall contact and maintained sample integrity.

Implementing the recommended design adjustments would likely resolve the observed sample loss issues. The next steps for validating this platform would be to characterize flow rate and duration parameters for minimizing background signal while maintaining specificity. Minimizing background could be assessed by incubating magnetic Protein A/G Dynabeads in a phage library without antibodies and then measuring the quantity of nonspecific phage retained after washing with qPCR. This process would be repeated for a combination of flow rates and durations. To assess the signal retention, magnetic Protein A/G Dynabeads would be incubated in a phage library with a control antibody such as GFAP, amplified with E. coli lysis, and subsequently measured for quantity of the control epitope with qPCR. This process would be repeated using the same flow and duration conditions as the nonspecific study. The parameter space could then be mapped and an optimal condition would be where the ratio of GFAP to nonspecific phage is maximized. To further reduce the quantity of nonspecific phage, alternative materials and fabrication methods to produce a sheath without a rough surface should be explored. The next phase of studies would focus on comparing the traditional benchtop PhIP-seq protocol to the protocol replicated with the magnet array and continuous washing platform using the optimal wash duration and flow rate conditions. The assay is naturally noisy and has high variability, but

consistency in amplification of epitopes targeted by control antibodies and characterized patient sera would be considered a success.

4.3.2 Macrofluidic in-tubing magnetic bead washing

Significant magnetic bead loss was observed with the previous approach which captured beads on the exterior of a sheath, so an alternative approach which captured beads within a microbore tubing array was explored. This system was designed to aspirate beads from a well plate, magnetically capture them within microbore tubing, thoroughly wash, and then dispense the beads into a new well plate. Given that there were no direct mechanical manipulation of the beads and the beads would interface with significantly less, the alternative design was expected to be more efficient at capturing and washing the sample.

The initial design of the system is shown in Figure 4-2 and consisted of 21 gauge blunt end Luer-lok needles that were arrayed into a 3D printed body and connected to flexible microbore tubing that was passed through the center of a 0.5 inch solenoid actuator that had the metal pin removed. The idea was that the electromagnetic solenoid could be activated when the beads were to be captured and washed, and it could be deactivated to dispense the washed beads. The solenoid was included in the design as the electronic control of magnetic bead capture would be greatly facilitate assay automation and the center cavity of the solenoid ideally has a uniform magnetic field so all tubing would experience identical magnetic force. Fluid flow was controlled by a Masterflex peristaltic pump which could be switched between dispensing and aspirating directions. The pump was connected to an array of manifolds that would split the flow into 96 separate channels to interface with a well plate. The needle array was designed to seated directly over a 2” deep well plate with the needles extending to the bottom of the wells. After aspiration and bead

capture, the system was moved to sit over a 3D washing platform which dispensed RIPA wash buffer into 96 chambers at the same rate at which the buffer was aspirated by the needle array.

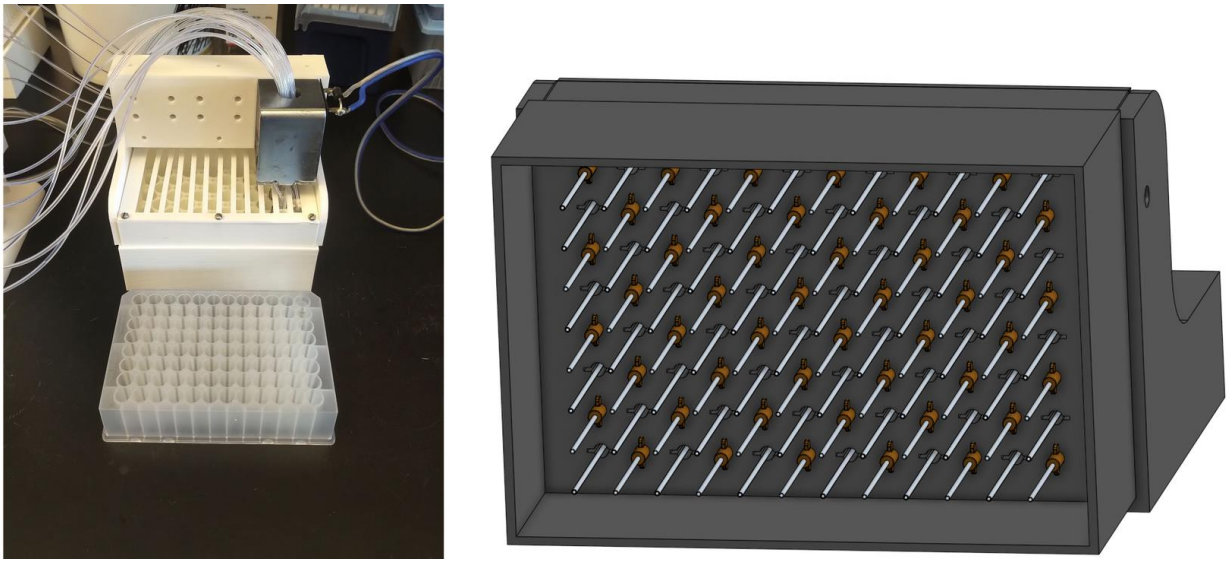


Figure 4-2. In-tubing bead washing platform.
Needles aspirate the contents of a well into clear flexible tubing that passes through the center of an electromagnet. The electromagnet can be selectively powered and depowered to capture and release magnetic beads. The version with a sliding permanent magnet fixture is not shown.

Under initial testing conditions of aspirating at 1 mL/min and running the electromagnet at the rated 12V voltage for continuous operation, it was observed that both the magnetic field of the solenoid was not strong enough to retain the beads and alternating needle lengths to accommodate packing at a 9 mm pitch to interface with the well plate lead to different fluidic resistances between neighboring wells. During aspiration, the different resistances led to some wells being drained quicker than others and aspirating air into the line while other wells still had sample remaining. Uniformity of sample processing and retention of magnetic beads were considered to be important factors and led to a secondary design.

The second approach replaced the blunt end Luer-lok needles with 21 gauge stainless steel tubing that was directly inserted into the Tygon tubing and the solenoid was replaced with permanent magnets that were mounted onto a sliding mechanism to change the proximity of the

magnets to the tubing to influence the retention of magnetic beads. The stainless steel tubing was placed into the 3D printed body at a pitch of 9mm with uniform length. While testing the system under identical conditions, uniform aspiration and dispensing rates were observed and beads appeared to be captured and released within the tubing. However, it was also observed that beads had been trapped at the interface where the needle had been inserted into the Tygon tubing and the could not be washed away. The trapping of beads within the tubing posed a significant risk of cross contamination between sample plates and would require a through rebuilding of the tool to clean the needles and cut the ends of the tubing between each plate, which is not practical.

While further development of this in-line magnetic bead washing platform was not pursued, failure at the stainless steel tubing to Tygon tubing interface could be resolved by replacing both components with a more rigid tubing such as PEEK. In this approach, the PEEK tubing would extend from the flow distributors directly to the sample wells. While this may have worked, any surface that touches samples risks carryover between sample plates and should be regularly maintained if not treated as disposable. PhIP-seq experiments are particularly sensitive to contamination as residual phage is exponentially amplified over the course of several rounds and early contamination may skew assay results.

4.3.3 Microfluidic Immunoprecipitation of Phage

Previously explored macrofluidic approaches risked significant sample loss or contamination from sample carryover, so a microfluidic immunoprecipitation approach without beads was developed. A device containing a single channel for immunoprecipitation would mitigate risk of contamination at the expense of being low throughput. The tradeoff for a low throughput approach would be speed and automation capabilities, which are desirable features in

cases where an urgent and highly multiplexed diagnostic panel for identifying autoimmune diseases is needed.

The microfluidic approach implemented here utilized a multilayer PDMS device bonded to glass microscope slides with Quake valves for controlling the flow of reagents through the device. The top layer of the device served as the flow layer where the immunoprecipitation reaction occurred and the bottom layer served as the control layer for valve actuation. Information detailing the process of microfluidic mold fabrication and PDMS device fabrication can be found in **Chapter 3**. Microfluidic devices in initial experiments were controlled by an ad-hoc pneumatic control system that operated at a single pressure, and later experiments were controlled by the controller described in **Chapter 3**. Information regarding experimental device preparation and operating the published microfluidic controller can be found in **Appendix D**.

4.3.3.1 Device Description

The flow channel had a cross section that is 160 μm in width and 40 μm in height with a path length of 1250 mm. In total, the surface area of the immunoprecipitation channel was 500 mm^2 and it enclosed a volume of 8 μL. These dimensions were selected to achieve a surface area that was comparable to the surface area of 20 μL of Dynabeads used in in the benchtop assay, which had an approximated surface area of 244 mm^2. The benchtop assay also used 0.5 μL of serum which has an average concentration of 10 μg/μL of IgG[19]. The IgG occupies a surface area of 211 mm^2 if one assumes that it has a hydrodynamic radius of 5.8 nm[20]. This resulted in an ideal occupancy rate of 42.2% for the device and 86.4% for Dynabeads.

With the aid of computational fluid dynamics simulation in Comsol, the flow channel was computed to have a resistance of 4.65 PSI•min/μL resulting in volumetric flow rate of 2.15 μL/min

at 10 PSI with a maximum velocity of 9.73 mm/s at the center of the channel. This is equivalent to a flow rate of 0.27 volumes/min and the time to exchange 1 device volume is 223 s while the leading edge of the flow profile takes 129s to traverse the channel. Multiple device designs were evaluated with minor variations to remove the on-chip micropump and reduce the dead volume following the valves to minimize background caused by phage diffusing into the channel during elution. These modifications did not impact the fundamental device characteristics, however. The initial integrated micropump had a geometry that was adapted from Goulpeau et al. It consists of 3 active membranes that had an overlap of 550 μm x 700 μm with the flow channel to displace fluid. This resulted in a volumetric displacement of 9.1 nL per pump cycle. A drive frequency of 5 Hz resulted in a flow rate of 2.73 μL/min and the time needed to recirculate one device volume was calculated to be 176 s. The initial and final device designs are presented in Figure 4-3.

The microfluidic approach utilized a surface functionalization strategy to capture antibodies from sera, immunoprecipitated phage, and subsequently elute the captured phage for downstream amplification as depicted in Figure 4-4. The surface chemistry approach is similar to previously published methods used by the MITOMI microfluidic device, but it uses biotinylated protein A and protein G to capture antibodies from sera. In short, the process involved creating a molecular sandwich on the device surface by passivating the flow channel with biotinylated BSA, followed by activation with streptavidin, and functionalization with biotinylated protein A and protein G. Control antibodies or sera would then be introduced into the device to bind to protein A and G. Afterwards, phage was then be flowed through the system and immunoprecipitated to antibodies that interact with the human peptidome, washed, and eluted for off-chip amplification with E. coli to complete one round of enrichment and amplification. This process included a RIPA wash between each of the steps where reagents were introduced.

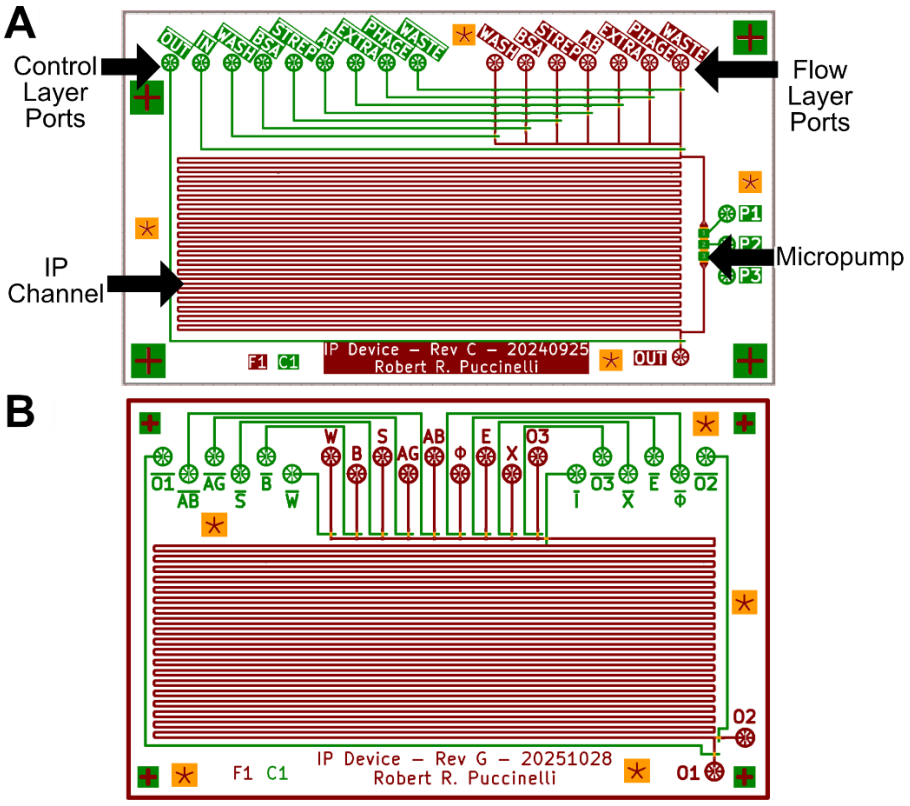


Figure 4-3. Evolution of immunoprecipitation device design.
While the core function of the device stayed constant over multiple revisions, several changes were implemented to improve performance. A) An early design with an embedded recirculating pump with callouts for key features. B) The final device design included extra inputs for additional reagents such as elution buffer, and extra outlet to separate device waste from phage elution, minimal overlapping control lines to ensure that all input channels had uniform channel resistance, reagent valves that terminated close to the main channel to minimize dead volume, longer valve geometries to facilitate layer alignment, and a removal of the recirculating micropump due to its association with many device failures.

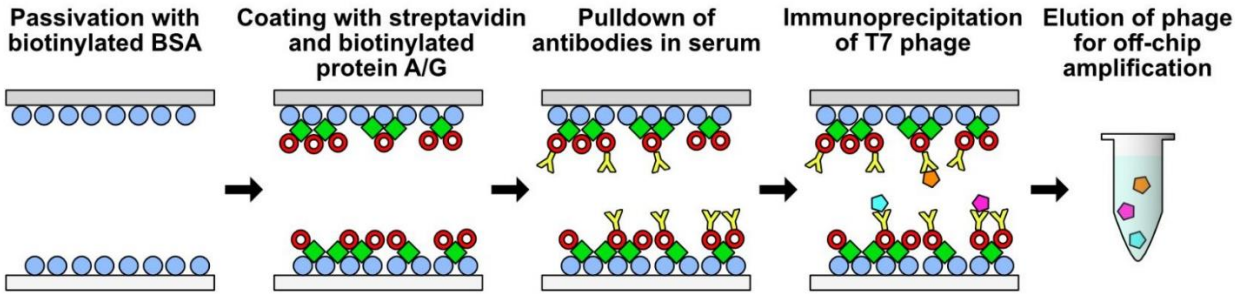


Figure 4-4 Microfluidic Immunoprecipitation and Elution Strategy.
The PDMS microfluidic flow channel is passivated with bBSA before being functionalized with streptavidin and protein A and G. Afterwards, antibodies can be flowed into the device and captured by protein A and G. Next, a T7 phage peptide library is immunoprecipitated, washed, and eluted for down stream amplification with E. coli.

4.3.3.2 Characterization of the Immunoprecipitation Surface Functionalization in a Well Plate

The molecular sandwich used for immunoprecipitating T7 phage was characterized with fluorescent antibodies in a clear bottom 384 well plate for validation. In particular, the steps where biotinylated protein A, biotinylated protein G, control antibodies, and human peptidome library T7 phage with FLAG tags were introduced were stained and imaged. Secondary antibodies for each of these molecular layers were screened for non-specific binding in the microfluidic device in the step immediately preceding the step where the molecular layer of interest was added. Additionally, the stains for the control antibodies and T7 phage were validated in parallel using protein A and protein G Dynabeads in parallel. These control antibodies and phage had been validated by others in prior experiments and this parallel approach sought to identify challenges that may be inherent to using a PDMS microfluidic device. The biotinylated BSA and streptavidin were not evaluated due to their successful use in previous MITOMI experiments. The list of conditions and the purpose of each condition is listed in Table 5.1. Chicken antibodies were selected to stain protein A and G due to the fact that they are not reactive.

To prepare the plates, 10 μL of biotinylated BSA (29130, ThermoFisher) was deposited into each well and briefly centrifuged to pull the BSA to the bottom of the well. Afterwards, the plate was incubated at 4 °C for 30 minutes. The wells were then aspirated and washed two times with 25 μL of RIPA buffer before 10 μL of streptavidin (43-4302, ThermoFisher) was dispensed into the wells. The plate was then incubated at 4 °C for 30 minutes. The wells were then washed as previously described before 10 μL of protein A, protein G, or a mixture of the two at a concentration of 2 μM was added to the wells and incubated at 4 °C for 30 minutes. The wells were then washed before rabbit anti-GFAP (Z0334, Agilent) or rabbit anti-T7 (PA1-32386, Invitrogen) antibodies diluted 1:50 in 1x antibody storage buffer were added to the wells and

incubated at 4 °C for 90 minutes. Wells were then washed before 10 μL of T7 human peptidome phage (10^{10} phage/mL) were added to the wells and incubated at 4 °C for 60 minutes. The wells were then washed before fluorescently tagged secondary antibodies for FITC anti-protein A, FITC anti-protein G, FITC anti-rabbit, or Cy5 anti-FLAG diluted 1:50 in 1x antibody storage buffer were added to the wells and incubated at 4 °C for 60 minutes. Wells that did not include a full molecular sandwich were left sitting in RIPA until stained with secondary antibodies. Additionally, wells that used 10 μL of a mixture of protein A Dynabeads (Thermo Fisher Scientific) and protein G Dynabeads (Thermo Fisher Scientific) were prepared separately in PCR tubes where the beads could be pulled down with magnets between washed. The processed magnetic beads were then transferred to the well plate for imaging. Wells were imaged with a Nikon TE microscope for 100 ms with 2x2 binning and processed for median intensities and standard deviations using FIJI.

Following image processing, the signal intensities were then background subtracted and plotted for visualization in Figure 4-5. Both the net difference of non-specific and complex formation signals and the ratiometric change of fluorescence at each step is presented. The change of fluorescence is defined as the net difference of the median fluorescent intensity of the non-specific binding control without the molecular layer of interest and the specific binding with the molecular layer of interest. While there appears to be an observable signal when the protein A, protein G, anti-GFAP antibody, and anti-T7 antibody layers are added, the addition of T7 phage displaying FLAG tagged epitopes did not produce a meaningful signal. This is partly due to the reactivity of rat antibodies with protein A and G. Chicken antibodies targeting FLAG tags were not available and it proved otherwise difficult to confirm the presence of specifically bound phage.

Table 4-1. Surface functionalization screening conditions
Conditions for validating reagents used to create a molecular sandwich for immunoprecipitation were assessed and imaged in a well plate. Dynabeads serve as a positive control that represents traditional benchtop phage immunoprecipitation. Abbreviations: non-specific binding (N.S.), complex formation (C.F.), background signal (B.G.), streptavidin (Strep), protein A (Pro A), protein G (Pro G), protein A and G (Pro AG), anti-GFAP antibody (GFAP), anti-T7 antibody (T7).

Reaction Number	Molecular Sandwich	Secondary antibody	Purpose	Fluorescent Intensity
1	BSA-Strep	Chicken anti-Pro A	N.S.	1169 ±88
2	BSA-Strep	Chicken anti-Pro G	N.S.	1353±98
3	BSA-Strep-Pro A	Chicken anti-Pro A	C.F.	1957±126
4	BSA-Strep-Pro G	Chicken anti-Pro G	C.F.	2711±240
5	BSA-Strep-Pro AG	No secondary (FITC)	B.G.	551±49
6	BSA-Strep-Pro AG	Chicken anti-rabbit IgG	N.S.	2173±225
7	BSA-Strep-Pro AG	Rat anti-FLAG	N.S.	1786±114
8	BSA-Strep-Pro AG-GFAP	Chicken anti-rabbit IgG	C.F.	2963±305
9	BSA-Strep-Pro AG-T7	Chicken anti-rabbit IgG	C.F.	3260±611
10	BSA-Strep-Pro AG-T7-Phage	Rat anti-FLAG	C.F.	1432±100
11	BSA-Strep-Pro AG	No secondary (FITC)	B.G.	742±67
12	BSA-Strep-Pro AG	No secondary (Cy5)	B.G.	579±49
13	Dynabead	Chicken anti-rabbit IgG	N.S.	744±62
14	Dynabead	Rat anti-FLAG	N.S.	1753±130
15	Dynabead-GFAP	No secondary (FITC)	B.G.	1228±75
16	Dynabead-GFAP	No secondary (Cy5)	B.G.	832±69
17	Dynabead-T7	Chicken anti-rabbit IgG	C.F.	10319±651
18	Dynabead-GFAP	Chicken anti-rabbit IgG	C.F.	12722±1497
19	Dynabead-T7	Rat anti-FLAG	N.S.	2032±141
20	Dynabead-T7-Phage	Rat anti-FLAG	C.F.	2071±144

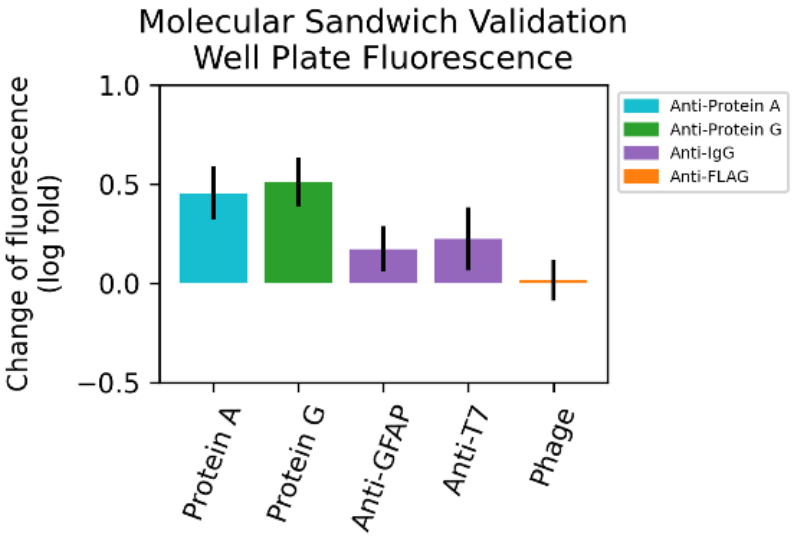


Figure 4-5. Well plate surface functionalization intensities.

4.3.3.3 Characterization of the Immunoprecipitation Surface Functionalization in a PDMS Microfluidic Device

Molecular interactions that happen in a PDMS microfluidic device may differ from those which occur in a well plate due to being in a different chemical environment, so molecular sandwiches were assembled and evaluated in the immunoprecipitation microfluidic device. In short, the surface functionalization procedure was repeated in the device while flowing reagents through the device at 2.5 PSI which corresponded to an effective rate of 2 μL/min. 50 μL of biotinylated BSA, streptavidin, a mix of biotinylated protein A and G, anti-GFAP antibody, and FITC labeled anti-rabbit IgG antibody were serially introduced into the device with 30 minute RIPA washes between steps. The negative control testing for non-specific binding of the secondary antibody replaced the anti-GFAP antibody with 1x antibody storage buffer. The negative control testing for non-specific binding of the primary antibody omitted the step where biotinylated protein A and G mixture was introduced into the device. Devices were imaged with a Nikon TE microscope for 1000 ms in the Cy5 channel with 2x2 binning and qualitatively compared.

The images presented in Figure 4-6 demonstrate that the molecular sandwich for immunoprecipitating phage was formed successfully in the microfluidic device. Conditions for assessing non-specific binding of the immunoprecipitating antibody and the reporter indicate that minimal background activity is present. In addition, when the molecular sandwich is formed and stained with a secondary antibody, the device channels appear to fluoresce which indicated successful functionalization. While the integrity of the molecular sandwich could be validated up to the immunoprecipitating antibody layer, T7 phage could not be visualized due to reagent limitations that were previously discussed in the well plate validation studies.

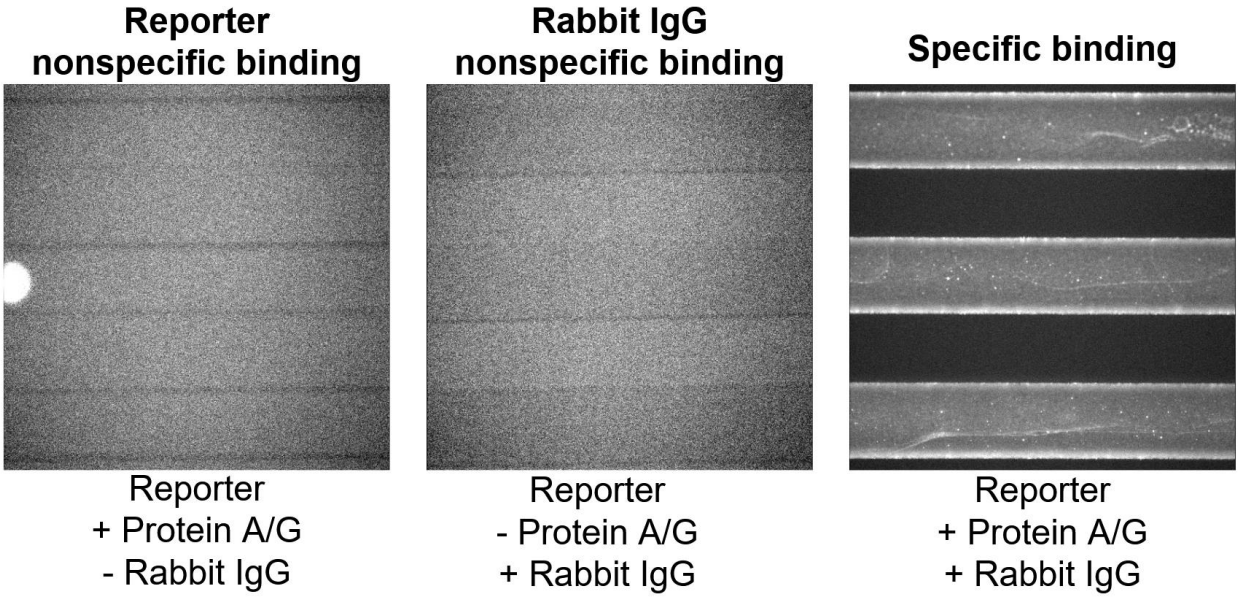


Figure 4-6. Characterization of device surface functionalization.

4.3.3.3 Characterization of the Capture Efficiency of Immunoprecipitation Antibodies

When applying PhIP-seq to identify autoantibodies in patients, the most limiting resource is the patient sera as it is typically drawn during a brief period when an individual presents to a clinic with symptoms. Additional samples are often not available. To maximize the efficiency of the sample being used, an assay that quantified the presence of antibodies in the microfluidic device flowthrough was developed. Briefly, the surface of the device was prepared to the protein A and G mixture had been deposited on the surface. Afterwards, different devices flowed 3.2 mg/mL GFAP antibody diluted 1:50 in 1x antibody storage buffer into the device at input pressures of 1, 3, and 5 PSI. The 20 μL of flow through was then incubated with 5 μL of a mixture of protein A and G Dynabeads that were prepared according to the standard PhIP-seq benchtop protocol[142]. The samples were incubated on a rotator for 1 hour at 4 °C. Afterwards, the samples were washed with RIPA buffer 3 times before being incubated with chicken anti-rabbit IgG diluted 1:50 in 1x storage buffer for 30 minutes at room temperature in a dark environment. The positive control consisted of 40 μL of GFAP diluted 1:50 in antibody storage buffer that was further diluted with 8

μL of RIPA to mimic on-device dilution of the sample. The negative control consisted of 40 μL of 1x antibody storage buffer diluted with 8 μL of RIPA.

Following incubation, the samples were washed 3 times with RIPA buffer and imaged on a Nikon microscope in a clear bottom well plate. Using FIJI, an ROI the size of 25% of the image frame was placed where the samples had uniform intensity and the median and standard deviation was computed. Figure 4-7 plots the fraction of antibodies captured by the device as the pressure varied. The captured fraction was computed by normalizing the signal from the uncaptured antibodies in the flow through relative to an unprocessed input sample. At an input pressure of 1 PSI, roughly 65% of the antibodies were not pulled down onto the protein A and G molecular sandwich deposited on the surface of the device.

The immunoprecipitation channel is designed to have a surface area that theoretically supports approximately 20 μg of antibodies if they were densely packed and limited by steric hinderance, which is equivalent to 2 μL of human sera at average concentrations. The 30% capture rate of the flow through at 1 PSI indicates that 1 μg of GFAP antibody was captured, which corresponds to a 5% occupancy ratio on the surface of the device. This may be partly attributable to being dependent on the integrity of the molecular sandwich for capturing the antibodies and the molecular dynamics of the antibody as it passes through the device.

The molecular sandwich consists of the sequential layering of biotinylated BSA, streptavidin, and biotinylated protein A and G. The device passivation is dependent on a stochastic process of BSA colliding with the surface of the channel and being oriented in a way that presents the biotinylated sites so that they are accessible. While BSA has a small hydrodynamic radius around 4 nm,[143] it will not fully pack and occupy the entire surface as it bounces off of neighboring molecules and also gradually loses effective concentration as molecules bind to the surface while

it flows downstream. Similarly, streptavidin depends on being able to access the biotinylated sites of BSA while being limited by similar mechanics. This effect compounds with the biotinylated protein A and G and the subsequent antibody. While not shown, initial fluorescent images of the functionalized device surface showed an intensity gradient that was stronger at the inlet of the immunoprecipitation channel and weaker at the outlet indicating that the surface of the channel is not effectively saturated under the conditions used and may be a significant factor for why a 5% occupancy rate had been observed.

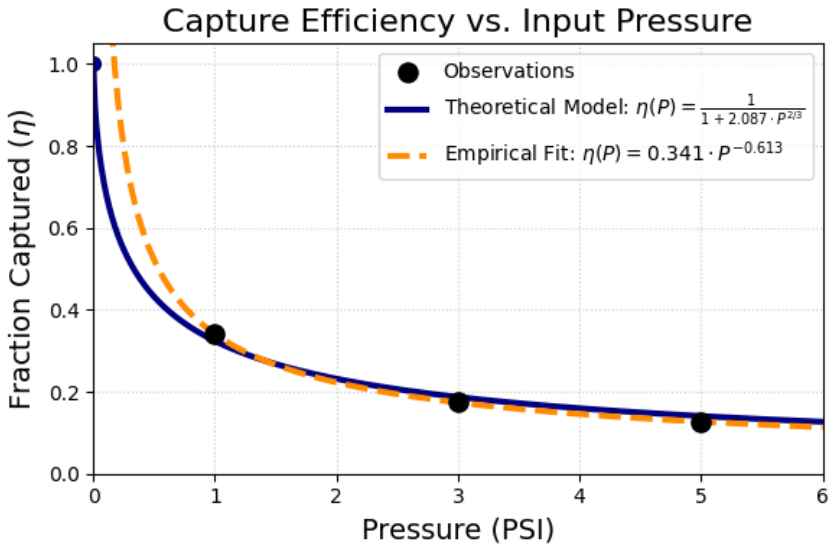


Figure 4-7. Antibody capture efficiency.
A monoclonal antibody was passed through functionalized devices at various pressures and the fraction of captured antibody was computed by measuring the relative concentration in the flow-through. A power law fit and parameters for a theoretical mass-transport limited model were computed using experimental observations.

The antibody capture efficiency was also assessed by repeating the experiment but creating a negative control for each pressure condition by flowing antibody through a device that had only been passivated with biotinylated BSA and measuring the changes of intensity with a Western blot. The negative controls and samples each had a band intensity difference of roughly 3% between all conditions. The approach used here is not further documented as other approaches may be more effective for quantification.

4.3.3.3 Surface chemistry optimizations

Following the assessment that the immunoprecipitation channel had not been uniformly functionalized, additional studies were performed to identify limiting factors. In short, multiple devices were functionalized to evaluate the effect of doubling the volume of streptavidin, the mixture of biotinylated protein A and G, and an anti-GFAP control antibody. The input pressure of the reagents was set to 2.5 PSI and the reagents were run until fully depleted. The functionalized surface was stained with a 4:50 dilution of chicken anti-rabbit IgG to ensure that excess stain was available to saturate the captured antibodies.

Afterwards, the full devices were imaged, stitched, and an intensity line profile was taken to measure the change of channel intensity of the immunoprecipitation channel as it progressed through the device. The line profile was selected to be in a position that minimized the impact of image vignetting and device defects. The intensity data of the line profile was then converted into a histogram and fit with a bimodal distribution to identify the intensity offset between the device background and the channel fluorescence. Figure 4-8 demonstrates the process and Table 4-2 presents the quantitative results. To facilitate comparison across devices, the intensities were normalized such that the background had an intensity of 1.

Table 4-2. Screen of reagent volumes for optimizing channel functionalization.
Different surface functionalization conditions were evaluated to identify limiting reagents. The mean intensities of the device background and the immunoprecipitation channel were computed and the delta between the means was measured. To assist with inter-device comparisons, the delta was normalized to the background mean intensity. Abbreviations: S – streptavidin, A/G – mixture of protein A and G, Ab – antibody.

Condition	Background	Channel	Δ Peaks	Δ Normalized
Standard Device	617.6±1.2	782.2±3.0	164.6±3.2	0.267±0.005
2x S	634.4±1.0	941.7±5.0	307.3±5.1	0.484±0.008
2x A/G	619.3±1.1	840.4±3.8	221.1±3.95	0.357±0.006
2x S and A/G	1947±3	3384±18	1437±18.2	0.738±0.009
2x S, A/G, and Ab	1879±3	3088±17	1209±17.3	0.643±0.009

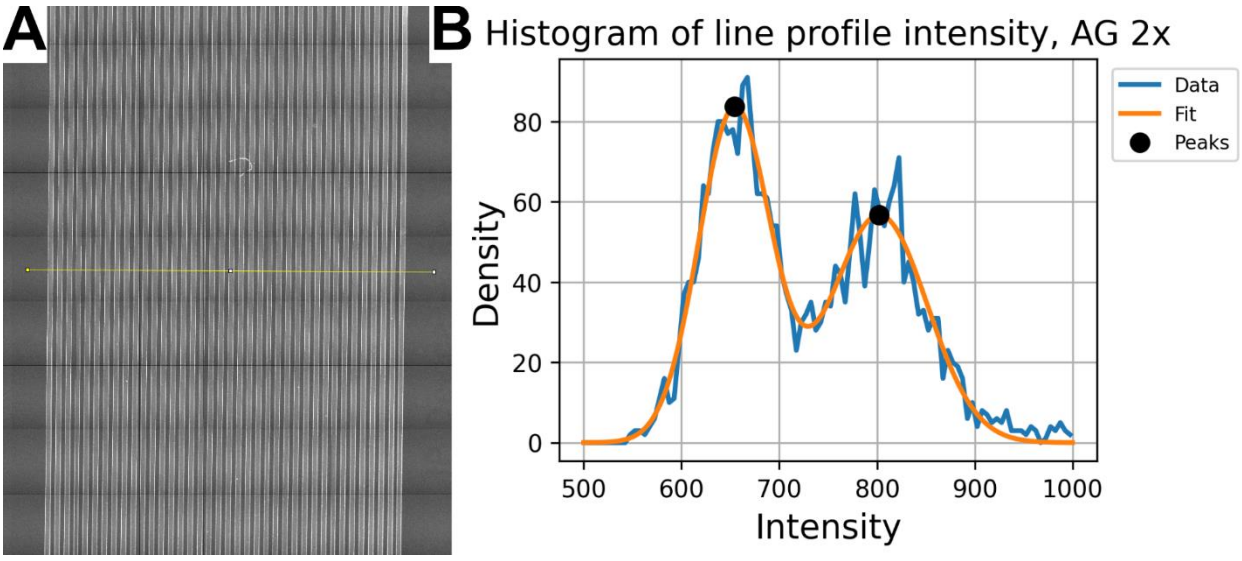


Figure 4-8. Process for assessing uniformity of channel functionalization.
(A) A line profile of the microfluidic device is taken to measure the intensities across the device. (B) A histogram of the intensities is made and fitted with a bimodal distribution to find the distance between the mean channel and background intensities.

Data collected from the assays indicate that the devices had capacity for additional streptavidin and protein A and G. The normalized difference of intensities between the background and immunoprecipitation channel were all consistently higher when additional reagents were used. While additional protein A and G alone increased the channel signal by roughly 30%, all conditions that used additional streptavidin had an increase of signal of 80% at a minimum, which increased with additional protein A and G. Conditions that effectively report a 200% increase of fluorescence roughly correspond to a capture of 3 μg of antibody and a 15% surface occupation ratio. While the trend from the data clearly indicates the benefit of additional reagents, challenges with the stability of the image acquisition software, challenges with the device design revision that was used in the experiments, and backflow of residual reagents from the disconnected ports on the devices did impact some of the channel intensities.

4.3.3.3 Phage washing and elution optimizations

While optimizations for surface functionalization were necessary for maximizing sample capture, identification of washing conditions for minimizing non-specific binding of T7 phage to the device and characterization of elution timings for sample recovery were also required. To assess optimal washing conditions, a device was prepared without the addition of a capture antibody. Afterwards, 50 μL of human peptidome library was introduced into the device before being washed with RIPA buffer for 60 minutes. During the wash, the flow through was collected into a new container every 5 minutes to discretize the output concentration of phage over time. The samples were then amplified via qPCR with primers targeting the FLAG and strep tags of the inserted epitope of the peptidome library using KAPA HiFi HotStart Real-Time Library Amp Kit (Roche) and a LightCycler qPCR machine (Roche) to assess the relative concentration of phage in the samples.

To compute the relative concentration, an arbitrary threshold of 0.2 was set and the cycle number at which each sample passed the threshold was linearly interpolated. The lowest cycle count was set to be a relative concentration of 1 and the concentration of all other samples was computed by considering the relationship between exponential amplification and cycle number of PCR. Plots for the qPCR curves and the relative concentration of phage in the wash flow through is provided in Figure 4-9.

Over the course of the 60 minute wash, the nonspecifically bound phage can be seen decaying with diminishing effectiveness with a plateau around the 40 minutes into the wash. 40 minutes of washing corresponds to roughly 8 device volumes with a flow rate of 2 μL per minute. The relative concentration at that time point was found to be 10^{-4} times lower than the initial concentration, or effectively 10^{6} phage per mL or 10^{3} phage per μL given a stock concentration of

10^{10} phage per mL The geometry of the device revision used in the experiment may have contributed to the background signal as the valve junctions of the input reagents had notable dead volume following the valve. This dead volume may have led to the gradual diffusion of concentrated phage stock into the wash buffer. This dead volume concern was addressed in Revision F of the device design, but the washing and elution conditions were not revisited.

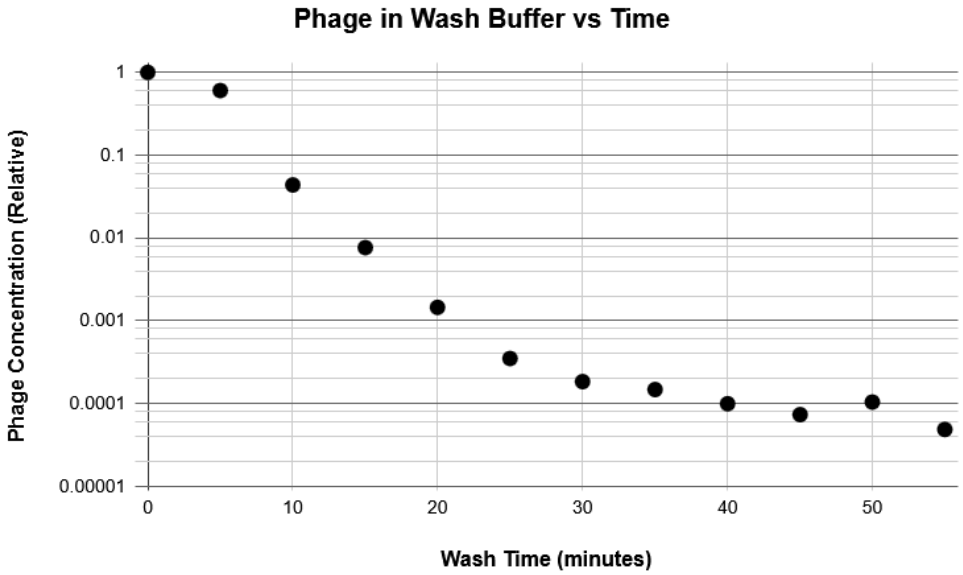


Figure 4-9. Concentration of T7 phage in RIPA wash flow through.

A similar process was used to characterize the elution conditions. A device was prepared using, anti-T7 antibodies and human peptidome before being washed for 60 minutes and eluting with Gentle Elution Buffer for 30 minutes. The wash flow through was collected in intervals of 10 minutes and the eluant was collected in intervals of 5 minutes. The fractions were then diluted ten-fold and concentration was measured with qPCR as previously outlined. The ten-fold dilution was used to prevent the salts of the elution buffer from interfering with the polymerase in the PCR reaction. While the wash buffer presented with a similar curve plateauing at 40 minutes, the magnitude differed significantly with a flattening of the curve at a relative concentration of 10^{-2}. The elution presented with a similar curve and magnitude as the wash, which made it difficult to

interpret. The elution fractions were expected to have an increase of phage concentration as the elution buffer disrupted interactions between phage and antibody, but that behavior was not evident. The first fraction at 5 minutes was expected to consist mostly of RIPA buffer and yet it had the highest signal of the fractions.

4.3.3.4 Microfluidic enrichment of GFAP fragment 11 with PhIP-seq

While a faint signal may have been present in the fractions, a PhIP-seq experiment with serial rounds of amplification was expected to be more capable of generating a definitive signal. Microfluidic devices were prepared with anti-GFAP antibodies and a modified phage library composed of roughly 70% of GFAP fragment 11 was used. The anti-GFAP antibody had been previously characterized to show binding to fragment 11. After a 60 minute RIPA wash, eluate was flowed into the device for 10 minutes, incubated for 10 minutes, and then collected for 20 minutes. 25 µL of the eluate was used then amplified in 500 µL of E. coli at OD600 of 0.387 for 2 hours at 37 °C before the lysis reaction was stopped with 50 µL of 5M NaCl and stored at 4 °C overnight. The lysate was then centrifuged at 5000 RPM for 30 minutes and the supernatant was used as the phage source in a new device. This was repeated for a total of 3 amplifications. In the third round of amplification, the E coli was slow to lyse, so the lysate was stored at 4 °C overnight and continued the following day.

The lysate was then cleaned and prepared for sequencing as described in the benchtop phage protocol[142]. After preparation, the concentration was measured on an Agilent TapeStation with a DNA high sensitivity kit and found to have a concentration of 253 nM while a typical concentration is often around 350 nM. The purified library was sequenced with a MiSeq i100 5M 300 cycle assay cartridge (Illumina) with a 15% PhiX control spike-in. The input library, initial

flow through of the input library, and all three rounds of amplification were sequenced and details regarding each sample are listed in Table 4-3.

Table 4-3. PhIP-seq results of GFAP Fragment 11 from on-chip immunoprecipitation.
Devices with anti-GFAP control antibodies were used in a 3 round PhIP-seq experiment using a modified peptidome library. Samples included the input library, the library flow through from the initial device, and the phage recovered from E. coli lysate. Abbreviations: Diversity – the ratio of unique peptides to total reads, F11 – T7 phage displaying fragment 11 of the GFAP epitope, % TR – percent of total reads GFAP F11 occupies.

Sample	Total Reads	Unique Peptides	Diversity	GFAP F11 Count	% TR
Input Library	135495	34400	0.254	85939	63.43
Library Flow Through	265236	51115	0.193	185519	69.94
Amplification 1	106516	59877	0.562	2261	2.12
Amplification 2	336855	96420	0.286	4043	1.2
Amplification 3	241969	42712	0.177	3157	1.3

Sequencing results indicated that the assay had relatively poor performance and that adjustments to the implementation should be made. Of the 5M reads available on the cartridge, only 1M were used in collecting data from the samples. In addition, the samples were not uniformly represented and could have been better balanced prior to pooling. While the input library was measured to have roughly 60% of its reads attributed to GFAP fragment 11, the results of amplification 1 show that the percent of total reads drops to 2%. Similarly, while there is a 30% difference in the number of total reads between the two samples, amplification 1 had more than a 100% increase in the number of unique peptides sequenced. These results indicate that GFAP fragment 11 was severely de-enriched within the microfluidic assay and that it was possible that the elution process was not effective at releasing the phage from the antibodies.

If one assumes that GFAP failed to bind to the immunoprecipitation antibodies and the phage signal represents nonspecific binding within the device, one could expect the results of the

initial amplification to be similar to that of the input with a moderately high presence of GFAP fragment 11. It is possible that GFAP fragment 11 has biochemical characteristics that reduce its capacity to bind nonspecifically to the device surface and a majority of the reads observed in amplification 1 are a result of the binding capacity of the immunoprecipitation channel. To validate this hypothetical, a negative control should have been run in parallel using a device that did not immunoprecipitated GFAP fragment 11.

Similarly, a standard peptidome library may have been more advantageous in this experiment. Successful enrichment of epitopes in PhIP-seq experiments is typically denoted as a significant increase in the number of reads of an epitope with respect to the input library or negative controls. The phage library used had little room for exponential growth of GFAP fragment 11 reads over several rounds of amplification due to the high starting concentration. A characteristic of serial enrichment and amplification is the continual decrease in sample diversity as nonspecific interactions are out amplified by specific binders. This can be observed in the data presented in Table 4-3 and may indicate that the passivated PDMS environment may passively enrich for a subset of epitopes.

4.3.3.5 Future directions of the microfluidic assay

The work presented sought to develop a microfluidic assay for automating the process of enriching phage from a complex peptidome library used in PhIP-seq experiments. Over the course of the work, multiple design iterations were explored to improve the robustness of the device while also minimizing properties such as the trapping of concentrated phage library in post-valve dead volumes. In addition, the chemistry of the molecular sandwich for capturing phage was validated, optimization conditions for improving immunoprecipitation yields were explored, and characterization of washing conditions was performed.

While an initial PhIP-seq experiment with three rounds of amplification and on-chip enrichment did not yield the results needed to validate the assay, it was informative for future studies. Using a library that does not have epitope fragments of interest over represented provides room to observe exponential enrichment of immunoprecipitated phage over several rounds of amplification. Similarly, a negative control that does not immunoprecipitate specific epitopes would be useful for characterizing which phage clones have a natural affinity for the passivated PDMS microenvironment. This microenvironment signature is also present in the benchtop PhIP-seq protocol where certain epitope fragments naturally bind to the protein A and G beads used in the assay. Following validation with control antibodies, the next step would be to validate the platform using complex sera that has been previously characterized with the standard PhIP-seq protocol. The microfluidic approach remains to be promising approach for automating enrichment of epitopes PhIP-seq that requires further validation studies before it can be confidently used as an automated immunoprecipitation platform.

4.3.4 Summary of continuous flow approaches for phage immunoprecipitation

PhIP-seq is a time intensive and laborious method that is effective at identifying potential antigens that may be targeted in autoimmune diseases. Several methods were explored as means to automate and standardize the immunoprecipitation and washing process of the protocol. Of those discussed here, both the macrofluidic magnet array and the microfluidic immunoprecipitation device remain as potentially viable strategies. The macrofluidic magnet array was pivoted away from while in early development, but it could have used further engineering and protocol modifications before advancing on to biochemical validation studies. The microfluidic platform was the most extensively characterized approach that required multiple iterations of device design and various surface chemistry validation and optimization studies. An initial attempt

of enriching and amplifying phage in the PhIP-seq protocol was not fruitful, but adaptations to the validation study may prove to be effective. Following validation, the next step following control antibody studies would be to compare assay results to the benchtop method with complex sera. While a method for automating immunoprecipitation and washing of phage in the PhIP-seq protocol was not seen to completion, there continues to be potentially viable strategies.

4.4 Exploration of rapid diagnostics

While exploratory assays such as PhIP-seq play a critical role in the identification of relevant disease markers, the depth of screening comes with a trade-off of assay speed and labor. This behavior can be propagated to many large-scale multiplexed assays including microarray and sequencing technologies. While these methods are powerful, they cannot be effectively leveraged in cases where rapid confirmatory assays are desirable. The following sections include experimental work that was performed to evaluate the feasibility of alternative methods for detecting autoantibodies using Luminex beads and the development of a CRISPR-Cas assay for the detection of specific nucleic acid sequences.

4.4.1 Assessment of Luminex beads as a diagnostic for autoimmune epitopes validated with PhIP-seq

Many autoimmune diseases may present with similar symptoms, but effective responses for treating their condition are often dependent on understanding the etiology of their disease. Turn-around time is also particularly important for autoimmune diagnostics as active flares may cause irreversible damage if the disease cannot be correctly managed. While the PhIP-seq protocol is an effective method for conducting an unbiased screen of more than 500,000 potentially reactive autoantigens, it also takes approximately 5 days to complete and return clinically actionable data.

In contrast, Luminex bead assays can complete an equivalent screen of sera against a select panel of epitopes in several hours. The following work examines the feasibility of translating known autoantigens discovered through PhIP-seq into a Luminex bead assay as a way to potentially lay the foundation for a process to convert numerous PhIP-seq research findings into a clinically actionable diagnostic.

To prepare the assay, Luminex MagPlex magnetic microspheres were functionalized using BSA conjugated peptides. 30-mer peptide fragments were collected from epitopes associated various autoimmune diseases characterized via PhIP-seq, modified to have an N-terminal cysteine, and synthesized with a cysteine-conjugated BSA (Lifetein) as listed in Table 4-4. Many of the epitopes were identified using data collected from published and unpublished Phage-Assisted Scanning Epitope Recovery (PhASER) datasets[144]. The BSA served to improve the solubility of the peptides and facilitate coupling to MagPlex beads. Peptides were reconstituted according to the manufacturer's instructions. The peptides were conjugated to the Luminex MagPlex beads following the manufacturer's xMAP carbodiimide coupling protocol using their antibody coupling kit such that each peptide was represented by a single set of color-coded beads at a ratio of 6µg total protein per million beads. A control set of beads containing BSA alone was also prepared.

Immediately prior to the experiment, bead populations were mixed at 50,000 per bead population/mL in PBS with 0.05% Tween-20 containing 1% non-fat milk. Sera samples were diluted 1:1 in sample buffer if not already prepared, then diluted to 1:250 in PBST with 1% non-fat milk before 50 µL of the buffered samples was mixed 1:1 with the prepared beads. The sera used included a set of monoclonal antibodies targeting ZSCAN1 generated from sera of a patient diagnosed with ROHHAD[144] diluted to 1 mg/mL in healthy sera, a set of sera from patients diagnosed with systemic lupus erythematosus, serum from a patient diagnosed with APS1, and

serum from a healthy control. In addition, to assess the limit of detection of the platform, a dilution series of the generated ZSCAN1 monoclonal antibodies was prepared ranging from 10^{-4} to 10^{-11} mg/mL in both sample buffer and healthy sera. Samples were incubated at room temperature for 1 hour on a shaker at 1300 RPM, washed 3 times with a Tecan Hydrospeed 96-well magnetic plate washer with PBST. Samples were then stained with 1:2,000 phycoerythrin-conjugated anti-human IgG Fc antibody (BioLegend) in PBST for 1 hour at room temperature before being washed an additional 3 times with PBST. The fluorescent intensity of the samples was then measured on a Luminex LX 200. The mean fluorescence intensity for each peptide was standardized by dividing the MFI of the peptide by the MFI of the BSA control. Results for the single point measurements are presented in Figure 4-10 and the dilution series data is presented in Figure 4-11.

Table 4-4. List of epitopes used in the Luminex xMAP assay.
Peptide fragments from various autoimmune diseases that were validated with PhIP-seq on a human peptidome library were selected in a pilot Luminex diagnostic assay. The diseases included rapid-onset obesity with hypothalamic dysfunction, hypoventilation, and autonomic dysregulation (ROHHAD)[68], spontaneous preterm birth (Preterm)[145], autoimmune polyglandular syndrome type 1 (APS1)[67], and systemic lupus erythematosus (SLE). The native epitope for RFX6 included an internal cysteine residue, so two variants were produced to assess the effect of a non-terminal BSA conjugation.

Protein	Fragment	Association	Peptide Sequence
ZSCAN1	1	ROHHAD	CRGTVAREMLPRPKAPASPRRPQTPTPSEQ
ZSCAN1	2	ROHHAD	CSPRRPQTPTPSEQDADPGPASPRDTEAQR
ZSCAN1	8/9	ROHHAD	CTHFIEHQKTHREEGPFPSPESGKVFLHNS
ZSCAN1	13	ROHHAD	CRGGRPFQCADGMVFTWVTHFIEHQKTHRE
ZSCAN1	15	ROHHAD	CLHNSVLTEHGKIHLLEPPRKKAPRSKGPR
ZSCAN1	17	ROHHAD	CPKRPFQCSVCGKAFPWMVHLIDHQKLHTA
IL1RA	4	Preterm	CVKSGDETRLQLEAVNITDLSENRKQDKRF
RFX6	33/34	APS1	CGRKQTSSFYTDTSSPVACRTPVLASSLQT
RFX6	33/34	APS1	CGRKQTSSFYTDTSSPVASRTPVLASSLQT
PLIN1	-	APS1	CSVAMQAVSRRRSEVRVPWLHSLAAAQEED
IL17F	3	APS1	CIESRSTSPWNYTVTWDPNRYPSEVVQAQS
YTHDF2	0	SLE	CKDGLNDDDFEPYLSPQARPNNAYTAMSDS

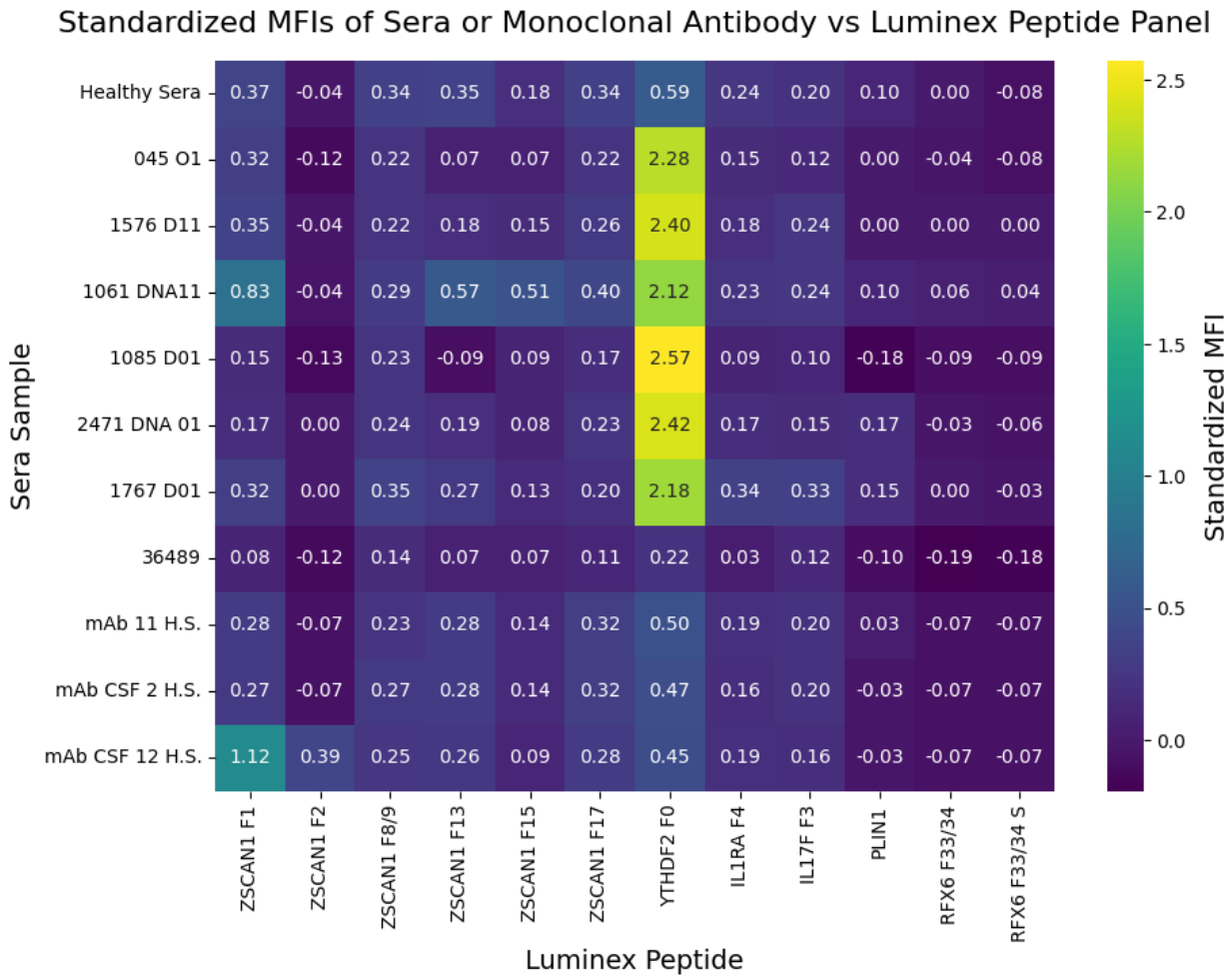


Figure 4-10. Single point serum measurements on a Luminex peptide panel.

The datasets from the single point measurement and dilution series assays indicate a mixed success. In the prepared assays, monoclonal antibody derived from sera sample 11 was expected to bind to an unrepresented ZSCAN1 fragment, the monoclonal antibody derived from CSF sample 2 was expected to bind to ZSCAN1 fragment 13, and the monoclonal antibody derived from CSF sample 12 was expected to bind to ZSCAN1 fragment 1/2. The SLE samples were expected to react with YTHDF2, and the APS1 sample was expected to react with any of the IL17F, PLIN1, and RFX6 epitopes. Figure 4-10 indicates that the SLE sera generated strong signals against the YTHDF2 epitope, but only other than the appreciable signal was generated by the monoclonal antibody derived from CSF sample 12. This may be partly due to observed solubility challenges

for ZSCAN1 fragment 13. The APS1 sera did not appear to be reactive with any of the APS1 epitopes present in the assay.

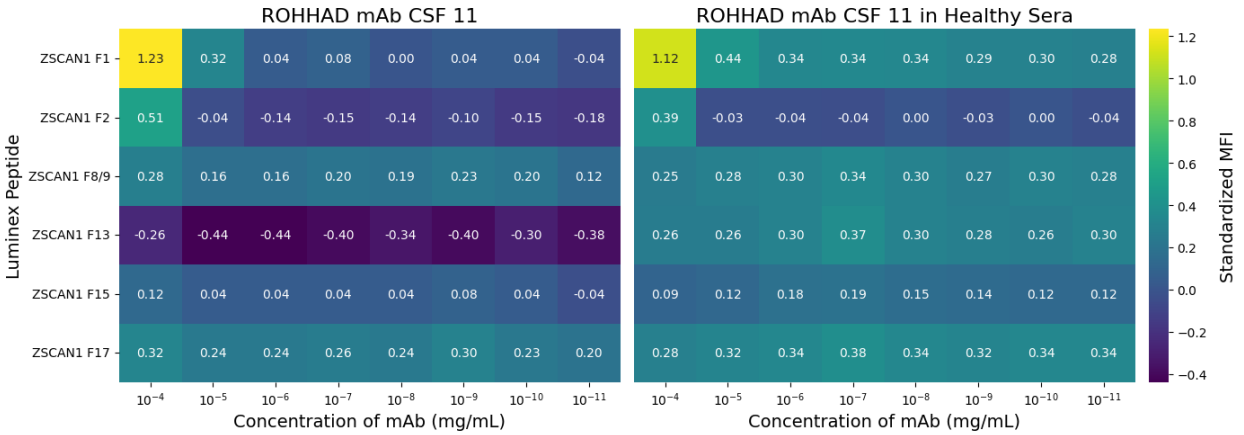


Figure 4-11. Dilution series measurements of a ROHHAD mAb on a Luminex panel.

The dilution series data with the ROHHAD derived monoclonal antibodies also indicate mixed results. The derived monoclonal antibodies were diluted in sample buffer and healthy sera to simulate samples with both clean and typical background signals. The monoclonal from sera sample 11 behaved as expected and did not bind to any of the epitopes present in the assay. The monoclonal from CSF sample 2 did not bind to ZSCAN1 fragment 13, which is likely due to insufficient solubilization of the peptide during sample preparation as observed while conjugating peptides to the Luminex beads. The monoclonal from CSF sample 12 did effectively bind to ZSCAN1 fragment 1 and fragment 2, but the signals are undetectable beyond below a concentration of 10 ng/mL in sample buffer and 100 ng/mL in healthy sera.

While sera is known to inhibit the binding on Luminex beads and increase the background noise,[146] a limit of detection in the low pg/mL range is still achievable[147] in some assays using the Luminex xMAP platform. Additionally, while the concentration of specific autoantibodies can widely vary between patients and diseases[148], concentrations ranging from 10^{-4} to 10^{-1} mg/mL are routinely observed[149,150]. Even though the sensitivity of the dilution series data does not reach the

limit of detection described by others, it may still be possible to leverage the Luminex assay as a rapid diagnostic for autoimmune diseases that have been previously characterized with PhIP-seq given that the physiological concentration of autoantibodies is often notably higher than the limit of detection. However, the diverse biochemical nature of peptide fragments will require optimizations to ensure that all samples can be solubilized in the reconstitution buffer and that the buffer does not interfere with the xMAP carbodiimide conjugation reaction.

Similarly, the strong signals associated with the SLE sera indicate that the synthetic conditions explored with the monoclonal ZSCAN1 antibodies do not necessarily reflect physiological conditions and that naturally complex samples can successfully leverage the platform. While the APS1 sample was expected to bind to PLIN1 at a minimum, it is possible that the peptides similarly failed to solubilize, that the epitope was in a position that is difficult to access, or that the sera has lost some reactivity over time.

The position of the epitope on the peptide fragment does appear to have a non-trivial effect. The RRPQTPTP epitope targeted by the monoclonal derived from CSF sample 12 is present in both ZSCAN1 fragment 1 and fragment 2, but the epitope is closer to the N-terminal of fragment 2. Given that the signal for fragment 2 has a magnitude that is less than half of the signal for fragment 1, there is the possibility that the lower signal may be attributed to accessibility. Further investigation into this could be worthwhile to determine if the observed effect is related to the position of the epitope or the biochemical composition of the peptide.

In summary, epitopes that were discovered to have reactivity with autoantibodies with a human peptidome PhIP-seq assay were translated onto a Luminex xMAP assay to assess the feasibility of leveraging the Luminex technology as a rapid diagnostic for autoimmune diseases. The mixed results indicate that the platform may be viable with additional optimizations. Complex

sera from SLE patients had a strong signal while serum from a patient with APS1 did not appear to be reactive. This discrepancy may be partly attributed to the reconstitution of the peptide fragments as it was observed that at least one peptide from the ZSCAN1 set failed to solubilize. Additionally, in assessing the sensitivity of the assay with monoclonal antibodies that were diluted in clean sample buffer and a complex, healthy control sera, it was observed that signal was difficult to detect below 10 ng/mL. While the Luminex platform is capable of reaching lower limits of detection, this may not be physiologically relevant as many autoimmune diseases have autoantibodies that are present at a concentration ranging from hundreds of ng/mL to hundreds of μg/mL. Continued work on this platform would help provide a clearer assessment of the limits and nuances of implementing a Luminex-based autoimmune diagnostic.

4.4.2 Exploration of CRISPR arrays for detecting infectious diseases

CRISPR-Cas endonucleases have continued to form the basis of new nucleic acid detection systems since the 2010s. The work explored in this section sought to enhance the multiplexing capacity of the technology by transitioning to spatially encoded multiplexing strategies with microarrays or microfluidic chambers rather than relying on low density spectral encoding strategies that had been previously published. One such strategy is detailed in Figure 4-12, where CRISPR-Cas enzymes are functionalized with a flexible linker, activated with a crRNA guide, and microarrayed onto a planar surface. Once they detected complementary nucleic acids in the sample, the catalytic unit of the endonuclease would become active and start cleaving neighboring quencher-fluorophore pairs. The work described below begins with the synthesis of SARS-CoV-2 N gene RNA and a partial replication of a study conducted by Fozouni et al.[85] to confirm reagent activity. Afterwards, evaluation of functionalized CRISPR-Cas endonucleases and alternative pivots are detailed.

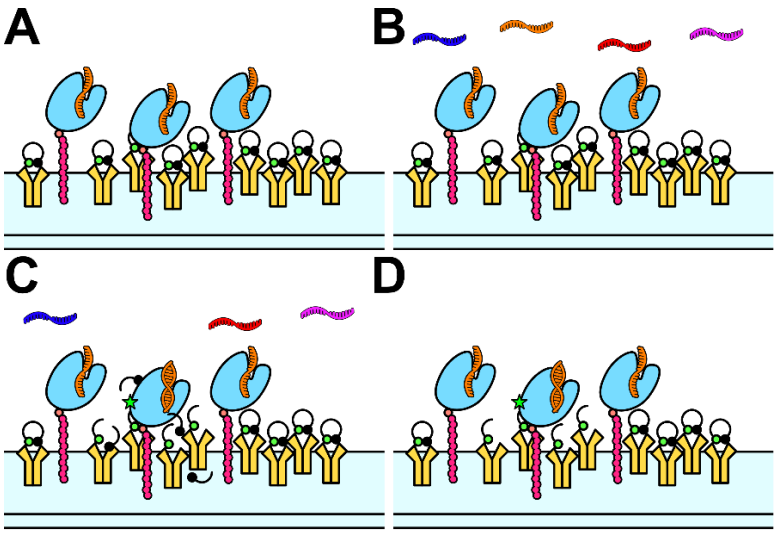


Figure 4-12. Strategy for a spatially multiplexed CRISPR-Cas assay.
(A) Cas RNP complexes are spotted onto a solid substrate where the flexible linker binds to the surface. The surface is also patterned with antibodies that keep fluorophore-quencher pairs in close proximity. (B) Sample is introduced into the assay. (C) Hybridization to the RNP spot causes local degradation of the fluorophore quencher pair. (D) Intensities can be measured following a wash.

4.4.2.1 Isolation and expression of SARS-CoV-2 N gene RNA

SARS-CoV-2 N gene was partially reverse transcribed from several total nucleic acid extracts from SARS-CoV-2 positive patient samples using the LunaScript RT Master Mix (NEB) according to the manufacturer's protocol with NIID_2019-nCOV_N_R2 and a custom primer targeting the first 1000 base pairs of the gene. Afterwards, 2 µL of the reverse transcription reaction was PCR amplified with CloneAmp HiFi PCR Premix (Takara Bio) without adding additional primers according to the manufacturer's protocol. 10 µL of the PCR product was then run on a 2% agarose gel before samples with a band around 1 kb were extracted and cleaned with a QiaQuick Gel Extraction kit (Qiagen). Cleaned products were then reamplified with CloneAmp using the same primers that were used in the reverse transcription reaction. PCR reactions were then cleaned with a DNA Clean and Concentrator kit (Zymo) and subsequently measured for concentration with a Nanodrop Microvolume Spectrophotometer (Thermo Fisher Scientific).

A T7 expression promoter sequence was then added to the amplified and purified SARS-CoV-2 N gene DNA with an additional round of CloneAmp PCR with a custom T7 promoter primer and NIID_2019-nCOV_N_R2. The product was then run on a gel, extracted and purified, reamplified with the T7 primer, and measured for concentration as previously described.

The PCR product was then cloned into a plasmid provided with the Zero Blunt TOPO kit (Thermo Fisher Scientific) according to the manufacturer's instruction before transforming Stellar competent cells (Takara Bio) according to the supplied protocol. Following a 24 hour incubation, 10 colonies were picked ang growing in 3 mL of LB with carbenicillin overnight before being cleaned with a QIAprep Spin Miniprep kit (Qiagen) and sequenced. Due to a conflicting T7 promoter on the plasmid, the PCR amplification and clean up previously described was repeated on samples that had a confirmed SARS-CoV-2 N gene insert. The 300ng of the cleaned product was T7 expressed while shaking at 300 RPM overnight at 37 °C with a custom T7 reaction buffer to yield a total of 50 µg of N gene RNA from 3 samples as measured by a Qubit 3 Fluorometer (Thermo Fisher Scientific).

4.4.2.2 Confirmation of CRISPR-Cas endonuclease assay reactivity

Fozouni et al. had previously developed a sensitive LbuCas13a based fluorescent detection platform for detecting SARS-CoV-2 RNA. [85] In their work, they characterized the performance of 12 unique CRISPR RNAs (crRNA) that specifically targeted different 20-nucleotide regions of the viral N gene. The performance of these crRNA is shown in Figure 4-13. Prior to developing a unique LbuCas13a assay, a subset of the crRNAs published in their study were selected to validate baseline assay activity of the reagents being used. The crRNAs included crRNA 1, 2, 4, and 7 to form a basis of weak and strong performing guide RNAs. In addition, a nontargeting crRNA (NT) and a water control without RNA (crH2O) were included as alternative negative controls.

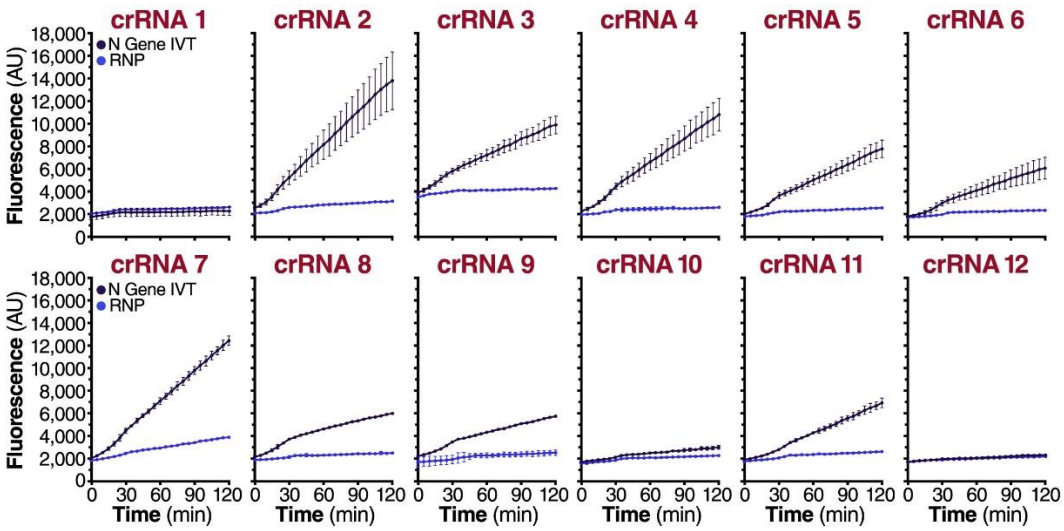


Figure 4-13. Performance of various crRNAs against SARS-CoV-2 N gene RNA.
This figure is reproduced with permission and sourced from Fozouni et al.[85]

Synthesizing the crRNAs involved ordering reverse complement ssDNA of the crRNA sequence with a T7 promotor (IDT), hybridizing ssDNA primer consisting of the T7 promotor to the template, and completing an overnight T7 expression reaction using materials and methods previously described. Ribonucleoprotein (RNP) complexes were formed using the same methods as published by Fozouni et al. so that the final concentration consisted of 50 nM of RNP in the presence of 167 nM of RNAse Alert substrate (IDT) and 480 fM of SARS-CoV-2 N gene RNA. The reactions were dispensed into a clear bottom 384 well plate and fluorescent activity was measured by a SpectraMax M3 microplate reader at 37 °C for 120 minutes with readings being taken every 5 minutes. The resulting data is shown in Figure 4-14.

While the performance of the replication experiment was inferior to that of the published assay, the behavior of the crRNAs was mostly aligned. Both crRNA 2 and 4 exhibited signal that was strongly differentiable from the non-reactive crRNAs controls 1, NT, and H2O. The crRNA 7 did not appear to be reactive in the replicated assay whereas it had a strongly differentiable signal in the published study. Similarly, while the published study showed signals that reached an

intensity that was 8 times greater than their initial intensity, the replicated experiment demonstrated a signal enhancement of close to 2 times the original. It is possible that some of the observed differences may be due to insufficient mixing of reagents or utilizing concentrations that were lower than expected. Regardless, some of the positive and negative controls replicated the desired behavior, which was sufficient for functional validation.

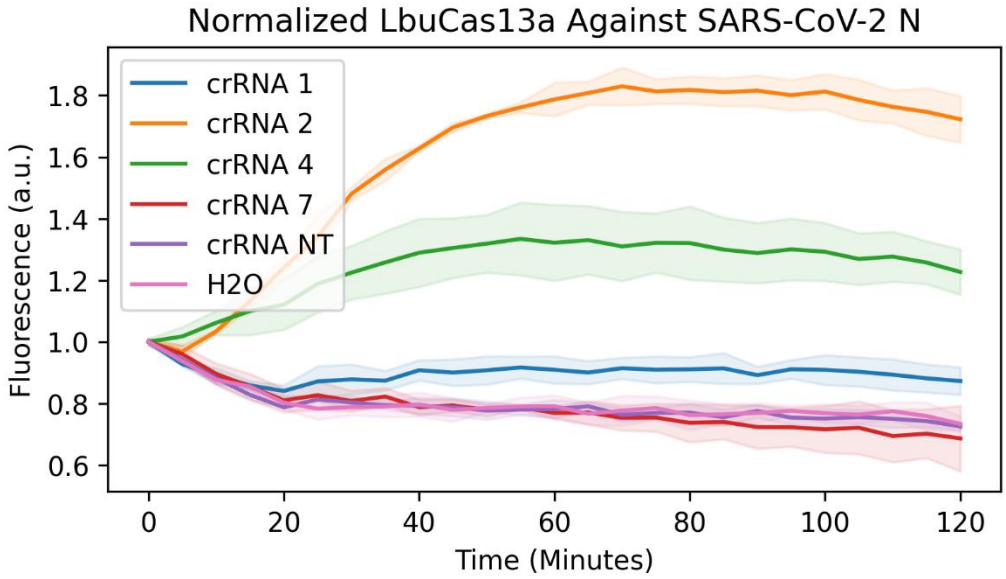


Figure 4-14. LbuCas13a activity against SARS-CoV-2 N gene RNA.

4.4.2.3 Functionalizing LbuCas13a with biotin-PEG3500 linkers

Adding linkers to enzymes risks altering their kinetics and ability to function, so the concentration of biotin-PEG3500-NHS linker to LbuCas13a was evaluated following characterization of the linkage reaction. 2 mg of biotin-PEG3500-NHS was reconstituted in 500 µL of anhydrous DMSO for a concentration of 500 µM. Next, the LbuCas13a buffer was exchanged with 20 mM HEPES at pH 7.4 to remove primary amines from the storage buffer. 21 µL of LbuCas13a was resuspended in HEPES to a total of 500 µL and centrifuged at 14k RCF in a 30 kDa Amicon Ultra (MilliporeSigma) for 13 minutes. The buffer exchange process was repeated a total of 3 times. 3 µL of LbuCas13a was then incubated with 3 µL of biotin-PEG3500-

NHS at various concentrations for 60 minutes at 4 °C. Following incubation, 3 µL of streptavidin-FITC at a concentration of 210 µM was added to the reaction and incubated for 60 minutes at 4 °C. The purpose of the streptavidin was to increase the molecular weight of the LbuCas13a to facilitate band separation of functionalized enzymes with PAGE. Samples were subsequently mixed with 2x Gel Loading Buffer (Thermo Fisher Scientific), denatured at 72 °C for 10 minutes, and a 4-12% Bis-Tris gel was run in MOPS at 100V for 200 minutes. Gels were stained with Sypro Ruby overnight. The resulting gel is provided in Figure 4-15.

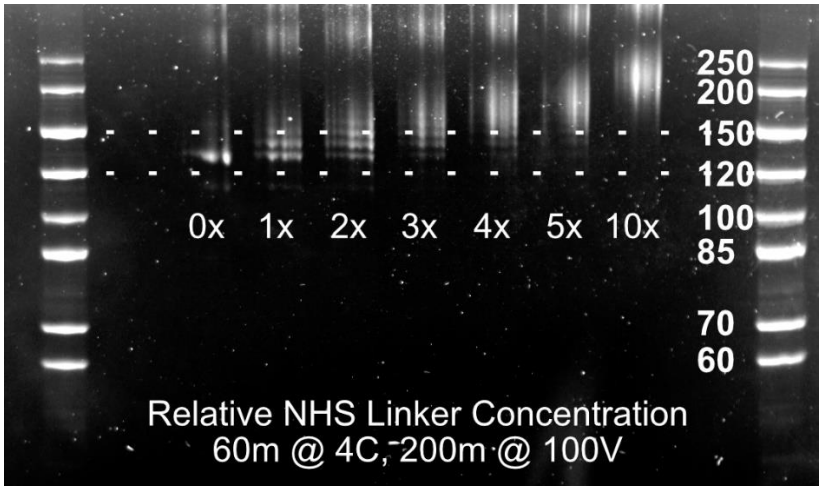


Figure 4-15. Functionalization of LbuCas13a with biotin-PEG3500 linkers.

The PAGE experiment demonstrated that the linkage reaction was successful and bands were clearly separated in the samples that had linker present. A line profile analysis (data not shown) was used to characterize the distribution of the number of linkers bound to LbuCas13a at different ratios. The analysis indicated an equimolar ratio of linker to endonuclease yielded the highest concentration of enzymes at a rate of roughly 40% of the total population. This linker to endonuclease ratio was used for all subsequent experiments unless otherwise indicated.

4.4.2.4 Assessing LbuCas13a activity with linkers

Endonuclease activity was evaluated with crRNAs 4 and NT with and without the addition of biotin-PEG3500-NHS linker and in the presence of streptavidin coated magnetic beads. The purpose of the assay was to identify if either linkers or beads altered enzymatic kinetics. RNP complexes were formed as previously described. Afterwards, half the RNP complexes were functionalized with 1x biotin-PEG3500-NHS linker as previously described. The controls without linker were incubated with DMSO instead. To prevent excess linker form binding to the streptavidin coated magnetic beads, the samples were buffer exchanged with HEPES 3 times as previously described. Afterwards, all samples were mixed with an equal volume of magnetic beads that had been washed with HEPES 3 times and placed on a rotator for 2 hours at 4 °C. Beads were pulled and a portion of the supernatant from all samples was processed with PAGE as previously described. Next, assay activity was assessed by saving the remaining supernatant, washing magnetic beads 3 times, and then using evaluating endonuclease activity for both supernatant and bead samples in the activity assay previously described. The resulting assay and PAGE gel are shown in Figure 4-16.

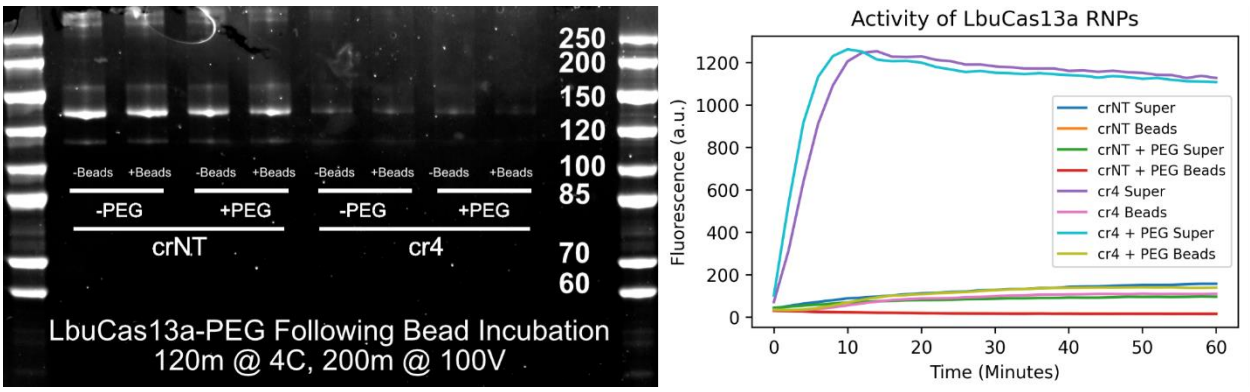


Figure 4-16. PAGE gel and endonuclease activity of functionalized LbuCas13a in the presence of streptavidin-coated magnetic beads.
LbuCas13a was tested for change in activity with due to functionalization or binding to beads. The addition of beads did not noticeably change the concentration of RNPs in the supernatant according to the PAGE gel. In addition, crRNA 4 only showed activity in supernatant and not with beads.

While the crRNA 4 sample continued to exhibit endonuclease activity, this was only observed in the supernatant samples and not the samples containing beads. This suggested that either the magnetic beads failed to bind the functionalized LbuCas13a or that the beads had an inhibitory effect on the assay. Also, given that roughly half of all the LbuCas13a endonucleases do not acquire a linker in the functionalization reactions, it was unclear if functionalized endonucleases retained catalytic activity. The negative controls with the crRNA NT performed as expected, which indicated that neither the beads and the linker contributed to the degradation of the fluorescent reporter. The gel samples with and without beads did not provide a clear indication that functionalized endonucleases were binding to the magnetic beads.

To assess if the linker could bind to magnetic beads, similar studies using fluorescent tagged antibodies and GFP were conducted. While the supernatant was fluorescent, the beads did not appear to have an increase in fluorescent signal. The process was repeated with different concentrations of linker, a longer 20k PEG linker, and a new set of beads, which also failed to yield a fluorescent signal. While the linker was previously shown to bind to soluble streptavidin, it did not appear to bind to the magnetic beads. An assay comparing the 3500 and 20k PEG linkers in the presence of beads with crRNA 4 was conducted in the event that the linker influenced RNP complex binding to beads and the endonuclease assay was more sensitive than binding fluorescent proteins. None of these assays yielded actionable results, so alternative strategies were considered.

4.4.2.5 Spatial multiplexing of LbuCas13a with microfluidic microarrays

Multiplexing LbuCas13a by physically anchoring the enzymes onto a solid substrate posed challenges that did not appear to have a nontrivial solution. An alternative multiplexing strategy could be leverage the use of microfluidic microchambers that contained RNP complexes with

unique crRNAs deposited into them with a microarrayer. In the assay, sample would be used to fill the microchambers and resolubilize the RNP complexes before the microchambers were sealed with an oil plug. Microfluidic device molds were fabricated with an Asiga Max X DLP 3D printer, coated with a Teflon-based aerosol mold release, and fabricated with Sylgard 184 Silicone Elastomer (Dow) in a 10:1 ratio and are shown in Figure 4-17. Prior to translating the assay into a microfluidic format, assay compatibility with different printing buffers was first explored.

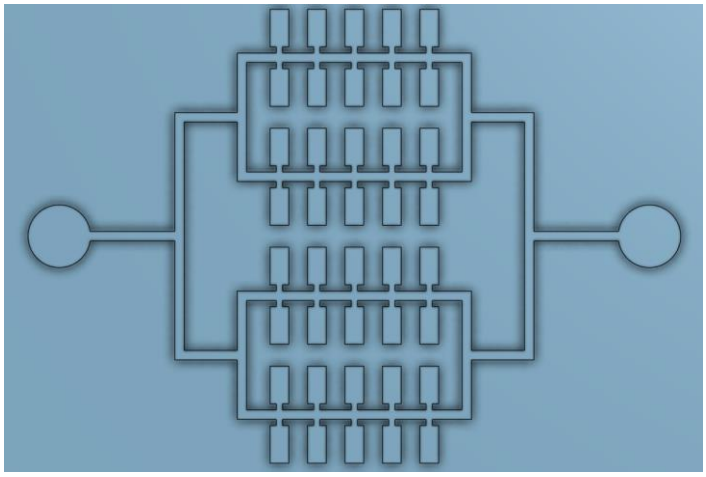


Figure 4-17. Design of microchamber array device for spatial multiplexing of LbuCas13a.

RNP complexes were prepared with crRNA 4 and buffer exchanged with 10% glycerol in 20mM HEPES using Amicon filters as previously described. Four 0.5 µL spots of the buffered complex were deposited onto a cleaned glass microscope slide and left to desiccate at room temperature while being stored away from direct sunlight. 20 µL of assay mix containing reporter, SARS-CoV-2 N gene and buffer was deposited onto the slide and sealed with a coverslip. The reaction was incubated at 37 °C for 1 hour before being measured on a Nikon TE microscope. No activity was observed, so RNP complexes were buffered in simulated printing buffers containing, Cas salt buffer diluted to 0.1x, Cas salt buffer with 5% glycerol, standard reaction buffer with BSA at a concentration of 1 mg/mL, standard reaction buffer with 5% PEG 3k, or Cas salt buffer with 5% trehalose and 0.1% Pluronic F-127, Cas salt buffer with 10% glycine, Cas salt buffer with 10%

trehalose, Cas salt buffer with 10% glycine and 10% trehalose. Cas salt buffer consisted of 20mM HEPES at pH 7.5, 50 mM KCl, and 5mM MgCl. LbuCas13a with crRNA 4 or a water control were mixed 1:9 with the printing buffer and 1 μL was desposited into an 8-chamber gasket (Grace Biolabs) mounted onto a cleaned microscope slide. This process was repeated twice. To assess the degradation of endonuclease activity as it was desiccated in different buffers, one set of gaskets was immediately mixed with 19 μL of reaction buffer, incubated at 37 °C for 30 minutes, and imaged on a Nikon TE. The other set was desiccated at 2.5 hours at room temperature before an identical reaction was conducted and measured.

Table 4-5. Endonuclease activity in various printing buffers

Condition	Loss of Activity (Net t_0 / Net t_{150})
CSB 0.1x	120
Glycerol 5%	2.10
BSA 1 mg/mL	0.76
PEG 3k 5%	9.96
Trehalose 5%, Pluronic 0.1%	11.46
Glycine 10%	512
Trehalose 10%	0.88
Glycine 10%, Trehalose 10%	7.45

To measure the loss of activity of the endonuclease in the printing buffer, the net median intensity of the 0 minute desiccation time point between the crRNA 4 and H_2O samples was divided by the net median intensity of the 150 minute desiccation time point. An ideal print buffer would have a net median intensity that did not change significantly over the desiccation process. Of the conditions assessed, the BSA 1 mg/mL and trehalose at 10% conditions appeared to maintain a moderately stable signal that slightly grew over time.

An initial test print screen of LbuCas13a RNPs with crRNAs 4 and NT that were mixed in 10% trehalose, 10% glycine, and a combination of 10% of trehalose and 10% glycine was

performed with a Scienion microarrayer. Single droplets of approximately 314 pL were deposited in an 8 by 5 array where each row represented a different buffer and RNP combination. Following deposition, slides were immediately lyophilized by placing slides on an aluminum block chilled to -70 °C and placed into a vacuum chamber depressurized to -29.5 inHg and left to dry overnight. The array could not be located following lyophilization due to particulate contamination of the glass slides and were not processed further. A screen of glass slides and vacuum freezing lyophilization conditions revealed that the particulate contamination was caused by overnight ambient conditions and processing glass slides in a sealed slide holder maintained a clean surface. Subsequent assays sought to evaluate whether the biochemical microenvironment of the microfluidic device would lead to degradation of the reporter in the endonuclease assay, but insufficient microfluidic control equipment at the time of the experiment led to serial failures and an ultimate pivot away from the platform.

4.4.2.6 Summary of spatially multiplexed LbuCas13a nucleic acid detection assay

A spatially multiplexed CRISPR-Cas assay would permit for a significantly higher depth of multiplexing than methods that depend on spectral differentiation of fluorescence reporters. The methods discussed here explored the feasibility of functionalizing LbuCas13a with long, flexible linkers that would allow the endonucleases to be preprogramed with crRNA and spotted as a microarray onto a solid substrate. The work utilized crRNAs that had been previously published to detect SARS-CoV-2 N gene as well as a custom non-targeting crRNA. Prior to spotting as a microarray, assay activity in the presence of magnetic beads that would sequester functionalized endonucleases and no signal was detected in the assays. Additional screens with fluorescent proteins, alternative beads, and an alternative, longer linker did not indicate a viable path for both functionalizing LbuCas13a and pulling down onto a solid substrate. An alternative method using

RNP complexes that were microarrayed and overlaid with a microfluidic device was investigated, but pivoted away from due to the lack of microfluidic control instrumentation at the time of the experiment. The data provided here is insufficient for assessing the viability of the methods or dismissing the approach entirely.

4.5 Future work

The work presented in this chapter explores the development of methods for enabling the continuous washing of immunoprecipitated T7 phage in the PhIP-seq protocol and possible approaches for translating PhIP-seq into a rapid diagnostic as well as increasing the depth of multiplexing of CRISPR-Cas assays through the use of spatially arrayed and immobilized endonucleases. Macrofluidic methods for magnetic bead washing included the development of an inverted magnetic bead washing platform and a platform that leveraged in-tubing capture and washing of phage. While neighboring samples had no risk of mixing in the in-tubing approach, the interface where tubing interfaced with sample aspiration needles produced a dead volume where magnetic beads were trapped and likely would have resulted in sample mixing and contamination between sequential plates. This approach did not appear to be viable. The method leveraging a magnet array and sheath may be feasible once engineering limitations and minor protocol adjustments are made and could be worth reexamining.

The microfluidic approach for automating the process of phage immunoprecipitation and washing was characterized more thoroughly. Validation of surface chemistry, optimization of molecular concentrations and washing conditions, and an initial attempt at performing the PhIP-seq protocol with multiple rounds of amplification were discussed. While many issues were addressed in the process and multiple design iterations were fabricated, the method still requires further development. Every phage immunoprecipitation platform has a biochemical

microenvironment that naturally and weakly enriches for specific epitopes due the diversity presented. A control study that characterizes the background signature of the microfluidic platform should be preformed. In addition, validation with a complex phage peptidome library is necessary and using control antibodies against native library without an overexpressed target would be beneficial for assessing amplification of a target under ideal conditions. The experiments conducted used an overexpressed target and it was not possible to observe signal enhancement due to the high starting baseline. After PhIP-seq has been performed under ideal conditions, the following experiments should be to characterize the platform with diseased sera and compare the signature of the microfluidic assay to that of the traditional benchtop assay. PhIP-seq is a naturally noisy method, but the consistent presence epitope signatures should be observable. An automated immunoprecipitation and washing method with a microfluidic device may be realistic and achievable. A microfluidic approach would improve the standardization of the PhIP-seq method, reduce assay time, and allow for new experimental methods to be explored, such as characterization of autoantibody binding strengths through gradient elution.

Evaluation of the translating clinically relevant autoimmune findings for PhIP-seq into a diagnostic leveraging the Luminex xMAP platform highlights a possible approach for implementing a confirmatory assay. Epitopes for characterized ROHHAD, spontaneous preterm birth, autoimmune polyglandular syndrome type 1, and systemic lupus erythematosus diseases were translated onto Luminex MagPlex microspheres and used to test for reactivity in SLE patient sera, APS1 patient sera, and ZSCAN1 monoclonals. Findings showed that the complex SLE sera had a high signal, ZSCAN1 monoclonals were detectable at 100 ng/mL when the epitope was properly solubilized, while the APS1 sera did not appear to be reactive. While the xMAP platform can achieve a limit of detection on the order of pg/mL, many autoantibodies are often present in

sera in concentrations between 100 μg/mL and 100 ng/mL and the detection of ZSCAN1 was physiologically relevant. While it is unclear whether the xMAP platform would be a practical approach for all autoantibodies in diseased sera, there are indications that this approach may be viable for detecting autoantibodies with further optimization in the reconstitution of epitopes and further characterization of the specificity and sensitivity of the platform for various diseases.

The development of a multiplexed CRISPR-Cas assay through spatial arraying of RNP complexes was investigated, but conclusive results pertaining to the viability of the approach were not obtained. While biotin PEG linkers were confirmed to bind to LbuCas13a, assays found that functionalized RNPs mixed with streptavidin coated beads were not reactive. Beads, linkers, and concentrations were varied, but endonuclease activity on the solid substrates was not detected. While the primary goal was to transition the functionalized RNPs onto a planar solid substrate, beads would have favorable reaction kinetics. The lack of signal with the beads led to a pivot to spatially arrayed RNPs that were to be encapsulated by a microchamber microfluidic device. Printing buffers were screened to identify which conditions would preserve endonuclease activity once desiccated and rehydrated. However, transitioning RNPs into a printed microarray with a microfluidic device proved to be problematic as microfluidic control instrumentation had not been developed at the time of the work. This approach may be viable, but it requires further investigation into the whether or not the microfluidic environment contains innate catalytic activity, confirmation of endonuclease activity of printed RNPs, concentration optimizations, and likely a microfluidic device redesign for fabrication yield and device robustness.

Chapter 5.
Future Directions

5.1 Overview

This dissertation describes persistent barriers in laboratory automation, protein purification, and high-throughput molecular diagnostics. Chapter 1 outlined how preanalytical volumetric variability, vendor lock-in, high capital costs, and rigid, proprietary black boxes hinder the widespread adoption of automated fluidic platforms and limit clinical diagnostic throughput. In response. Chapter 2 demonstrated an open-source, parallelized, four-channel fast protein liquid chromatography (FPLC) system that enabled the rapid, automated purification of recombinant antigens and diverse antibodies with high consistency and low inter-channel variance. Chapter 3 detailed the design and implementation of an extensible, low cost Microfluidic Control Platform (MCP) leveraging an open-source model-view-controller (MVC) architecture, custom solenoid driver electronics, and Raspberry Pi embedded computing. Chapter 4 explored diagnostic and assay automation, investigating macrofluidic and microfluidic approaches for Phage Immunoprecipitation Sequencing (PhIP-seq) wash protocols, translating PhIP-seq-discovered autoimmune epitopes to a multiplexed Luminex xMAP bead assay, and developing spatially multiplexed CRISPR-Cas endonuclease assays to bypass the spectral limitations of existing fluorescence-based diagnostic techniques.

While these contributions demonstrate how open-source engineering can democratize instrumentation and streamline complex biological assays, several technical challenges and translational opportunities remain. This chapter outlines future engineering extensions and a long-term vision for open, automated laboratory ecosystems.

5.2 Extension of Existing Technology

5.2.1 Scaling and advanced automation of the FPLC system

The four-channel parallel protein purifier established a cost-effective benchmark for milligram-scale purification. To meet industrial and high-throughput structural biology demands, the system can be scaled and refined to include (1) inline analytical instrumentation, (2) automated gradient elution and buffer exchange protocols, and (3) increased number of parallel columns and an autosampler to improve throughput. As an example, integrating multi-wavelength UV-vis absorbance flow cells, pH meters, and conductivity sensors into each parallel channel would enable real-time chromatogram generation, automated peak detection, and dynamic fraction collection based on target protein elution profiles. Similarly, expanding the fluidic manifold to include automated dual-pump gradient mixing would allow complex salt or pH gradient elutions. The addition of a secondary integrated and automated desalting or size-exclusion chromatography (SEC) columns in series would also allow fully automated purification, buffer exchange, and concentration in a single run. However, scaling the fluidic manifold from 4 to 16 or 96 parallel channels while being coupled with robotic autosamplers for automated crude lysate loading would transform the benchtop system into a high-throughput platform capable of processing hundreds of unique protein variants per day for structural genomics, antibody engineering, or vaccine candidate screening. The addition of all these systems would likely put the system in a class of its own.

5.2.2 Enhancements to the Microfluidic Control Platform (MCP)

Although the MCP successfully coordinates 24 solenoid valves, dual pressure lines, and auxiliary devices such as syringe pumps via standard HTTP and WebSocket APIs, several hardware and software iterations can further expand its capabilities. This includes the use of (1)

closed-loop pressure and flow control, (2) scaling the system with modular valve controllers, and (3) integrating environmental and optical sensing. The current platform relies on manual regulators and open-loop pressure delivery. Incorporating inline microfluidic flow sensors (e.g., thermal or optical flow meters) paired with digital proportional pressure regulators would enable real-time closed-loop feedback. This capability would eliminate flow drift caused by dynamic channel resistance, temperature fluctuations, or viscosity changes during multi-reagent assay execution. To support ultra high density elastomeric devices featuring thousands of Quake style microvalves, the solenoid driver electronics can be modularized. Daisy-chaining driver PCBs over an SPI or CAN bus architecture would allow a single Raspberry Pi compute node to control hundreds of pneumatic channels with synchronous switching. Future chassis revisions can incorporate onboard Peltier heating/cooling modules, LED illumination, and CMOS camera sensors directly controlled through the plfluidic software framework. This integration would allow autonomous, image-guided valve switching and real-time fluorescent signal monitoring during on-chip incubation or amplification. The extensible design of the hardware naturally lends itself to further refinement.

5.2.3 Refinement of automated PhIP-seq washing platforms

As previously discussed in Chapter 4, continuous washing methods using the macrofluidic magnet array and the microfluidic device continue to hold promise as being viable strategies for enabling time and resource intensive aspects of the PhIP-seq protocol. The microfluidic platform likely needs a few more control experiments to be fully validated and the magnet array could use engineering and protocol adjustments to facilitate proper sample positioning and retainment before validation studies can be initiated. Ideally, following the automation of the enrichment, next milestones would be to automate the phage amplification processes so that a full round of enrichment and amplification can be performed on a single device. As a moonshot, coupling a

hybridization assay to the output of the phage washing and amplification platform would transform the technique into an all-in-one sample to answer autoantibody detection platform.

5.2.4 Optimization of Spatially Multiplexed CRISPR-Cas Diagnostics

The spatial arraying of Cas endonucleases represents a promising alternative to overcome the spectral multiplexing limits of traditional fluorophore-quencher assays. However, preserving enzyme catalytic cleavage activity upon surface immobilization remains a critical hurdle. Key avenues for future investigation include (1) site-specific linker conjugation, (2) hydrogel encapsulation and printing, and (3) multilayer microfluidic platforms. Random biotinylation via amine-reactive NHS-PEG linkers may cause steric hindrance or alter the active site of the endonuclease, leading to loss of collateral cleavage activity upon bead or surface immobilization. Future work should explore site-specific terminal tags to anchor the Cas protein to planar substrates or hydrogel matrices in an orientation that preserves catalytic flexibility. Additionally, drying or desiccation in printing buffers was observed to lead to a loss of catalytic activity. By arraying RNP complexes inside porous, hydrated hydrogel droplets, it is possible that the endonucleases can maintain activity in a liquid environment. Similarly, preliminary microfluidic studies were implemented prior to the availability of photolithography and microfluidic control resources. Reimplementing the assay with a multilayer microfluidic device may yield productive multiplexed endonuclease assays.

5.3 Long term vision and outlook

The overarching vision driving this dissertation is the democratization of scientific instrumentation. For decades, advanced biological research has been restricted by cost-prohibitive and inflexible commercial platforms with black box ecosystems. By providing fully documented,

hardware, modular electronics, and intuitive, web-accessible software interfaces, open platforms lower the financial and engineering barriers for academic laboratories, lower income institutions, and global research centers.

Looking forward, the convergence of open-source fluidic hardware, standardized digital protocols, and real-time machine learning will enable the realization of autonomous, closed loop laboratories. In such environments, open-source controllers execute automated experimental protocols, integrated sensors analyze outputs in real time, and algorithms dynamically adjust experimental parameters for the next iteration. Whether applied to rapid pandemic response, personalized autoimmune diagnostics, or novel therapeutic engineering, accessible and automated fluidic platforms will remain fundamental in accelerating the scientific discovery process and improving human health.

References

1. Lippi G, Chance JJ, Church S, et al. Preanalytical quality improvement: from dream to reality. *Clinical Chemistry and Laboratory Medicine*. 2011;49(7). doi:10.1515/CCLM.2011.600

2. Guan XL, Chang DPS, Mok ZX, Lee B. Assessing variations in manual pipetting: An under-investigated requirement of good laboratory practice. *Journal of Mass Spectrometry and Advances in the Clinical Lab*. 2023;30:25-29. doi:10.1016/j.jmsacl.2023.09.001

3. Özcan Ö, Grundel PJM, Heijboer AC, Den Elzen WPJ. Pipetting Performance across Laboratory Personnel: A Comparative Study. *The Journal of Applied Laboratory Medicine*. 2026;11(3):460-469. doi:10.1093/jalm/jfag006

4. Schuster C, Habler K, Vogeser M. Effect of gravimetric correction and type of pipettes used in sample preparation on the precision of LC–MS/MS-based analyses. *Clinical Biochemistry*. 2021;91:63-66. doi:10.1016/j.clinbiochem.2021.01.017

5. Pandya K, Ray CA, Brunner L, Wang J, Lee JW, DeSilva B. Strategies to minimize variability and bias associated with manual pipetting in ligand binding assays to assure data quality of protein therapeutic quantification. *Journal of Pharmaceutical and Biomedical Analysis*. 2010;53(3):623-630. doi:10.1016/j.jpba.2010.04.025

6. Hentz NG, Knaide TR. Effect of Liquid-Handling Accuracy on Assay Performance. *SLAS Technology*. 2014;19(2):153-162. doi:10.1177/2211068213504095

7. Hayden RT, Yan X, Wick MT, et al. Factors Contributing to Variability of Quantitative Viral PCR Results in Proficiency Testing Samples: a Multivariate Analysis. *J Clin Microbiol*. 2012;50(2):337-345. doi:10.1128/JCM.01287-11

8. Yang X, Wang X, Li B, Chu J. A high-precision automated liquid pipetting device with an interchangeable tip. *Review of Scientific Instruments*. 2023;94(9):094102. doi:10.1063/5.0139565

9. Almeida CAM, Roberts SG, Laird R, et al. Automation of the ELISpot assay for high-throughput detection of antigen-specific T-cell responses. *Journal of Immunological Methods*. 2009;344(1):1-5. doi:10.1016/j.jim.2009.02.007

10. Van Dorpe S, Lippens L, Boiy R, et al. Integrating automated liquid handling in the separation workflow of extracellular vesicles enhances specificity and reproducibility. *J Nanobiotechnol*. 2023;21(1):157. doi:10.1186/s12951-023-01917-z

11. Cai H, Wang D, Zhao Y, Yang C. Recent Advances in Microfluidic Chip Technology for Laboratory Medicine: Innovations and Artificial Intelligence Integration. *Biosensors*. 2026;16(2):104. doi:10.3390/bios16020104

12. Damir S, Delgado-Licona F, deMello A, Abolhasani M. Autonomous microfluidic labs: progress and prospects. *Lab Chip*. 2026;26(5):1014-1024. doi:10.1039/D5LC00908A

13. Zhang X, Liu A, Wang Y, et al. Advancements and Future Perspectives of Microfluidic Technology in Pediatric Healthcare. *Smart Medicine*. 2025;4(3):e70018. doi:10.1002/smmd.70018

14. Wartmann D, Rothbauer M, Kuten O, et al. Automated, Miniaturized, and Integrated Quality Control-on-Chip (QC-on-a-Chip) for Cell-Based Cancer Therapy Applications. *Front Mater*. 2015;2. doi:10.3389/fmats.2015.00060

15. Lee PJ, Ghorashian N, Gaige TA, Hung PJ. Microfluidic System for Automated Cell-Based Assays. *JALA: Journal of the Association for Laboratory Automation*. 2007;12(6):363-367. doi:10.1016/j.jala.2007.07.001

16. Holland I, Davies JA. Automation in the Life Science Research Laboratory. *Front Bioeng Biotechnol*. 2020;8:571777. doi:10.3389/fbioe.2020.571777

17. Doloi S, Das M, Li Y, et al. Democratizing self-driving labs: advances in low-cost 3D printing for laboratory automation. *Digital Discovery*. 2025;4(7):1685-1721. doi:10.1039/D4DD00411F

18. Pearce JM. Building Research Equipment with Free, Open-Source Hardware. *Science*. 2012;337(6100):1303-1304. doi:10.1126/science.1228183

19. Minchala LI, Peralta J, Mata-Quevedo P, Rojas J. An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software. *Applied Sciences*. 2020;10(14):4696. doi:10.3390/app10144696

20. Reddy KJ. Accessibility and Affordability of Technologies. In: *Innovations in Neurocognitive Rehabilitation*. Springer Nature Switzerland; 2025:285-304. doi:10.1007/978-3-031-88117-6_13

21. Wenzel T. Open hardware: From DIY trend to global transformation in access to laboratory equipment. *PLoS Biol*. 2023;21(1):e3001931. doi:10.1371/journal.pbio.3001931

22. Sultan HM, Batool L, Hussain I, Sabir M, Mujahid R, Mazhar M. An Overview of Protein Purification Strategies and Applications. *Medicinal and Medical Chemistry*. 2025;2(2). doi:10.22034/medmedchem.2025.531293.1037

23. Tang S, Tao J, Li Y. Challenges and solutions for the downstream purification of therapeutic proteins. *Antibody Therapeutics*. 2024;7(1):1-12. doi:10.1093/abt/tbad028

24. Camper DV, Viola RE. Fully automated protein purification. *Analytical Biochemistry*. 2009;393(2):176-181. doi:10.1016/j.ab.2009.07.009

25. Berkelman T, Urban M. Bio-Rad Technical Note 5513. Published online 2006.

26. Enders A, Grünberger A, Bahnemann J. Towards Small Scale: Overview and Applications of Microfluidics in Biotechnology. *Mol Biotechnol*. 2024;66(3):365-377. doi:10.1007/s12033-022-00626-6

27. Squires TM, Quake SR. Microfluidics: Fluid physics at the nanoliter scale. *Rev Mod Phys*. 2005;77(3):977-1026. doi:10.1103/RevModPhys.77.977

28. Whitesides GM. The origins and the future of microfluidics. *Nature*. 2006;442(7101):368-373. doi:10.1038/nature05058

29. Kirby BJ. *Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices*. 1st ed. Cambridge University Press; 2012. doi:10.1017/CBO9780511760723

30. Bringer MR, Gerdts CJ, Song H, Tice JD, Ismagilov RF. Microfluidic systems for chemical kinetics that rely on chaotic mixing in droplets. Ottino JM, Wiggins SR, eds. *Philosophical Transactions of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*. 2004;362(1818):1087-1104. doi:10.1098/rsta.2003.1364

31. Madadelahi M, Madou M. Rational PCR Reactor Design in Microfluidics. *Micromachines*. 2023;14(8):1533. doi:10.3390/mi14081533

32. Duncombe TA, Tentori AM, Herr AE. Microfluidics: reframing biological enquiry. *Nat Rev Mol Cell Biol*. 2015;16(9):554-567. doi:10.1038/nrm4041

33. An YQ, Huang SL, Xi BC, et al. Ultrafast Microfluidic PCR Thermocycler for Nucleic Acid Amplification. *Micromachines*. 2023;14(3):658. doi:10.3390/mi14030658

34. Luo X, Chen JY, Ataei M, Lee A. Microfluidic Compartmentalization Platforms for Single Cell Analysis. *Biosensors*. 2022;12(2):58. doi:10.3390/bios12020058

35. Gharib G, Bütün İ, Muganlı Z, et al. Biomedical Applications of Microfluidic Devices: A Review. *Biosensors*. 2022;12(11):1023. doi:10.3390/bios12111023

36. Duffy DC, McDonald JC, Schueller OJA, Whitesides GM. Rapid Prototyping of Microfluidic Systems in Poly(dimethylsiloxane). *Anal Chem*. 1998;70(23):4974-4984. doi:10.1021/ac980656z

37. Xia Y, Whitesides GM. Soft Lithography. *Angewandte Chemie International Edition*. 1998;37(5):550-575. doi:10.1002/(SICI)1521-3773(19980316)37:5<550::AID-ANIE550>3.0.CO;2-G

38. Unger MA, Chou HP, Thorsen T, Scherer A, Quake SR. Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography. *Science*. 2000;288(5463):113-116. doi:10.1126/science.288.5463.113

39. McDonald JC, Whitesides GM. Poly(dimethylsiloxane) as a Material for Fabricating Microfluidic Devices. *Acc Chem Res*. 2002;35(7):491-499. doi:10.1021/ar010110q

40. Grover WH, Skelley AM, Liu CN, Lagally ET, Mathies RA. Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices. *Sensors and Actuators B: Chemical*. 2003;89(3):315-323. doi:10.1016/S0925-4005(02)00468-9

41. Melin J, Quake SR. Microfluidic Large-Scale Integration: The Evolution of Design Rules for Biological Automation. *Annu Rev Biophys Biomol Struct*. 2007;36(1):213-231. doi:10.1146/annurev.biophys.36.040306.132646

42. Studer V, Hang G, Pandolfi A, Ortiz M, French Anderson W, Quake SR. Scaling properties of a low-actuation pressure microfluidic valve. *Journal of Applied Physics*. 2004;95(1):393-398. doi:10.1063/1.1629781

43. Thorsen T, Maerkl SJ, Quake SR. Microfluidic Large-Scale Integration. *Science*. 2002;298(5593):580-584. doi:10.1126/science.1076996

44. Sia SK, Whitesides GM. Microfluidic devices fabricated in Poly(dimethylsiloxane) for biological studies. *Electrophoresis*. 2003;24(21):3563-3576. doi:10.1002/elps.200305584

45. Sackmann EK, Fulton AL, Beebe DJ. The present and future role of microfluidics in biomedical research. *Nature*. 2014;507(7491):181-189. doi:10.1038/nature13118

46. Walsh DI, Kong DS, Murthy SK, Carr PA. Enabling Microfluidics: from Clean Rooms to Makerspaces. *Trends in Biotechnology*. 2017;35(5):383-392. doi:10.1016/j.tibtech.2017.01.001

47. Ortseifen V, Viefhues M, Wobbe L, Grünberger A. Microfluidics for Biotechnology: Bridging Gaps to Foster Microfluidic Applications. *Front Bioeng Biotechnol*. 2020;8:589074. doi:10.3389/fbioe.2020.589074

48. Mark D, Haeberle S, Roth G, Von Stetten F, Zengerle R. Microfluidic Lab-on-a-Chip Platforms: Requirements, Characteristics and Applications. In: Kakaç S, Kosoy B, Li D, Pramuanjaroenkij A, eds. *Microfluidics Based Microsystems*. NATO Science for Peace and Security Series A: Chemistry and Biology. Springer Netherlands; 2010:305-376. doi:10.1007/978-90-481-9029-4_17

49. Brower K, Puccinelli RR, Markin CJ, et al. An open-source, programmable pneumatic setup for operation and automated control of single- and multi-layer microfluidic devices. *HardwareX*. 2018;3:117-134. doi:10.1016/j.ohx.2017.10.001

50. Pisetsky DS. Pathogenesis of autoimmune disease. *Nat Rev Nephrol*. 2023;19(8):509-524. doi:10.1038/s41581-023-00720-1

51. Shurin M, Wheeler S. Clinical Significance of Uncommon, Non-Clinical, and Novel Autoantibodies. *ITT*. 2024;Volume 13:215-234. doi:10.2147/ITT.S450184

52. Choi MY, Costenbader KH. Understanding the Concept of Pre-Clinical Autoimmunity: Prediction and Prevention of Systemic Lupus Erythematosus: Identifying Risk Factors and Developing Strategies Against Disease Development. *Front Immunol*. 2022;13:890522. doi:10.3389/fimmu.2022.890522

53. Arbuckle MR, McClain MT, Rubertone MV, et al. Development of Autoantibodies before the Clinical Onset of Systemic Lupus Erythematosus. *N Engl J Med*. 2003;349(16):1526-1533. doi:10.1056/NEJMoa021933

54. Koenig M, Fritzler MJ, Targoff IN, Troyanov Y, Senécal JL. Heterogeneity of autoantibodies in 100 patients with autoimmune myositis: insights into clinical features and outcomes. *Arthritis Res Ther*. 2007;9(4):R78. doi:10.1186/ar2276

55. Di Pietro L, Chiccoli F, Salvati L, et al. Anti-RuvBL1/2 Autoantibodies Detection in a Patient with Overlap Systemic Sclerosis and Polymyositis. *Antibodies*. 2023;12(1):13. doi:10.3390/antib12010013

56. Huang Z, Gunarathne SMS, Liu W, et al. PhIP-Seq: methods, applications and challenges. *Front Bioinform*. 2024;4:1424202. doi:10.3389/fbinf.2024.1424202

57. Carlton LH, McGregor R, Moreland NJ. Human antibody profiling technologies for autoimmune disease. *Immunol Res*. 2023;71(4):516-527. doi:10.1007/s12026-023-09362-8

58. Vazquez SE, Mann SA, Bodansky A, et al. Autoantibody discovery across monogenic, acquired, and COVID-19-associated autoimmunity with scalable PhIP-seq. *eLife*. 2022;11:e78550. doi:10.7554/eLife.78550

59. Li S, Song G, Bai Y, et al. Applications of Protein Microarrays in Biomarker Discovery for Autoimmune Diseases. *Front Immunol*. 2021;12:645632. doi:10.3389/fimmu.2021.645632

60. Koo BK, Chae S, Kim KM, et al. Identification of Novel Autoantibodies in Type 1 Diabetic Patients Using a High-Density Protein Microarray. *Diabetes*. 2014;63(9):3022-3032. doi:10.2337/db13-1566

61. Predki PF, Mattoon D, Bangham R, Schweitzer B, Michaud G. Protein microarrays: A new tool for profiling antibody cross-reactivity. *HAB*. 2006;14(1-2):7-15. doi:10.3233/HAB-2005-141-202

62. Duarte JG, Blackburn JM. Advances in the development of human protein microarrays. *Expert Review of Proteomics*. 2017;14(7):627-641. doi:10.1080/14789450.2017.1347042

63. Larman HB, Zhao Z, Laserson U, et al. Autoantigen discovery with a synthetic human peptidome. *Nat Biotechnol*. 2011;29(6):535-541. doi:10.1038/nbt.1856

64. O'Donovan B, Mandel-Brehm C, Vazquez SE, et al. High-resolution epitope mapping of anti-Hu and anti-Yo autoimmunity by programmable phage display. *Brain Communications*. 2020;2(2):fcaa059. doi:10.1093/braincomms/fcaa059

65. Mohan D, Wansley DL, Sie BM, et al. PhIP-Seq characterization of serum antibodies using oligonucleotide-encoded peptidomes. *Nat Protoc*. 2018;13(9):1958-1978. doi:10.1038/s41596-018-0025-6

66. Mandel-Brehm C, Dubey D, Kryzer TJ, et al. Kelch-like Protein 11 Antibodies in Seminoma-Associated Paraneoplastic Encephalitis. *N Engl J Med*. 2019;381(1):47-54. doi:10.1056/NEJMoa1816721

67. Vazquez SE, Ferré EM, Scheel DW, et al. Identification of novel, clinically correlated autoantigens in the monogenic autoimmune syndrome APS1 by proteome-wide PhIP-Seq. *eLife*. 2020;9:e55053. doi:10.7554/eLife.55053

68. Mandel-Brehm C, Benson LA, Tran B, et al. ZSCAN1 Autoantibodies Are Associated with Pediatric Paraneoplastic ROHHAD. *Annals of Neurology*. 2022;92(2):279-291. doi:10.1002/ana.26380

69. Laborde CM, Castro-Santos P, Díaz-Peña R. Contribution of Multiplex Immunoassays to Rheumatoid Arthritis Management: From Biomarker Discovery to Personalized Medicine. *JPM*. 2020;10(4):202. doi:10.3390/jpm10040202

70. Tait BD. Detection of HLA Antibodies in Organ Transplant Recipients – Triumphs and Challenges of the Solid Phase Bead Assay. *Front Immunol*. 2016;7. doi:10.3389/fimmu.2016.00570

71. Feng A, Yang EY, Moore AR, et al. Autoantibodies are highly prevalent in non–SARS-CoV-2 respiratory infections and critical illness. *JCI Insight*. 2023;8(3):e163150. doi:10.1172/jci.insight.163150

72. Zhang H, Deng J, Liu Y, et al. Luminex-based pilot study identifies novel serum autoantibodies associated with disease activity in patients with lupus nephritis. *Clin Rheumatol*. 2025;44(7):2751-2762. doi:10.1007/s10067-025-07528-3

73. Zamecnik CR, Sowa GM, Abdelhak A, et al. An autoantibody signature predictive for multiple sclerosis. *Nat Med*. 2024;30(5):1300-1308. doi:10.1038/s41591-024-02938-3

74. Welch NL, Zhu M, Hua C, et al. Multiplexed CRISPR-based microfluidic platform for clinical testing of respiratory viruses and identification of SARS-CoV-2 variants. *Nat Med*. 2022;28(5):1083-1094. doi:10.1038/s41591-022-01734-1

75. Ackerman CM, Myhrvold C, Thakku SG, et al. Massively multiplexed nucleic acid detection with Cas13. *Nature*. 2020;582(7811):277-282. doi:10.1038/s41586-020-2279-8

76. Rahimi S, Balusamy SR, Perumalsamy H, Ståhlberg A, Mijakovic I. CRISPR-Cas target recognition for sensing viral and cancer biomarkers. *Nucleic Acids Research*. 2024;52(17):10040-10067. doi:10.1093/nar/gkae736

77. Han J. Multiplex PCR in Molecular Differential Diagnosis of Microbial Infections: Methods, Utility, and Platforms. In: Tang YW, Stratton CW, eds. *Advanced Techniques in Diagnostic Microbiology*. Springer US; 2013:627-646. doi:10.1007/978-1-4614-3970-7_34

78. Gu W, Deng X, Lee M, et al. Rapid pathogen detection by metagenomic next-generation sequencing of infected body fluids. *Nat Med*. 2021;27(1):115-124. doi:10.1038/s41591-020-1105-z

79. Gu W, Miller S, Chiu CY. Clinical Metagenomic Next-Generation Sequencing for Pathogen Detection. *Annu Rev Pathol Mech Dis*. 2019;14(1):319-338. doi:10.1146/annurev-pathmechdis-012418-012751

80. Lu D, Kalantar KL, Chu VT, et al. Simultaneous detection of pathogens and antimicrobial resistance genes with the open source, cloud-based, CZ ID pipeline. *Microbiology*. Preprint posted online April 12, 2024. doi:10.1101/2024.04.12.589250

81. Kalantar KL, Carvalho T, De Bourcy CFA, et al. IDseq—An open source cloud-based pipeline and analysis service for metagenomic pathogen detection and monitoring. *GigaScience*. 2020;9(10):giaa111. doi:10.1093/gigascience/giaa111

82. Gootenberg JS, Abudayyeh OO, Lee JW, et al. Nucleic acid detection with CRISPR-Cas13a/C2c2. *Science*. 2017;356(6336):438-442. doi:10.1126/science.aam9321

83. Chen JS, Ma E, Harrington LB, et al. CRISPR-Cas12a target binding unleashes indiscriminate single-stranded DNase activity. *Science*. 2018;360(6387):436-439. doi:10.1126/science.aar6245

84. Broughton JP, Deng X, Yu G, et al. CRISPR–Cas12-based detection of SARS-CoV-2. *Nat Biotechnol*. 2020;38(7):870-874. doi:10.1038/s41587-020-0513-4

85. Fozouni P, Son S, Díaz De León Derby M, et al. Amplification-free detection of SARS-CoV-2 with CRISPR-Cas13a and mobile phone microscopy. *Cell*. 2021;184(2):323-333.e9. doi:10.1016/j.cell.2020.12.001

86. Li L, Li S, Wu N, et al. HOLMESv2: A CRISPR-Cas12b-Assisted Platform for Nucleic Acid Detection and DNA Methylation Quantitation. *ACS Synth Biol*. 2019;8(10):2228-2237. doi:10.1021/acssynbio.9b00209

87. Kellner MJ, Koob JG, Gootenberg JS, Abudayyeh OO, Zhang F. SHERLOCK: nucleic acid detection with CRISPR nucleases. *Nat Protoc*. 2019;14(10):2986-3012. doi:10.1038/s41596-019-0210-2

88. Li Y, Arce A, Lucci T, Rasmussen RA, Lucks JB. Dynamic RNA synthetic biology: new principles, practices and potential. *RNA Biology*. 2023;20(1):817-829. doi:10.1080/15476286.2023.2269508

89. Ramachandran A, Santiago JG. CRISPR Enzyme Kinetics for Molecular Diagnostics. *Anal Chem*. 2021;93(20):7456-7464. doi:10.1021/acs.analchem.1c00525

90. Puccinelli RR, Sama SS, Worthington CM, Puschnik AS, Pak JE, Gómez-Sjöberg R. *Open-Source Milligram-Scale, Four Channel, Automated Protein Purification System*. Bioengineering; 2023. doi:10.1101/2023.08.09.552685

91. Tripathi NK, Shrivastava A. Recent Developments in Bioprocessing of Recombinant Proteins: Expression Hosts and Process Development. *Front Bioeng Biotechnol*. 2019;7:420. doi:10.3389/fbioe.2019.00420

92. Winters D, Chu C, Walker K. Automated two-step chromatography using an ÄKTA equipped with in-line dilution capability. *Journal of Chromatography A*. 2015;1424:51-58. doi:10.1016/j.chroma.2015.10.092

93. Holenstein F, Eriksson C, Erlandsson I, et al. Automated harvesting and 2-step purification of unclarified mammalian cell-culture broths containing antibodies. *Journal of Chromatography A*. 2015;1418:103-109. doi:10.1016/j.chroma.2015.09.040

94. Luan P, Lee S, Arena TA, et al. Automated high throughput microscale antibody purification workflows for accelerating antibody discovery. *mAbs*. 2018;10(4):624-635. doi:10.1080/19420862.2018.1445450

95. Hadfield J, Megill C, Bell SM, et al. Nextstrain: real-time tracking of pathogen evolution. Kelso J, ed. *Bioinformatics*. 2018;34(23):4121-4123. doi:10.1093/bioinformatics/bty407

96. Raybould MIJ, Kovaltsuk A, Marks C, Deane CM. CoV-AbDab: the coronavirus antibody database. Wren J, ed. *Bioinformatics*. 2021;37(5):734-735. doi:10.1093/bioinformatics/btaa739

97. Corrie BD, Marthandan N, Zimonja B, et al. iReceptor: A platform for querying and analyzing antibody/B-cell and T-cell receptor repertoire data across federated repositories. *Immunol Rev*. 2018;284(1):24-41. doi:10.1111/imr.12666

98. UM10204 I^2C-bus specification and user manual Rev. 7.0. Published online October 1, 2021.

99. Advamation - Know-How - Raspberry Pi I2C Bug. Accessed January 25, 2023. https://www.advamation.com/knowhow/raspberrypi/rpi-i2c-bug.html

100. I2C clock-stretching bug · Issue #4884 · raspberrypi/linux. GitHub. Accessed January 25, 2023. https://github.com/raspberrypi/linux/issues/4884

101. Amanat F, Stadlbauer D, Strohmeier S, et al. A serological assay to detect SARS-CoV-2 seroconversion in humans. *Nat Med*. 2020;26(7):1033-1036. doi:10.1038/s41591-020-0913-5

102. Byrum JR, Waltari E, Janson O, et al. MultiSero: An Open-Source Multiplex-ELISA Platform for Measuring Antibody Responses to Infection. *Pathogens*. 2023;12(5):671. doi:10.3390/pathogens12050671

103. ter Meulen J, van den Brink EN, Poon LLM, et al. Human Monoclonal Antibody Combination against SARS Coronavirus: Synergy and Coverage of Escape Mutants. Burton DR, ed. *PLoS Med*. 2006;3(7):e237. doi:10.1371/journal.pmed.0030237

104. Hu H, Voss J, Zhang G, et al. A Human Antibody Recognizing a Conserved Epitope of H5 Hemagglutinin Broadly Neutralizes Highly Pathogenic Avian Influenza H5N1 Viruses. *J Virol*. 2012;86(6):2978-2989. doi:10.1128/JVI.06665-11

105. Zhu X, Guo YH, Jiang T, et al. A Unique and Conserved Neutralization Epitope in H5N1 Influenza Viruses Identified by an Antibody against the A/Goose/Guangdong/1/96 Hemagglutinin. *J Virol*. 2013;87(23):12619-12635. doi:10.1128/JVI.01577-13

106. Robbiani DF, Gaebler C, Muecksch F, et al. Convergent antibody responses to SARS-CoV-2 in convalescent individuals. *Nature*. 2020;584(7821):437-442. doi:10.1038/s41586-020-2456-9

107. Pinals RL, Ledesma F, Yang D, et al. Rapid SARS-CoV-2 Spike Protein Detection by Carbon Nanotube-Based Near-Infrared Nanosensors. *Nano Lett*. 2021;21(5):2272-2280. doi:10.1021/acs.nanolett.1c00118

108. Routledge I, Epstein A, Takahashi S, et al. Citywide serosurveillance of the initial SARS-CoV-2 outbreak in San Francisco using electronic health records. *Nat Commun*. 2021;12(1):3566. doi:10.1038/s41467-021-23651-6

109. Peluso MJ, Takahashi S, Hakim J, et al. SARS-CoV-2 antibody magnitude and detectability are driven by disease severity, timing, and assay. *Sci Adv*. 2021;7(31):eabh3409. doi:10.1126/sciadv.abh3409

110. Costello SM, Shoemaker SR, Hobbs HT, et al. The SARS-CoV-2 spike reversibly samples an open-trimer conformation exposing novel epitopes. *Nat Struct Mol Biol*. 2022;29(3):229-238. doi:10.1038/s41594-022-00735-5

111. Ravalin M, Roh H, Suryawanshi R, et al. A Single-Component Luminescent Biosensor for the SARS-CoV-2 Spike Protein. *J Am Chem Soc*. 2022;144(30):13663-13672. doi:10.1021/jacs.2c04192

112. Biering SB, Gomes de Sousa FT, Tjang LV, et al. SARS-CoV-2 Spike triggers barrier dysfunction and vascular leak via integrins and TGF-β signaling. *Nat Commun*. 2022;13(1):7630. doi:10.1038/s41467-022-34910-5

113. Madou MJ. Chapter 7. Scaling Laws. In: *Fundamentals of Microfabrication and Nanotechnology*. Vol 3. 3rd ed. CRC Press; 2011:p.1677-1704. https://doi-org.libproxy.berkeley.edu/10.1201/9781315274164-40

114. Das C, Mondal A, Bose A, Das B. A comprehensive literature review on microfluidic biochip technologies: techniques, challenges and future trends. *Microfluid Nanofluid*. 2026;30(1):8. doi:10.1007/s10404-025-02864-x

115. Alexandre-Franco MF, Kouider R, Kassir Al-Karany R, Cuerda-Correa EM, Al-Kassir A. Recent Advances in Polymer Science and Fabrication Processes for Enhanced Microfluidic Applications: An Overview. *Micromachines*. 2024;15(9):1137. doi:10.3390/mi15091137

116. Gurkan UA, Wood DK, Carranza D, et al. Next generation microfluidics: fulfilling the promise of lab-on-a-chip technologies. *Lab Chip*. 2024;24(7):1867-1874. doi:10.1039/D3LC00796K

117. Damir S, Delgado-Licona F, deMello A, Abolhasani M. Autonomous microfluidic labs: progress and prospects. *Lab Chip*. 2026;26(5):1014-1024. doi:10.1039/D5LC00908A

118. Niculescu AG, Chircov C, Bîrcă AC, Grumezescu AM. Fabrication and Applications of Microfluidic Devices: A Review. *IJMS*. 2021;22(4):2011. doi:10.3390/ijms22042011

119. Fernandes Quero R, Jesus DPD, Fracassi Da Silva JA. Simple modification to allow high-efficiency and high-resolution multi-material 3D-printing fabrication of microfluidic devices. *Lab Chip*. 2023;23(16):3694-3703. doi:10.1039/D3LC00356F

120. Deb A, Nagesh G, Harrison T, Spicer D, Ting DSK, Ahamed MJ. A review of commercial and laboratory-based microfluidic devices based on glass and/or silicon substrates. *RSC Adv*. 2026;16(14):12663-12681. doi:10.1039/D5RA08490C

121. Alonzo LF, Hinkley TC, Miller A, et al. A microfluidic device and instrument prototypes for the detection of *Escherichia coli* in water samples using a phage-based bioluminescence assay. *Lab Chip*. 2022;22(11):2155-2164. doi:10.1039/D1LC00888A

122. Clime L, Brassard D, Geissler M, Veres T. Active pneumatic control of centrifugal microfluidic flows for lab-on-a-chip applications. *Lab Chip*. 2015;15(11):2400-2411. doi:10.1039/C4LC01490A

123. Addae-Mensah KA, Cheung YK, Fekete V, Rendely MS, Sia SK. Actuation of elastomeric microvalves in point-of-care settings using handheld, battery-powered instrumentation. *Lab Chip*. 2010;10(12):1618. doi:10.1039/c002349c

124. Buchenauer A, Funke M, Büchs J, Mokwa W, Schnakenberg U. Microbioreactors with microfluidic control and a user-friendly connection to the actuator hardware. *J Micromech Microeng*. 2009;19(7):074012. doi:10.1088/0960-1317/19/7/074012

125. Zhang Y, Tseng TM, Schlichtmann U. Portable all-in-one automated microfluidic system (PAMICON) with 3D-printed chip using novel fluid control mechanism. *Sci Rep*. 2021;11(1):19189. doi:10.1038/s41598-021-98655-9

126. Zhu H, Özkayar G, Lötters J, Tichem M, Ghatkesar MK. Portable and integrated microfluidic flow control system using off-the-shelf components towards organs-on-chip applications. *Biomed Microdevices*. 2023;25(2):19. doi:10.1007/s10544-023-00657-z

127. Watson C, Senyo S. All-in-one automated microfluidics control system. *HardwareX*. 2019;5:e00063. doi:10.1016/j.ohx.2019.e00063

128. Morbioli GG, Speller NC, Cato ME, Stockton AM. An automated low-cost modular hardware and software platform for versatile programmable microfluidic device testing and development. *Sensors and Actuators B: Chemical*. 2021;346:130538. doi:10.1016/j.snb.2021.130538

129. Filatov NA, Denisov IA, Evstrapov AA, Bukatin AS. Open-Source Pressure Controller Based on Compact Electro-Pneumatic Regulators for Droplet Microfluidics Applications. *IEEE Trans Instrum Meas*. 2022;71:1-10. doi:10.1109/TIM.2022.3158383

130. Kong DS, Thorsen TA, Babb J, et al. Open-source, community-driven microfluidics with Metafluidics. *Nat Biotechnol*. 2017;35(6):523-529. doi:10.1038/nbt.3873

131. Soenksen LR, Kassis T, Noh M, Griffith LG, Trumper DL. Closed-loop feedback control for microfluidic systems through automated capacitive fluid height sensing. *Lab Chip*. 2018;18(6):902-914. doi:10.1039/C7LC01223C

132. White JA, Streets AM. Controller for microfluidic large-scale integration. *HardwareX*. 2018;3:135-145. doi:10.1016/j.ohx.2017.10.002

133. Da Costa ET, Mora MF, Willis PA, Do Lago CL, Jiao H, Garcia CD. Getting started with open-hardware: Development and control of microfluidic devices. *Electrophoresis*. 2014;35(16):2370-2377. doi:10.1002/elps.201400128

134. Park W, Choe S woon, Kim M. Microscale Flow Control and Droplet Generation Using Arduino-Based Pneumatically-Controlled Microfluidic Device. *Biosensors*. 2024;14(10):469. doi:10.3390/bios14100469

135. Kehl F, Cretu VF, Willis PA. Open-source lab hardware: A versatile microfluidic control and sensor platform. *HardwareX*. 2021;10:e00229. doi:10.1016/j.ohx.2021.e00229

136. Kumar K. R. N, Hamid S, Niketa AK, Prajapati E, Kumar S. Valved microfluidics with Ostemers. *Lab Chip*. Published online 2026:10.1039.D5LC00968E. doi:10.1039/D5LC00968E

137. Ezra E, Maor I, Bavli D, et al. Microprocessor-based integration of microfluidic control for the implementation of automated sensor monitoring and multithreaded optimization algorithms. *Biomed Microdevices*. 2015;17(4):82. doi:10.1007/s10544-015-9989-y

138. Nobs JB, Maerkl SJ. Long-Term Single Cell Analysis of S. pombe on a Microfluidic Microchemostat Array. Polymenis M, ed. *PLoS ONE*. 2014;9(4):e93466. doi:10.1371/journal.pone.0093466

139. Quintard C, Tubbs E, Achard JL, Navarro F, Gidrol X, Fouillet Y. Microfluidic device integrating a network of hyper-elastic valves for automated glucose stimulation and insulin secretion collection from a single pancreatic islet. *Biosensors and Bioelectronics*. 2022;202:113967. doi:10.1016/j.bios.2022.113967

140. Zhou J, Ren K, Dai W, Zhao Y, Ryan D, Wu H. Pumping-induced perturbation of flow in microfluidic channels and its implications for on-chip cell culture. *Lab Chip*. 2011;11(13):2288. doi:10.1039/c0lc00466a

141. Chen Y, Tian Y, Xu Z, Wang X, Yu S, Dong L. Microfluidic droplet sorting using integrated bilayer micro-valves. *Applied Physics Letters*. 2016;109(14):143510. doi:10.1063/1.4964644

142. Kortbawi H, Mandell-Brehm C, O'Donovan B, et al. DeRisi Lab Phage Immunoprecipitation Sequencing (PhIP-Seq) v1. Preprint posted online September 13, 2023. doi:10.17504/protocols.io.4r3l229qxl1y/v1

143. Jachimska B, Wasilewska M, Adamczyk Z. Characterization of Globular Protein Solutions by Dynamic Light Scattering, Electrophoretic Mobility, and Viscosity Measurements. *Langmuir*. 2008;24(13):6866-6872. doi:10.1021/la800548p

144. Bodansky A, Ahyong V, Dayao M, et al. AEGIS reveals epitope- and clone-resolved convergence of CNS B and T cell autoreactivity in ROHHAD. *Immunology*. Preprint posted online May 11, 2026. doi:10.64898/2026.05.06.722823

145. Rackaityte E, Borges B, Kortbawi HM, et al. Evidence for Maternal Autoantibodies in the Pathogenesis of Preterm Birth. *Obstetrics and Gynecology*. Preprint posted online October 4, 2024. doi:10.1101/2024.10.03.24314850

146. Rosenberg-Hasson Y, Hansmann L, Liedtke M, Herschmann I, Maecker HT. Effects of serum and plasma matrices on multiplex immunoassays. *Immunol Res*. 2014;58(2-3):224-233. doi:10.1007/s12026-014-8491-6

147. Valekova I, Jarkovska K, Kotrcova E, et al. Revelation of the IFNα, IL-10, IL-8 and IL-1β as promising biomarkers reflecting immuno-pathological mechanisms in porcine Huntington's disease model. *Journal of Neuroimmunology*. 2016;293:71-81. doi:10.1016/j.jneuroim.2016.02.012

148. Matsuda KM, Yoshizaki A, Yamaguchi K, et al. Autoantibody Landscape Revealed by Wet Protein Array: Sum of Autoantibody Levels Reflects Disease Status. *Front Immunol*. 2022;13:893086. doi:10.3389/fimmu.2022.893086

149. Amiral J. Generation and pathogenicity of autoantibodies associated to thrombosis and hemostasis. *Explor Immunol*. Published online February 5, 2024:34-58. doi:10.37349/ei.2024.00127

150. Elliott C, Lindner M, Arthur A, et al. Functional identification of pathogenic autoantibody responses in patients with multiple sclerosis. *Brain*. 2012;135(6):1819-1833. doi:10.1093/brain/aws105

Appendix A.
Protein Purifier Build Guide

A.1 Overview

The Autopurifier is a low-cost benchtop-sized programmable protein purification instrument that was designed and built by the Bioengineering team at the Chan Zuckerberg Biohub San Francisco. It serves as an intermediate solution between the low volume throughput but high

parallelizability of centrifuge-based purification methods, and the high volume throughput, but low parallelizability, of common commercial instruments.

The Build Guide is not an all-encompassing document, but it will provide sufficient information for assembling the instrument and putting it into an operational state. While the system is designed to support a high degree of flexibility through its modular design, details regarding extending functionality and manipulating low level software is out-of-scope of this document. Parties interested in exploring these advanced topics are encouraged to review the Github repository containing the software for the system: https://github.com/czbiohub/ProteinPurifier

A.2 Manufacturing Custom Parts

Prior to assembling the system, it is strongly recommended that all custom parts are prepared ahead of time as this step will take the most time. All custom parts were either 3D printed with an Ultimaker or laser cut with a ULS system. The raw material for the laser cut panels was too wide for our system and required preprocessing with a bandsaw. In addition, some holes in 3D printed parts were intentionally undersized and then drilled out to ensure proper dimensions. This in particular applied to the fraction collector shaft coupler and the fraction collector limit switch holder. Many holes required tapping to support M2, M3, M4 and M5 screws and it is advised that those following the assembly guide refer to the specific taps that are outlined in the part documents found in Onshape:

https://cad.onshape.com/documents/768143c17dda5be636f2c7b2/w/652fdaeeba2c62cf8303fe3e/e/caf0bd7f2f19e0ffc6ed1810

The following parts were laser cut:

1. Component Panel - 3-0508 - ¼” acetal copolymer
2. Shelf - 3-0509 - ⅜” acetal copolymer
3. Fraction Collector Base Spacer - 3-0523 - ¼” acetal copolymer
4. Fraction Collector Base Plate - 3-0524 - ⅜” acetal copolymer
5. Fraction Collector Tube Holder, 1.5mL - 3-0526 - ¼” acetal copolymer
6. VESA Mounting Plate Adapter - 3-0656 - ¼” acetal copolymer

The following parts were 3D printed:

1. Burkert Solenoid Valve Holder - 3-0510
2. Mount for 9QX Peristaltic Pump - 3-0057
3. 1mL Column Holder - 3-0511
4. 5mL Column Holder - 3-0512
5. Column Mount Base - 3-0513
6. Tubing Mount Base - 3-0514
7. Tubing Clamp - 3-0515
8. Arduino Shield Mount - 3-0516
9. 3mm Spacer for PCB - 3-0517
10. Meanwell PSU Cover - 3-0518
11. Fraction Collector Stepper Mounting Plate - 3-0519
12. Fraction Collector Stepper Spacer - 3-0520
13. Fraction Collector Stepper Shaft Coupler - 3-0521
14. Fraction Collector Limit Switch Mount- 3-0522
15. Fraction Collector Tube Holder Spacer - 3-0525
16. Fraction Collector Flow Through Stabilizer - 3-0528

Required Tools

- Drill and drill bits for 3D printed parts
- Dremel for shortening shaft of rotary valve stepper motor (or use another stepper)
- Mill for a cutout on the aluminum extrusions supporting the fraction collector

- Various metric taps for laser cut and 3D printed parts
- Metric allen keys
- Laser cutter
- 3D printer
- Saw or bandsaw for trimming stock to fit in the laser cutter
- Soldering station
- Wire strippers
- PEEK tubing cutter
- Micro USB cable for programming Tic stepper drivers

A.3 Assembly

The system consists of several modules including an 8-to-1 rotary valve for selecting buffers, peristaltic pumps for controlling the flow rate of each column, solenoid valves for controlling fluidic flow paths, and a fraction collector for capturing the sample and column flow through. Figure C-1 presents a simplified representation of how the modules interface. The following sections will cover the mechanical and electrical assembly of all modules and will briefly touch upon software operation and control. Part numbers will be used extensively here and will refer to the part as listed in the bill of materials in the following **Appendix B**.

A. 4 Component Panel

The component panel is the core of the system and hosts columns and reagent inputs on one side of the panel and a majority of the fluidics and electronics on the other side. It consists of several submodules including:

- Rotary valve
- Peristaltic pumps
- Solenoid valves
- Power and communication electronics

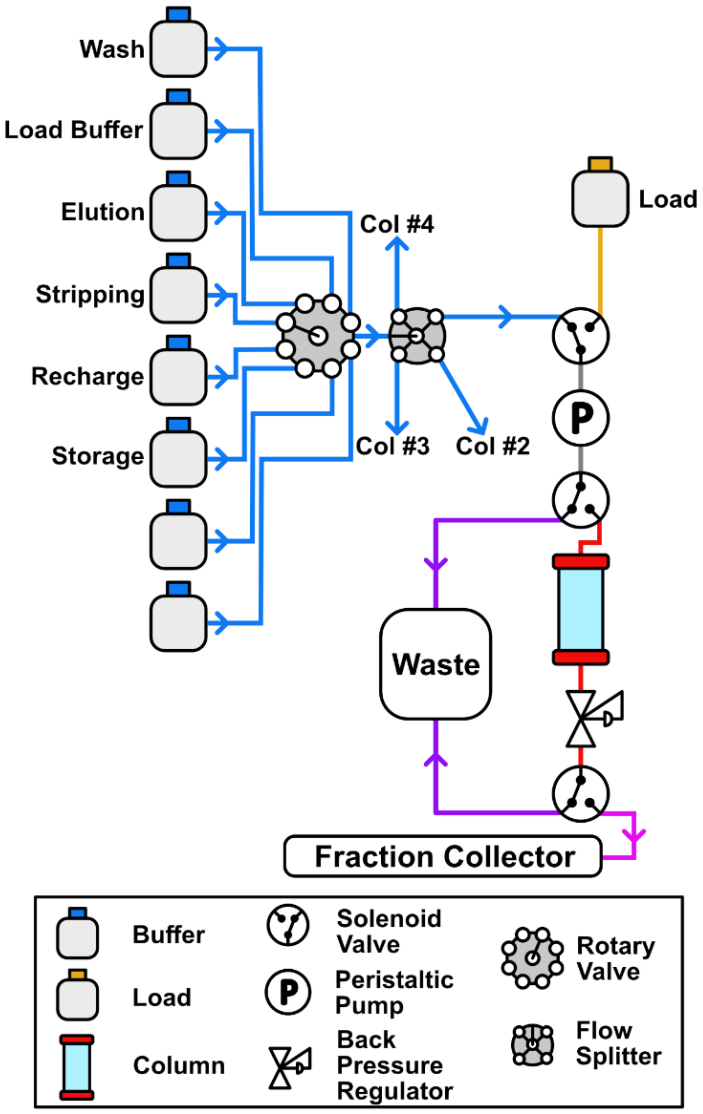


Figure A-1. Fluidic flow path for a single column.
From top left to bottom right: Up to (8) buffers connect to a rotary valve and are subsequently split into (4) fluidic paths, one path per column. Each path starts with an input valve that can select either a common buffer or a designated load. The flow rate of the inputs is controlled by a peristaltic pump. Before the column is an output valve that will either pass the input directly to waste for the purposes of purging air out of the lines or pass the input into the column. Following the column is an output valve that will either pass the flowthrough to waste or send it to the fraction collector.

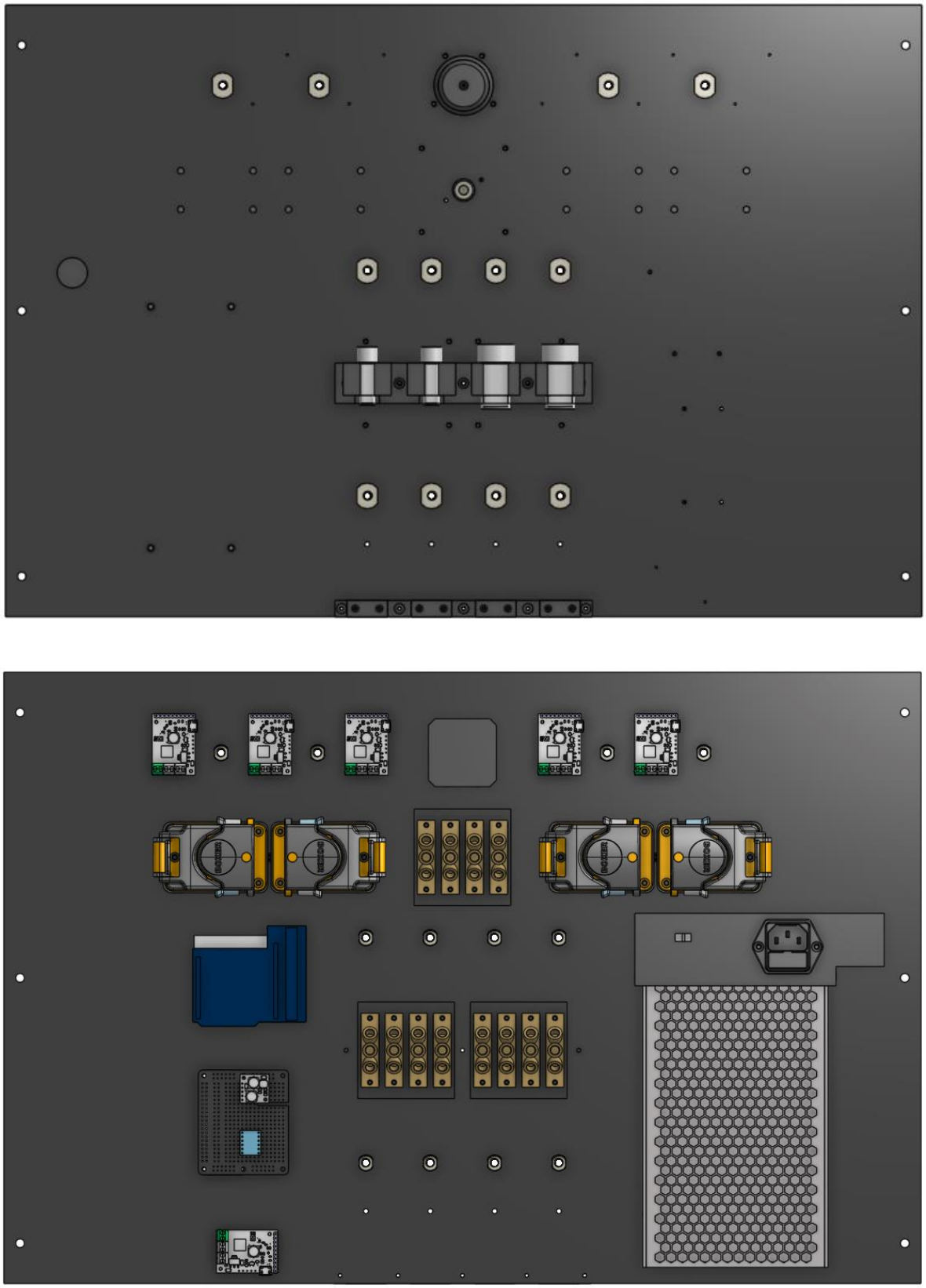


Figure A-2. Front and Back Views of the Component Panel.

A.5 Rotary Valve

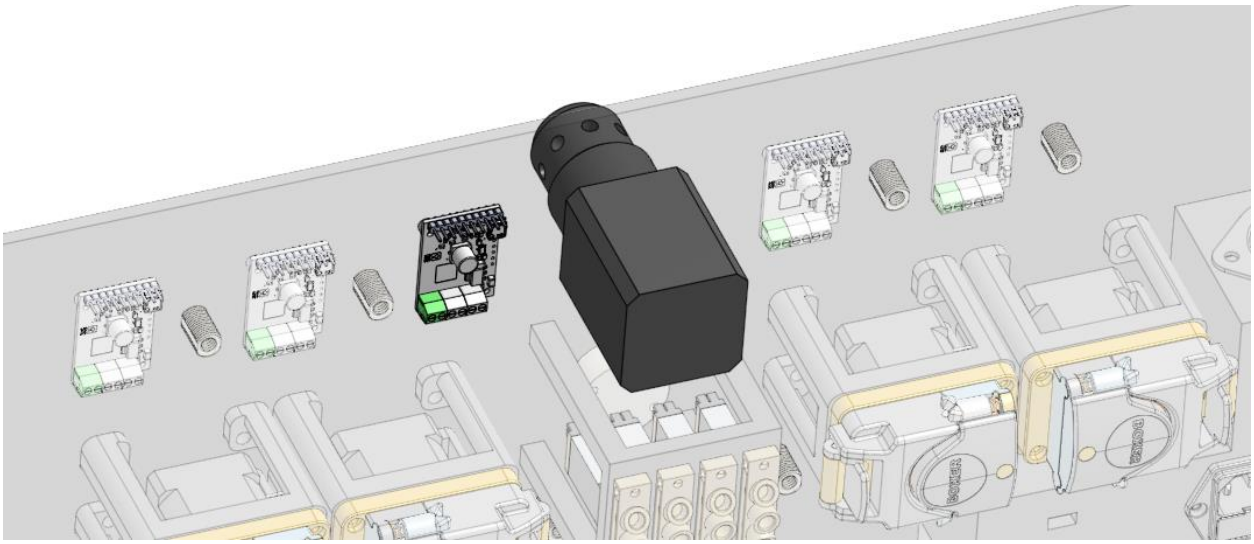


Figure A-3. Rotary Valve and Tic Stepper Driver.

The rotary valve consists of a Fluidic High Technology ERV001-08S14, a NEMA 17 17HS19-2004S1 stepper motor, a Pololu 3135 T500 stepper driver, and custom 3-0517 3mm spacers (3D printed). Begin by shortening the NEMA 17 motor shaft with a Dremel such that the shaft can be clamped by the rotary valve shaft coupler, but does not obstruct assembly of the two components. Fasten the two components together.

Prepare the Pololu Tic driver by soldering a screw terminal to the VIn and GND through holes. Solder wires to the SDA and SCL I2C communication through holes. Solder the motor coil wires to A1, A2, B1, B2 through holes. Refer to the datasheet for identifying which color wire corresponds to which motor coil. Reversing coil wires 1 and 2 does not matter as long as it is a single coil. Reversing wires will cause the motor to drive in the opposite direction, but that is easily corrected by configuring the driver firmware. Solder the encoder wires to the Tic driver.

White -> GND; Orange -> 5V; Yellow -> RX; Brown -> TX.

Using the Tic Software from Pololu (https://www.pololu.com/product/3135/resources), connect to the board via microUSB and apply these settings:

- I2C address: 0x10
- Tic current : 1909 mA
- Tic step size : ⅛
- Max velocity : 800 pulses / s
- Max acceleration : 400 pulses / s^2
- Max deceleration : 2000 pulses / s^2
- Homing direction: Reverse
- Tx pin: User Input, Analog
- Rx pin: Limit Switch Reverse
- SDA: Default
- SCL: Default

Homing represents either a reverse or forward limit switch. It is recommended to use a reverse limit switch and confirm that the motor can rotate freely in the forward direction without triggering the limit switch. If it triggers the limit switch, apply a setting to invert motor direction.

Using two 3-0517 spacers, fasten the Tic driver to the component panel. Fasten the rotary valve as well.

A.6 Solenoid Valves

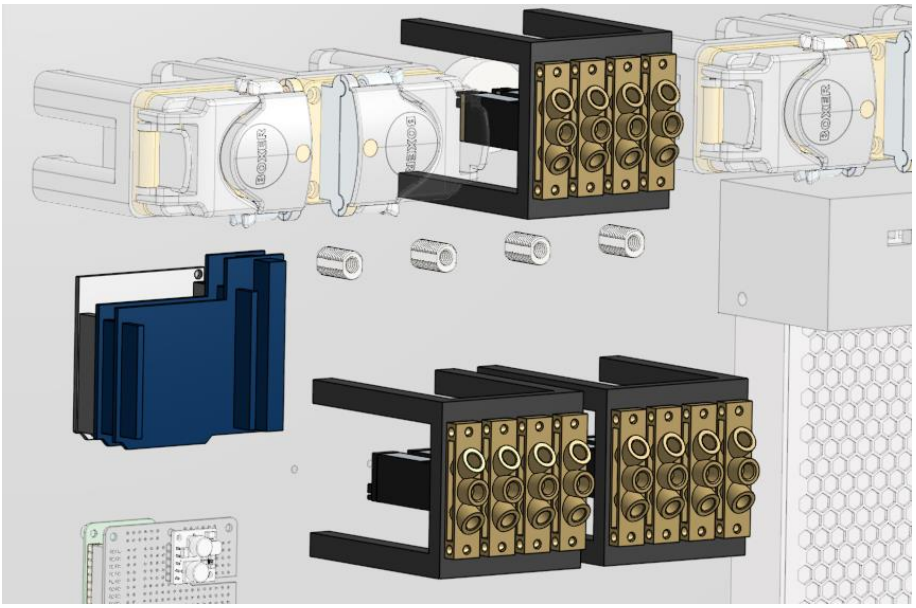


Figure A-4. Relay Drivers and Solenoid Valve Modules.

The solenoid modules consist of Burkert 299250 rocker solenoid valves, Burkert Type 2503 cables, 3-0510 solenoid valve mounts (3D printed), Freetronics RELAY8 relay drivers, and a 3-0516 Arduino shield mount (3D printed). Begin by attaching the Type 2503 cables to the solenoid valves and fasten solenoids to the solenoid holders. Install Idex 5-port manifold P-154 underneath the upper solenoid module. Secure the solenoid holders and the Arduino shield mount to the component panel.

Prepare the relay drivers by cutting the PULLUP traces on both boards and jumpering the "LINK DC to VIN" header. Attach one of the drivers to the shield mount and connect the lower two solenoid valve modules to the relay driver. The valve furthest from the driver should occupy position 1 on the driver and the closest should occupy position 8. Stack the second relay driver, bridge the A0 address header with a jumper, and repeat for the upper solenoid module. Screw a few inches of 18AWG wires into the relay power terminals. Connect a pair of jumper wires to the SDA and SCL I2C bus pins such that one end of each wire remains unconnected.

- I2C address - Reagent Input (Bottom) : 0x20
- I2C address - Column Input/Output (Top) : 0x21

A.7 Peristaltic Pumps

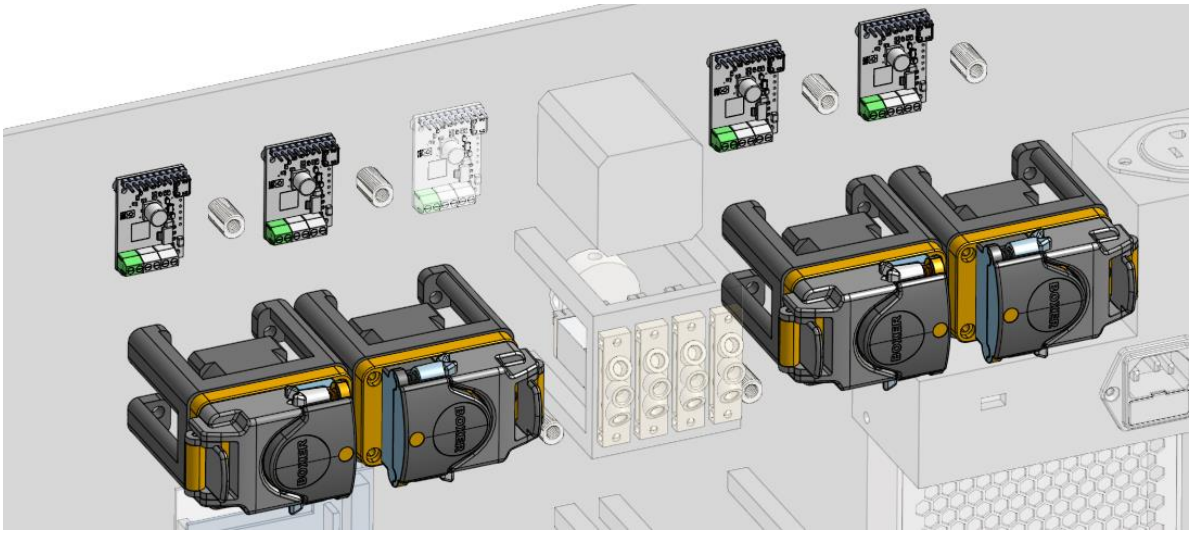


Figure A-5. Tic Stepper Drivers and Peristaltic Pump Modules.

Each peristaltic pump module consists of a Boxer Pump 9QX, 3-0057 mount for 9QX peristaltic pump (3D printed), a Pololu 3135 T500 stepper driver, and 3-0517 3mm spacers (3D printed).

Prepare the Pololu Tic driver by soldering a screw terminal to the VIn and GND through holes. Solder wires to the SDA and SCL I2C communication through holes. Solder the motor coil wires to A1, A2, B1, B2 through holes. Refer to the datasheet for identifying which color wire corresponds to which motor coil.

Using the Tic Software from Pololu (https://www.pololu.com/product/3135/resources), connect to the board via microUSB and apply these settings:

- I2C address - Pump 1 (closest to power button) : 0x30
- I2C address - Pump 2 : 0x31
- I2C address - Pump 3 : 0x32

- I2C address - Pump 4 : 0x33
- Tic current : 495 mA
- Tic step size : 1/4
- Max velocity : 1000 pulses /s
- Ma acceleration : 5000 pulses / s
- SDA: Default
- SCL: Default

Fraction Collector Stepper Driver

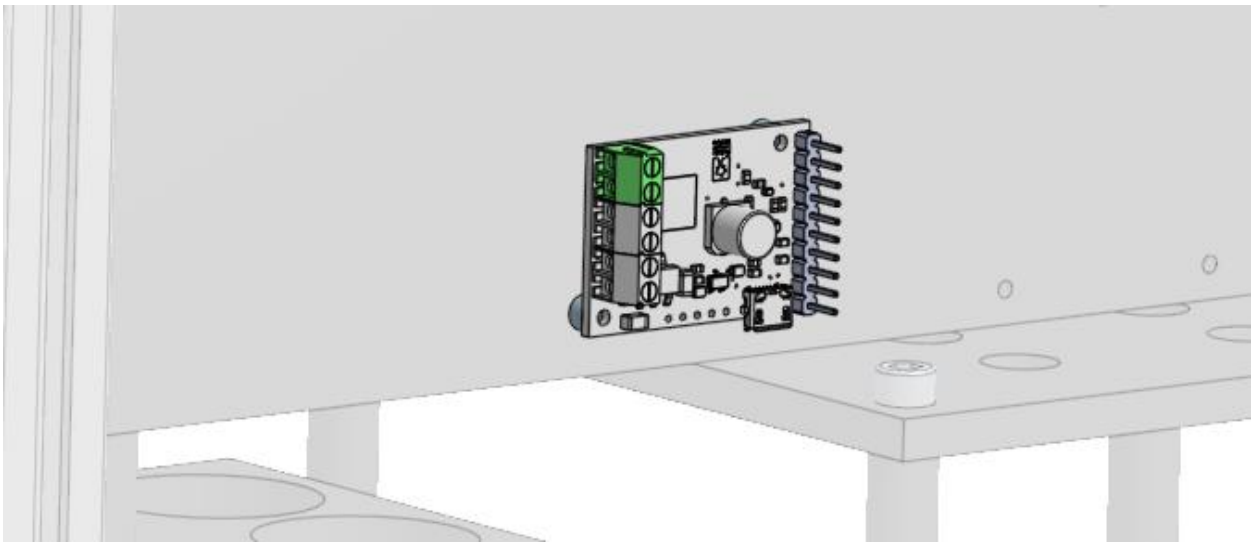


Figure A-6. Fraction Collector Stepper Driver.

Prepare the Pololu Tic driver by soldering a screw terminal to the VIn and GND through holes. Solder wires to the SDA and SCL I2C communication through holes. Solder the motor coil wires to A1, A2, B1, B2 through holes. Refer to the datasheet for identifying which color wire corresponds to which motor coil.

Using the Tic Software from Pololu (https://www.pololu.com/product/3135/resources), connect to the board via microUSB and apply these settings:

- I2C address : 0x40
- Motor current : 634mA

- Microsteps : ¼
- Max Velocity : 5000 pulses / s
- Max Acceleration : 5000 pulses / s^2
- Homing Speed Towards: 2000 pulses / s
- Homing Speed Away: 1000 pulses / s
- Homing Direction : Forward
- RC - Limit Switch Forward

  Routing Tubing, Rear

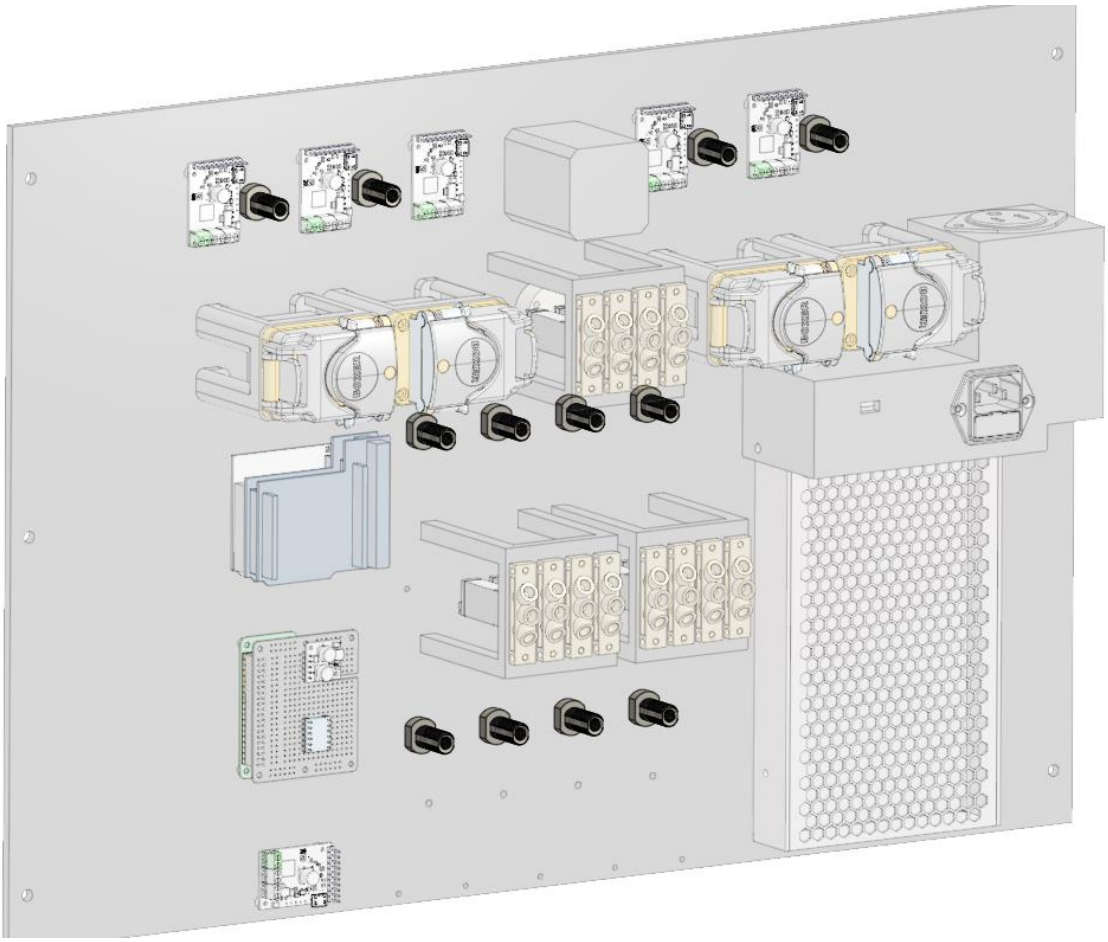


Figure A-7. Bulkhead Union Locations on Component Panel.

When routing capillary tubing throughout the system, a few things should be considered. PEEK capillary tubing should be cleaved with a PEEK cutter. These cutters are designed to prevent the end of the tubing from being crushed during the cutting process and virtually guarantees

unobstructed flow through the tubing. The cutters also provide a blunt end cut that helps maintain a flush seal during installation. In addition, it is a good practice to ensure that the fluidic lines cut across the (4) flow paths are equal in length to maintain identical flow resistances and output timings.

Begin by temporarily unmounting the upper solenoid valve module and installing 1532L PEEK tubing on the 5-port manifold using the nuts and ferrules provided by the manifold. Once the nuts have been hand tightened to secure the tubing, reinstall the solenoid module. Route the PEEK tubing from the 5-port manifold to the "NO" ports of the solenoid valves above them. All PEEK tubing interfacing with the solenoid modules or the bulkhead unions will use P-201 Idex flangeless fittings.

Install Idex P-441N bulkhead unions onto the component panel. Route PEEK tubing from the upper bulkhead unions to the "NC" ports of the upper solenoid valves.

Route PEEK tubing from the "NC" ports of the solenoid module closest to the relay driver to the middle row of bulkhead unions.

Route PEEK tubing from the "C" ports of the solenoid module closest to the power supply to the lower bulkhead unions. Route PEEK tubing from the "NC" ports of the same module to the front panel by passing through the through holes immediately below the lower bulkhead unions.

Install waste line PFA tubing to the "NO" ports of both lower solenoid modules. The free end of the tubing will ultimately send flowthrough to a waste bottle. Ensure that the lines have sufficient length to allow the waste bottle to be repositioned as needed. 2ft per line should be sufficient. Clear PFA is recommended to assist users in identifying which lines are pulling in air during setup without damaging the column.

Join the “C” ports of the upper solenoid manifold to the “C” ports of the lower solenoid manifold closest to the relay driver with Boxer 9000.535 peristaltic pump tubing and Idex P-668 hose barbs. Ensure that the tubing length is sufficient to be routed through the peristaltic pump.

A.8 Routing Tubing, Front

Tubing should be routed on the front side of the panel once it has been mounted to the chassis to prevent damage to the components during handling.

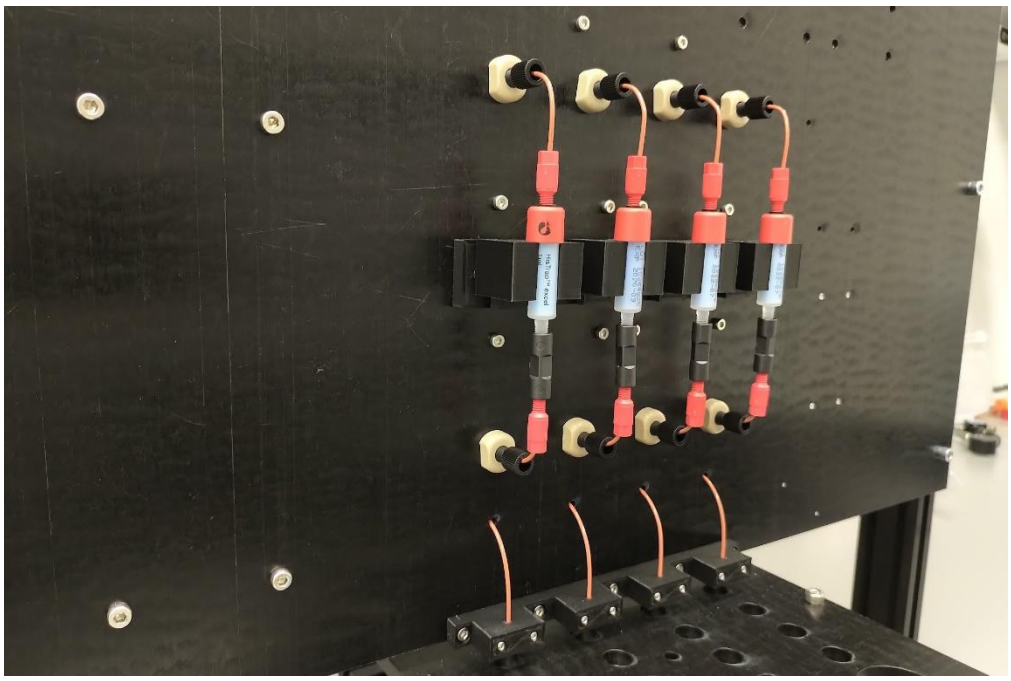


Figure A-8. Components and Tubing for Columns.

Begin by fastening the 3-0513 column mount base (3D printed) and the 3-0514 tubing clamp base (3D printed) to the component panel. Insert columns into the 3-0511 or 3-0512 column holders (3D printed) and place them on the dovetail mount. Fasten one end of Idex 1532L PEEK tubing to the bulkhead connector with Idex P-201 flangeless fittings above a column and connect the other end to column immediately below using the connectors provided with the column. Repeat for the bulkhead connector immediately below the column.

Pass PEEK tubing from the “NC” ports of the solenoid module closest to the power supply through the through holes below the lower bulkhead unions. Clamp the protruding PEEK tubing with the 3-0515 tubing clamps such that ~3mm of tubing extends beyond the lower edge of the component panel.

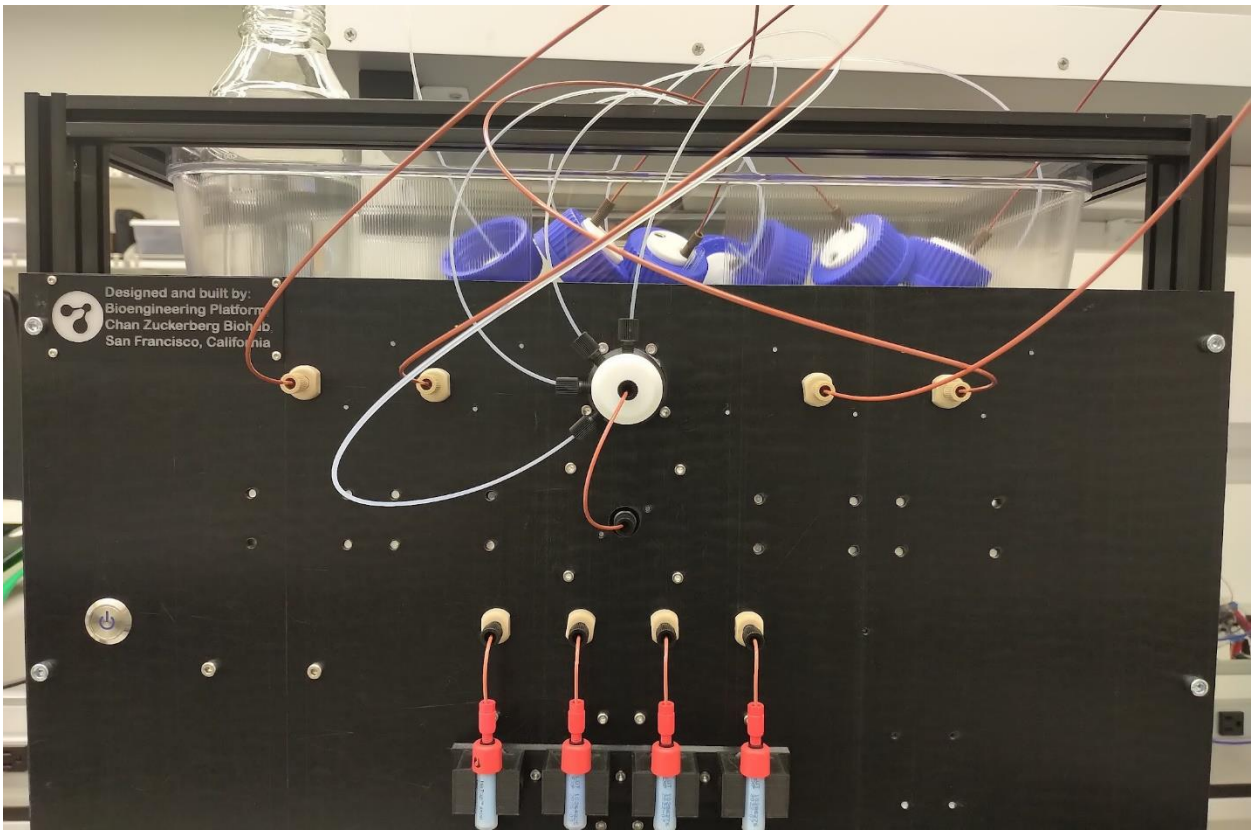


Figure A-9. Tubing Connections for Buffers and Loads.

Connect 2 ft of 1532L PEEK or 1507L PFA tubing to the bulkhead unions and the ports on the rotary valve with Idex P-201 flangeless fittings. The other end can be attached to the Cole-Parmer EW-12018-00 GL45 bottle caps. The large inner diameter PFA tubing is recommended if the user intends to operate multiple columns at a flow rate of 5 mL/min. Take care to tighten the fittings such that they are snug, but not overly tight. The rotary valve is composed of PTFE which is a very low friction material unlike all the other connectors on the panel. If the PTFE port is tightened too much or the nut is inserted at a slight angle, it will strip the threads and damage the

port. If the port is tightened too little, air will be pulled into the lines during operation. The tightness can be safely evaluated and adjusted while pumping a buffer to waste during setup.

A.9 Electronics

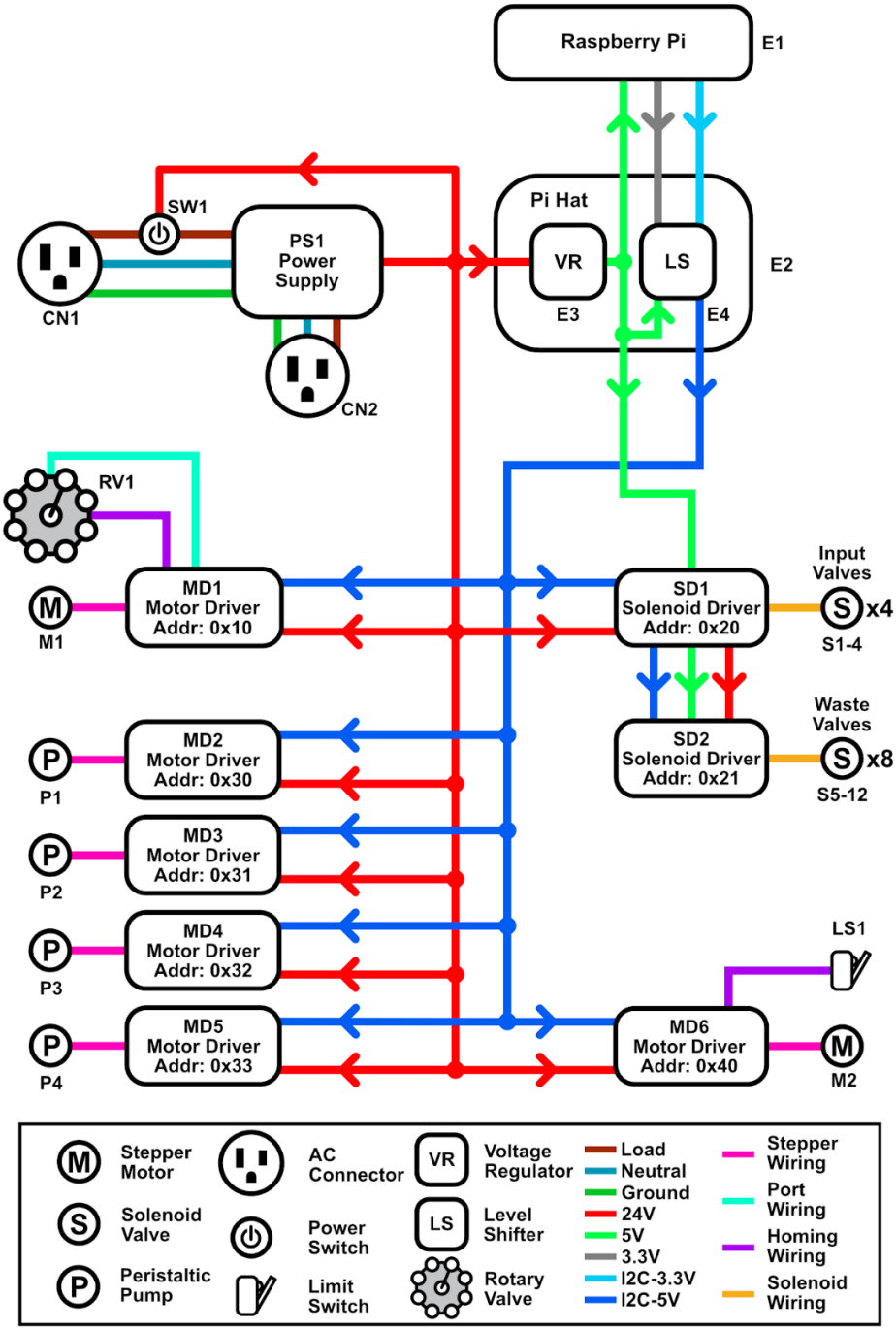


Figure A-10. Detailed View of Electrical Connections.

All electronic Pololu Tic stepper drivers, Freetronics relay drivers, and the Raspberry Pi will tap into a 24V rail that is routed from the PSU to the fraction collector Tic board using Amazon B07114RK67 T-tap wire connectors. Similarly, all Tic boards and Freetronics relay drivers will tap into the Raspberry Pi I2C lines using T-tap wire connectors.

A.10 Raspberry Pi Hat

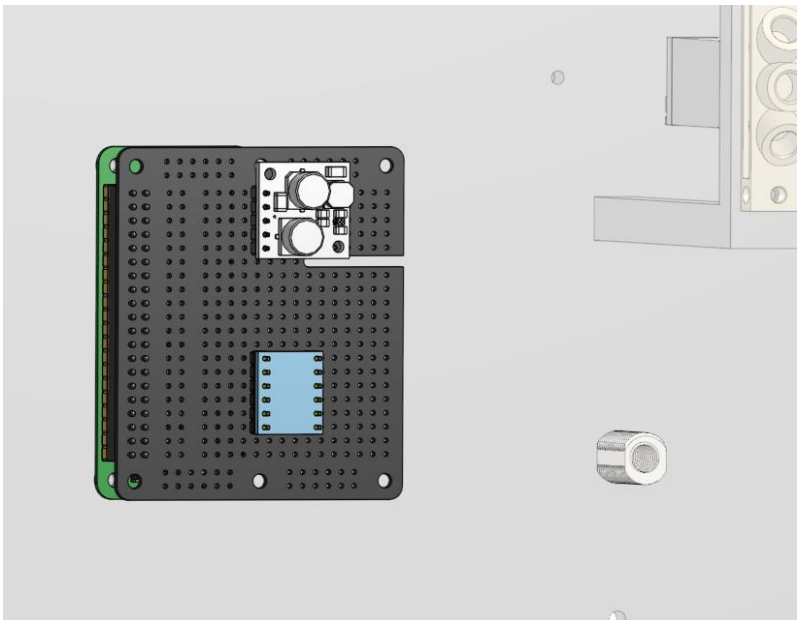


Figure A-11. Raspberry Pi Module with Protoboard Hat and Power and Level Shifter ICs.

The purpose of the B01MY8TYSD protoboard is to provide space for the step down converter and the bidirectional level shifter. The step down converts the 24V power rail to 5V to supply the Raspberry Pi Zero W, the Freetronics RELAY8 IC, and the B07LG646VS bidirectional level shifter. The bidirectional level shifter provides an interface between the Raspberry Pi and all the other peripherals.

Solder header pins to the step down converter and the bidirectional level shifter before soldering both to the protoboard. Using wire jumpers, solder the 5V output of the step down

converter to a power rail on the protoboard. Repeat for the GND input. Solder a few inches of 18AWG wire to the 24V and GND input of the step down converter. These will later be connected to power lines with T-tap connectors.

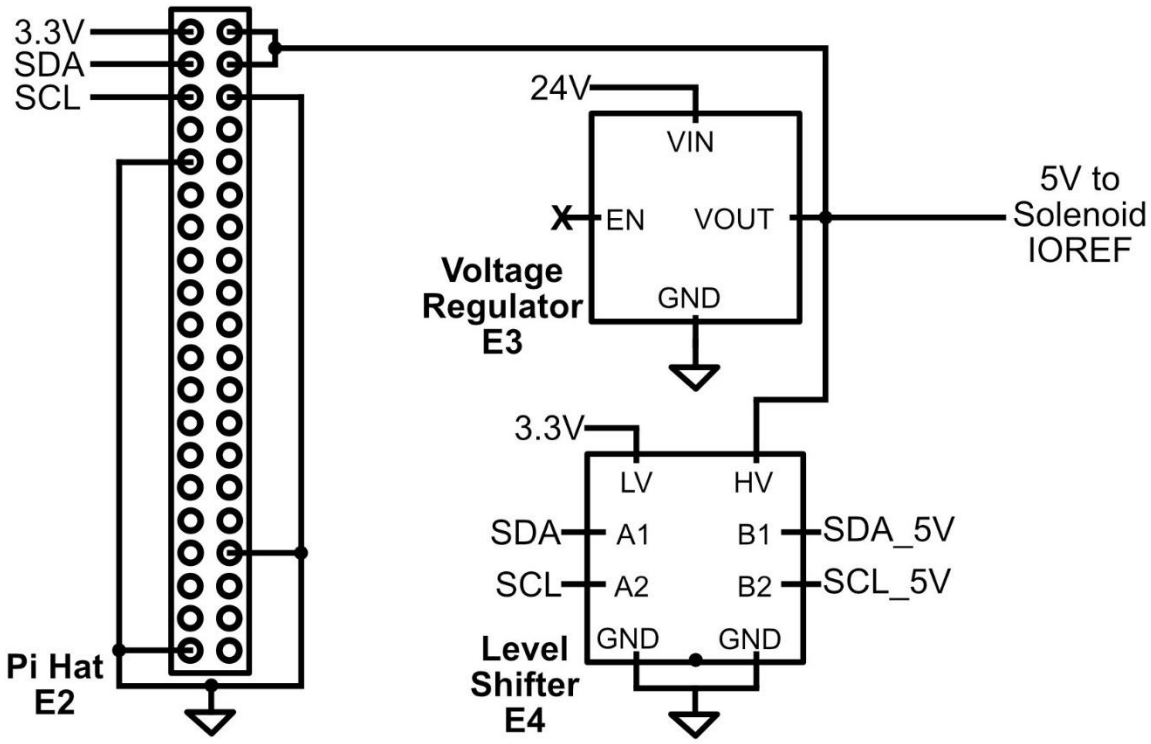


Figure A-12. Schematic for Raspberry Pi Hat.

Solder jumper wires from the 5V power rail to the 5V input of the Raspberry Pi and to the high side supply of the bidirectional level shifter. Solder jumper wires from the GND power rail to the GND input of the Raspberry Pi and the GND input of the high and low side of the bidirectional level shifter. Solder a jumper wire from the 3V3 output of the Raspberry Pi to the low side supply of the bidirectional level shifter. Solder a jumper wire to the 5V rail and connect it to the IOPWR pin on the RELAY8 board.

Solder jumper wires from Pin 16 and Pin 20 of the Raspberry Pi to two available buffer pins on the low side of the bidirectional level shifter. Pin 16 will be the I2C data line (SDA) and

Pin 20 will be the I2C clock line (SCL). Solder a few inches of 20-24 AWG wire to the corresponding buffer lines on the high side.

A.11 I2C Bus

Connect Amazon B07LG646VS T-tap connectors to the I2C lines on all peripherals. The pair of wires extending from the boards should be clamped into the stem of the connector. Run a pair of 20-24 AWG wires from the fraction collector Tic stepper driver to the furthest peristaltic pump stepper driver such that all peripherals can access the lines. Connect the branches of the T-tap connector such that all devices share the same SDA and SCL lines. Trim off excess wire that extends beyond the stepper drivers.

A.12 Power Electronics

Begin by preparing the power supply module and soldering 18 AWG wires to the 719W-00/03 AC receptacle and cover exposed leads with heat shrink. Insert a 5ST 2.5-R fuse into the receptacle and mount it to the 3-0518 PSU cover (3D printed). Solder the wire extending from the “L” terminal of the AC receptacle to the “NO” terminal of the AV1911P724Q04 power switch. Solder a wire to the “C” terminal of the power switch and screw it into the “L” terminal of the 1866-3332 Meanwell power supply. Screw the “N” and “GND” wires from the AC receptacle to the corresponding terminals on the power supply. Solder wires to the LED “+” and “-” terminals of the power switch and screw them into “V+” and “V-” terminals of the power supply. Screw a pair of 18 AWG to the unused set of “+” and “-” terminals of the power supply and feed the wires through the cutout on the power supply cover.

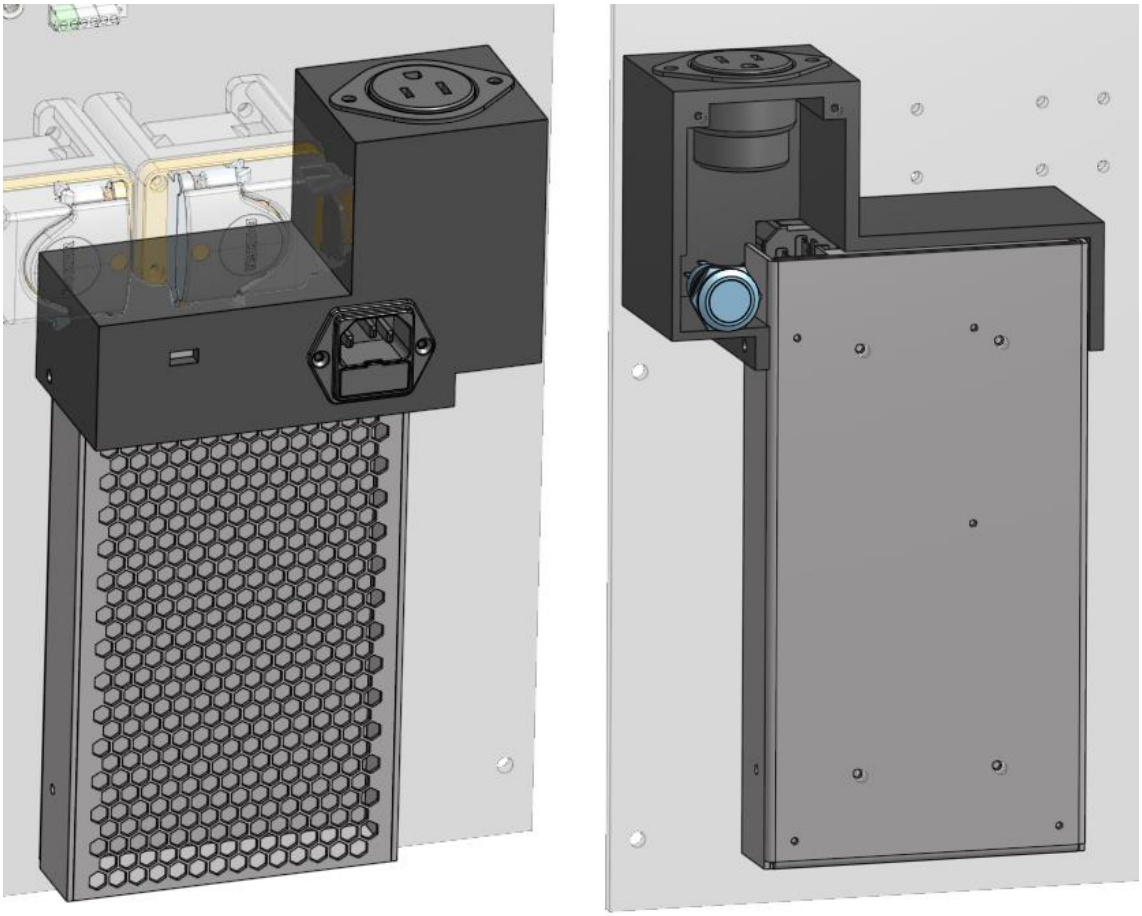


Figure A-13. Power Supply with AC Connectors and Panel Switch.

Solder 18 AWG wires to the 1866-3332-ND AC receptacle dedicated to the touch screen and screw them into the corresponding terminals on the Meanwell power supply.

Mount the power supply and power switch to the component panel. Fasten the power supply cover to the power supply such that it sits flush with the component panel and power supply. Wires may need to be adjusted.

Connect B07LG646VS T-tap connectors to the power lines on all peripherals. Run the pair of 18 AWG wires across the component panel such that all peripherals are able to tap into the power lines ensuring that all devices share a common 24V and GND line. Trim off excess wires that extend past the fraction collector Tic stepper driver.

A.13 Chassis

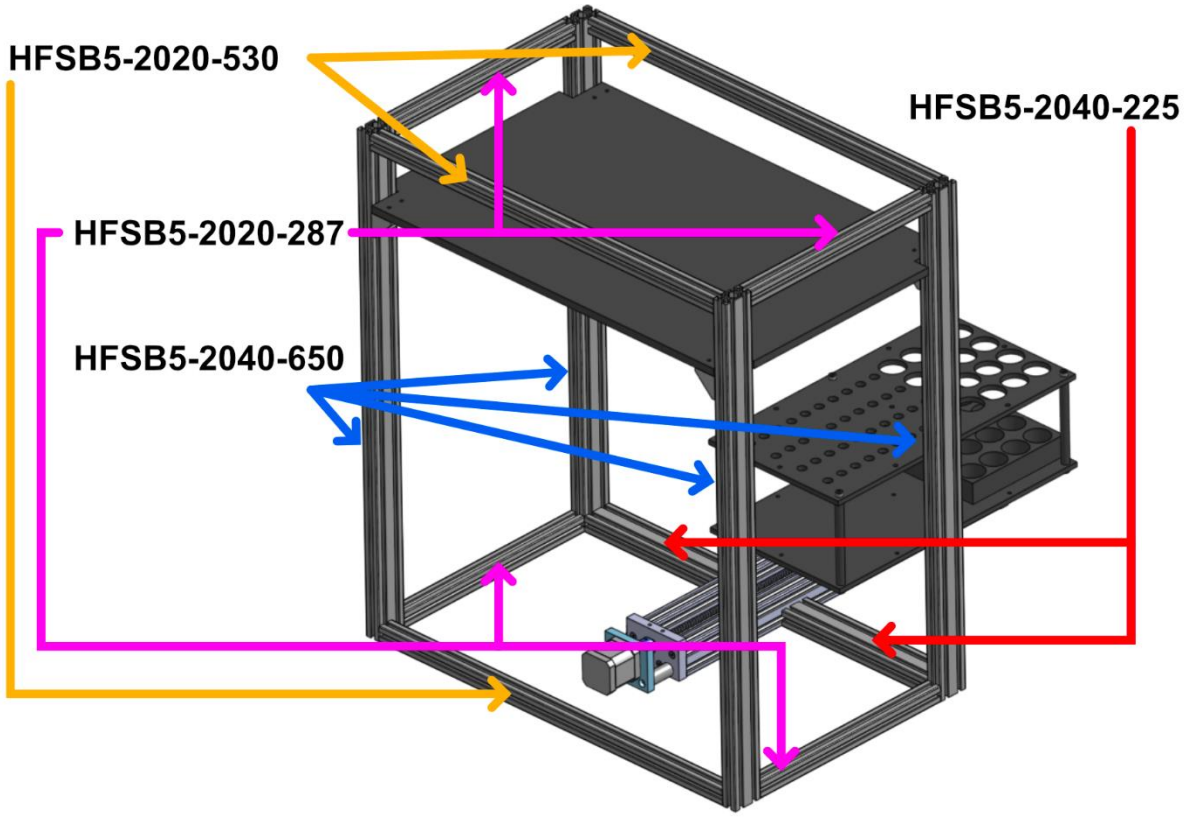


Figure A-14. Chassis with Shelf and Tube Rack Mounted on Fraction Collector.

The chassis consists of aluminum extrusions, right angle blind joints to join the extrusions, a Delrin shelf, right angle brackets to support the shelf, and the fraction collector.

Begin by assembling the top of the chassis on a flat surface to assist with squaring the structure. The top section is composed of the following Misumi components: (4) HFSB5-2040-650, (2) HFSB5-2020-530, (2) HFSB5-2020-287, and (8) HBLPBS5.

Mount (4) Misumi HBLFSDK5_b right angle brackets to the inner face of the 2040 extrusions while ensuring that the flat face of the bracket faces the top end of the chassis. They will be held in place with Misumi HNTTSN5-5 insertion nuts, 2 per bracket.

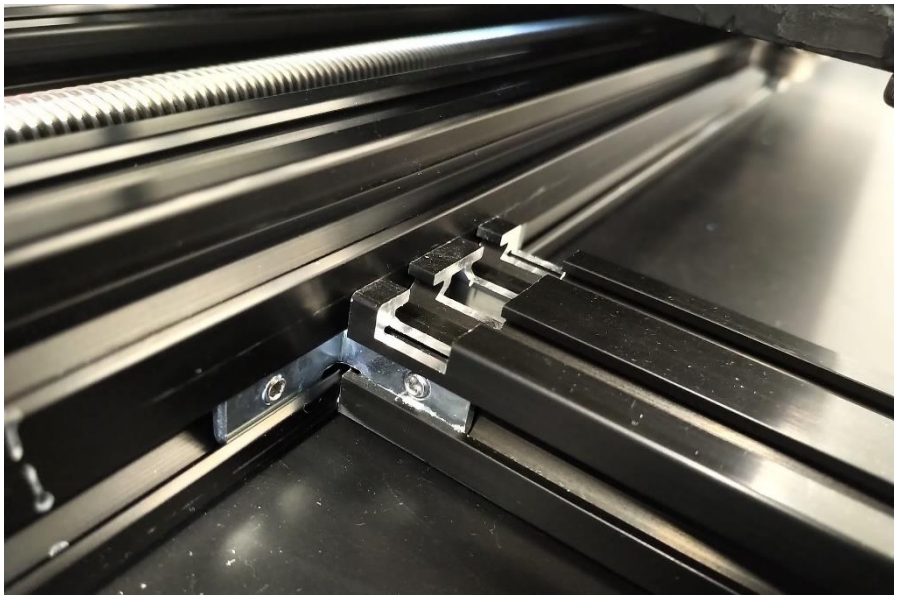


Figure A-15. Cutout on the Extrusions Supporting the Fraction Collector.

Mill a ½” wide, 3/16” deep cutout one the wide face of the two Misumi HFSB5-2040-225 extrusions such that the cut is offset ¼” from the end of the extrusion that abuts the fraction collector stage. Once milled, fasten the extrusions along with (2) Misumi HFSB5-2020-287 and (1) HFSB5-2020-530 to the bottom of the chassis.

Insert the 3-0509 shelf above the brackets and fasten. The final height of the shelf can be set once the McMaster-Carr 6686T42 polycarbonate storage container has been placed on the shelf. Adjust the height so that the container is level and the top lip is flush with the top edge of the aluminum extrusions.

Mount the component panel once the fraction collector has been installed.

A.14 Fraction Collector

The fraction collector is an OpenBuilds 2495-Bundle actuator that has been refitted with a NEMA 17 stepper motor and a tube rack. It is recommended to follow their assembly guide with the following exceptions:

- Before assembling the fraction collector, loosely fasten the aluminum extrusion to the chassis using (4) Misumi HBLPBS5 blind joints.
- Before assembling the fraction collector, loosely fasten the 3D printed 3-0522 limit switch mount near the motor side of the extrusion.
- Replace the provided shaft coupler with the 3D printed 3-0521 shaft coupler
- Replace the provided motor mount with the 3D printed 3-0519 mounting plate
- Replace the provided motor spacers with the 3D printed 3-0520 spacers

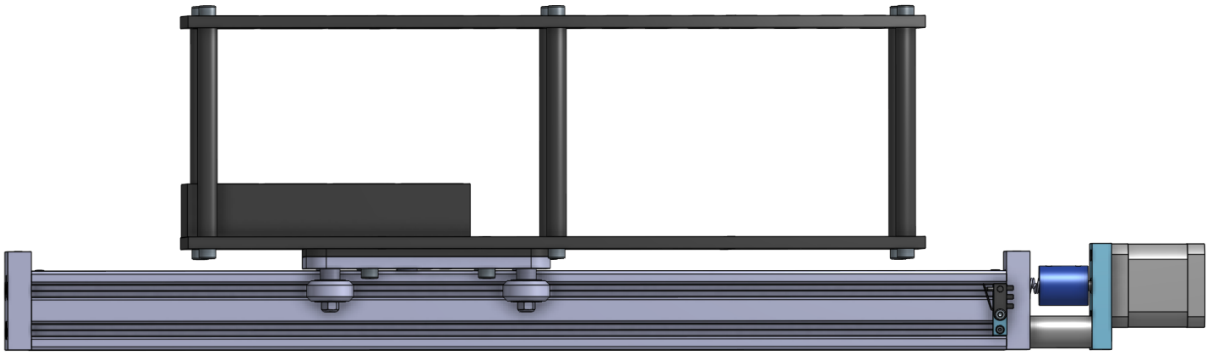


Figure A-16. Side View of the Fraction Collector with a Tube Rack and Spacer.

With the actuator assembled, begin assembling the tube rack by joining the 3-0524 base plate (laser cut) and the 3-0526 tube holder plate (laser cut) with 3-0525 spacers (3D printed). Once both plates are firmly secured, mount the 3-0528 flow through collection tube stabilizer (3D printed) to the base plate. With the tube rack complete, mount it to the fraction collector carriage with the 3-0523 spacer (3D printed) between the carriage and the base plate of the tube rack.

Relocate the fraction collector on the chassis until a tube in the position closest to the component panel can be safely retrieved when the fraction collector carriage is ¼" away from the motor-less end of the actuator. Fasten the fraction collector to the chassis.

Slide the limit switch mount until it reaches the end plate on the motor side of the actuator. Fasten it to the extrusion and attach a Digikey D2HW-C261H microswitch. Solder wires to the "NC" and "C" terminals of the microswitch and to the RC and GND terminals of the fraction collector Tic stepper driver. Fasten the wires to the chassis to ensure that they remain in a safe location at all times.

A.15 Touchscreen Interface

The touchscreen interface is mounted to the side of the chassis using a McMaster-Carr 3136N455 display mount and Misumi HNTTSN5-5 insertion nuts. Once attached, bolt the VESA mounting plate adapter 3-0656 (laser cut) to the touchscreen display and to the display mount. Connect the HDMI and USB cables of the display to the Raspberry Pi.

A.16 Software

The software is divided into three sections: preparing the OS on the SD card, changing the default account on log in, and installing the purifier python package.

A.16.1 Preparing the OS

The OS for the Raspberry Pi Zero W will be modified so that it will connect to the network once it has booted and use software implemented I2C since the hardware implementation is faulty.

1. Download Raspbian Lite OS from the Raspberry Pi Foundation (https://www.raspberrypi.org/downloads/raspbian/)
2. Unzip, rename and mount the .img of the OS to modify the image
3. Using terminal, navigate to the /boot/ volume and create an empty file called **ssh** to enable remote ssh access
4. Using terminal, create a file titled **wpa_supplicant.conf** and edit the following information to reflect your network:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
    ssid="<Your network name/SSID>"
    psk="<Your WPA/WPA2 security key>"
    key_mgmt=WPA-PSK
}
```

5. Using terminal, update the device tree overlay in **config.txt** to include the following statement at the end of the file:

   5a. dtoverlay=i2c-gpio,bus=5,i2c_gpio_sda=2,i2c_gpio_scl=3

6. Create a username and password (only for OS version bullseye and above)

   6a. Copy an encrypted password from the following command:

   echo 'mypassword' | openssl passwd -6 -stdin

   6b. Paste the password into a file titled userconf.txt in the following format

   username:encrypted-password

7. Close terminal and unmount the /boot/ volume.
8. Download and install an SD card burning software such as BalenaEtcher (https://www.balena.io/etcher/).
9. Insert an SD card into your computer
10. Open the etcher program, choose the modified .img file, select the SD card to burn and start flashing
11. When the card has finished flashing and verifying, transfer it to your Pi Zero

A.16.2 Changing Default System Settings

1. Power your Pi Zero
2. Let the Pi Zero boot up and expand the file system. The first boot may take up to 90s. You'll know that it has completed booting when the LED is solid green for several seconds without blinking
3. SSH into the system

   3a. Command: ssh pi@raspberrypi.local

   3b. Password: raspberry

4. Change the default password, hostname, enable the I2C bus, and set timezone

   4a. Command: sudo raspi-config

   4b. Go through the menu and change appropriate settings. Hostname is found in "Network". I2C is found in "Interfacing Options"

5. When the device reboots log in as shown in Step 3, but replace raspberrypi with your new hostname. Also, use your newly set password

6. Change the SSH settings to prevent the connection from closing during use for up to 24 hours

   6a. sudo nano /etc/ssh/sshd_config

   ClientAliveInterval 120

   ClientAliveCountMax 720

A.16.3 Setting up the touchscreen

A.16.3.1 Install Desktop

1. SSH into the system

   1a. Command: ssh pi@raspberrypi.local

2. Install Xorg to make the GUI work:

   2a. sudo apt-get install --no-install-recommends xserver-xorg

   2b. If you want to launch the Xorg GUI from terminal: ** not required

   sudo apt-get install --no-install-recommends xinit

3. Raspberry Pi Desktop (RPD) GUI

   3a. sudo apt-get install raspberrypi-ui-mods

A.16.3.2 Enable Auto Login

1. Run: sudo raspi-config
2. Choose option 3: Boot Options
3. Choose option B1: Desktop / CLI
4. Choose option B4: Desktop AutoLogin
5. Select Finish, and reboot the pi.

A.16.3.3 Install on screen keyboard

1. Update packages

   1a. sudo apt update

   1b. sudo apt upgrade

2. Install keyboard

   2a. sudo apt install matchbox-keyboard

A.16.3.4 Add virtual keyboard toggle to the Taskbar

1. Add a bash script to /usr/bin to grab the id of the virtual keyboard as store as a bash variable

   1a. sudo nano /usr/bin/toggle-keyboard.sh

   1b. Paste the following in the script:

```
#!/bin/bash
PID=`pidof matchbox-keyboard`
if [ ! -e $PID ]; then
  kill $PID
else
 matchbox-keyboard &
fi
```

   1c. Ctrl-O, Enter, Ctrl-X

   1d. sudo chmod +x /usr/bin/toggle-keyboard.sh

2. Create a file for the taskbar to read to load the keyboard

   2a. sudo nano /usr/share/raspi-ui-overrides/applications/toggle-keyboard.desktop

   2b. Paste the following:

```
[Desktop Entry]
Name=Toggle Virtual Keyboard
Comment=Toggle Virtual Keyboard
Exec=/usr/bin/toggle-keyboard.sh
Type=Application
Icon=matchbox-keyboard.png
Categories=Panel;Utility;MB
X-MB-INPUT-MECHANISM=True
```

2c. Ctrl-O, Enter, Ctrl-X

3. Copy the default config file to the pi config user

   3a. cp /etc/xdg/lxpanel/LXDE-pi/panels/panel /home/pi/.config/lxpanel/LXDE-pi/panels/panel

4. Modify the pi config file:

   4a. nano /home/pi/.config/lxpanel/LXDE-pi/panels/panel

   4b. Scroll to the bottom and paste the following:

```
Plugin {
  type=launchbar
  Config {
    Button {
      id=toggle-keyboard.desktop
    }
  }
}
```

   4c. Ctrl-O, Enter, Ctrl-X

   4d. sudo reboot

Taken from: https://pimylifeup.com/raspberry-pi-on-screen-keyboard/

A.16.4 Installing the Protein Purifier Python Package

1. Update Debian remote repository list

   1a. sudo apt-get update

2. Install system-wide utilities and Python 3 packages

   2a. sudo apt-get install python3-pip python3-venv python3-pyqt5 git i2c-tools

3. Create a virtual environment for the project

   3a. python3 -m venv venv-purifier

4. Create alias to easily initialize the virtual environment

   4a. nano ~/.bash_aliases

      alias purify="source ~/venv-purifier/bin/activate"

   4b. source ~/.bash_aliases

5. Clone the repository

5a. git clone https://github.com/czbiohub/ProteinPurifier.git

6. Install the Python package and dependencies

   6a. Start the virtual environment: purify

   6b. Install wheel: pip install wheel

   6c. Install the package: pip install git+https://github.com/czbiohub/ProteinPurifier

   6d. For other install options, refer to the repository README

A.16.5 Running the Protein Purifier Software as a Service

1. Create the purifier launch script:

   1a. sudo nano /usr/local/bin/purifier.sh

   1b. Paste the following:

```
#!/bin/bash
/home/pi/venv-purifier/bin/python /home/pi/ProteinPurifier/scripts/device_setup.py
```

2. Set the file to be executable by root:

   2a. sudo chmod 744 /usr/local/bin/purifier.sh

3. Create the service configuration file

   3a. sudo nano /etc/systemd/system/purifier.service

   3b. Paste the following:

```
[Unit]
Description=purifier management
After = network.target

[Service]
Type=simple
Restart=always
ExecStart=/usr/local/bin/purifier.sh

[Install]
WantedBy=multi-user.target
```

4. Start the service

   4a. sudo systemctl start purifier

5. Confirm that the service is active and running
   5a. sudo systemctl status purifier
6. Enable the service to launch at boot
   6a. sudo systemctl enable purifier
7. If needed, the purifier service can be restarted at anytime
   7a. sudo systemctl restart purifier

A.17 Programming the System

Documentation regarding programing and operating the system can be found in the GitHub repository located at : https://github.com/czbiohub/ProteinPurifier

If the settings outlined above were used, the only configuration change that might need to be implemented is the fraction collector positions that are computed from the autopurifier_hardware.config file. If a custom tube rack is used, determine the offsets, and starting positions for the fractions and flow through tubes.

Start by sending the fraction collector to the “Safe” position as outlined in an example in the czpurifier/ui directory. If the carriage bumps into the end plates of the actuator when it homes or travels to the safe position, adjust the position of the limit switch.

If the standard tube rack is being used, move the fraction collector to the position “Frac1” and adjust the position of the actuator on the chassis such that the first set of tubes closest to the component panel are immediately below the outlet tubing.

The examples in czpurifier/ui and the scripts directories provide working examples for how to operate or script the machine. For the purposes of reliability, always SSH into the machine to operate it. It is possible to control the system over the network without SSHing, but there have been unexplained disconnects over long periods of time.

Appendix B.
Bill-of-Materials for Protein Purifier

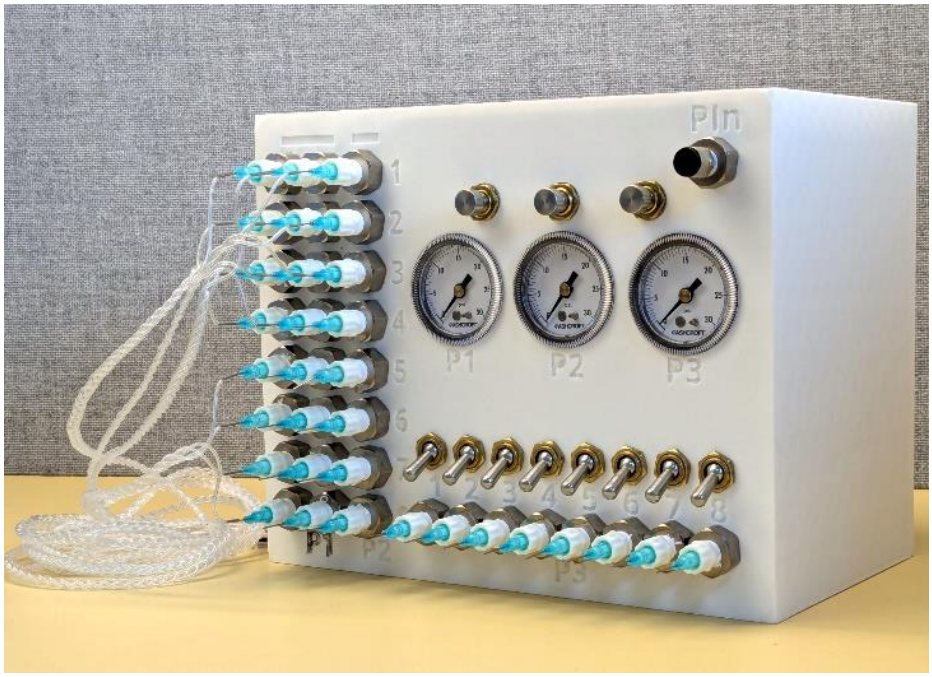
OTS – Off-the-shelf
Laser – Laser cut component
3D – 3D printed component

Item No.			Name	Qty	Vendor	Part number	Type
1			**Chassis**	**1**		**2-0103**	
	1.1		HFSB5-2040-650	4	Misumi	HFSB5-2040-650	OTS
	1.2		HFSB5-2040-225	2	Misumi	HFSB5-2040-225	OTS
	1.3		HFSB5-2020-530	3	Misumi	HFSB5-2020-530	OTS
	1.4		HFSB5-2020-287	4	Misumi	HFSB5-2020-287	OTS
	1.5		HBLFSDK5 b	4	Misumi	HBLFSDK5 b	OTS
	1.6		Blind Joints	22	Misumi	HBLPBS5	OTS
	1.7		Insertion Nuts	20	Misumi	HNTTSN5-5	OTS
2			**Component Panel**	**1**		**2-0096**	
	2.1		Component Panel	1	McMaster-Carr	3-0508	Laser
	2.2		ERV001	1	Fluidic HT	ERV001-08S14	OTS
	2.3		5 Port Manifold	1	Idex	P-154	OTS
	2.4		Bulk Head Union	12	Idex	P-441N	OTS
	2.5		**Burkert Selector Valves with Mount**	**3**		**2-0101**	
		2.5.1	HI_6724-299250 3/2-way Rocker Solenoid Valve with medium separation	4	Burkert	299250	OTS
		2.5.2	Burkert Solenoid Holder	1		3-0510	3D
		2.5.3	Burkert valve cable	4	Burkert	Type 2503	OTS
	2.6		**Boxer 9QX with Mount**	**4**		**2-0097**	
		2.6.1	Mount for Boxer 9QX - 300-000057	1		3-0057	3D
		2.6.2	9QX Stepper Pump	1	Boxer Pumps	9600.76	OTS
	2.7		**Column Mount**	**1**		**2-0094**	
		2.7.1	1mL Column Mount	2		3-0511	3D
		2.7.2	5mL Column Mount	2		3-0512	3D
		2.7.3	Column Mount Base	1		3-0513	3D
		2.7.4	1mL HiTrap	2	GE	17115101	OTS
		2.7.5	5mL HiTrap	2	GE	17115201	OTS
	2.8		**Tubing Mount**	**1**		**2-0093**	
		2.8.1	Tubing Mount Base	1		3-0514	3D
		2.8.2	Tubing Mount Clamp	4		3-0515	3D
	2.9		**Relay Driver Stack**	**1**		**2-0100**	
		2.9.1	Freetronics - RELAY8	2	Freetronics	RELAY8	OTS
		2.9.2	Arduino Shield Mount	1		3-0516	3D
	2.10		**Pi Zero with Populated Protoboard**	**1**		**2-0098**	
		2.10.1	Raspberry Pi 3B+	1	Amazon	8541589947	OTS

Item No.			Name	Qty	Vendor	Part number	Type
		2.10.2	Amazon B07LG646VS Bidirectional Level Shifter	1	Amazon	B07LG646VS	OTS
		2.10.3	2850 - V Reg with Headers	1	Pololu	2850	OTS
		2.10.4	Amazon - B01MY8TYSD - Pi Protoboard	1	Amazon	B01MY8TYSD	OTS
		2.10.5	3mm Spacer	4		3-0517	3D
		2.10.6	16GB SD Card	1	Amazon	SDSQUAR-016G-GN6MA	OTS
	2.11		**Tic with 3mm Spacer**	**6**		**2-0099**	
		2.11.1	Pololu tic-T500 stepper controller	1	Pololu	3135	OTS
		2.11.2	3mm Spacer	2		3-0517	3D
	2.12		**PSU**	**1**		**2-0092**	
		2.12.1	Meanwell PSU	1	Digikey	1866-3332-ND	OTS
		2.12.2	PSU Cover	1		3-0518	3D
		2.12.3	Fused AC Receptacle	1	Digikey	719W-00/03	OTS
		2.12.4	Fuse	1	Digikey	5ST 2.5-R	OTS
		2.12.5	AC Receptacle	1	McMaster	8036K500	OTS
		2.12.6	NEMA5-15P AC Power Cord	1	Digikey	AC30UNA	OTS
	2.13		Nema 17 Stepper Motor for ERV001	1	Amazon	17HS19-2004S1	OTS
3			**Fraction Collector**	**1**		**2-0095**	
	3.1		C-Beam XLarge Linear Acutator - 500mm - NO MOTOR	1	OpenBuilds	2495-Bundle	OTS
	3.2		Stepper Mounting Plate	1		3-0519	3D
	3.3		Stepper Spacer	2		3-0520	3D
	3.4		Nema 17 Stepper Motor	1	Amazon	17HS19-2004S1	OTS
	3.5		**Homing Switch**	**1**		**2-0242**	
		3.5.1	Limit Switch Mount	1		3-0522	3D
		3.5.2	Digikey D2HW-C261H micro switch	1	Digikey	D2HW-C261H	OTS
	3.6		**Tube Rack**	**1**		**2-0091**	
		3.6.1	Actuator Spacer	1		3-0523	Laser
		3.6.2	Tube Holder Base	1		3-0524	Laser
		3.6.3	50mL Conical Tube	1		89039-658	OTS
		3.6.4	1.5mL Tube Holder	1		3-0526	Laser
		3.6.5	Rack Spacer	6		3-0525	3D
		3.6.6	1.5mL Eppendorf Tube	1			OTS
		3.6.7	Flow Through Stabilizer	1		3-0528	3D
	3.7		Shaft Coupler	1		3-0521	3D
4			**Touchscreen Interface**				
	4.1		Display Mount	1	McMaster-Carr	3136N455	OTS
	4.2		VESA mounting adapter plate	1		3-0656	Laser
	4.3		Touchscreen display	1	Amazon	B082HWF2HW	OTS
5			Shelf	1		3-0509	Laser
6			Power Switch	1	Arrow Electronics	AV1911P724Q04	OTS

Item No.			Name	Qty	Vendor	Part number	Type
7			PEEK Tubing	50 ft	Idex	1532L	OTS
8			PFA Tubing	50 ft	Idex	1507L	OTS
9			Back Pressure Regulator	4	Idex	P-790	OTS
10			Peristaltic Pump Tubing	5m	Boxer Pumps	9000.535	OTS
11			PEEK Barb for Peristaltic Pump Tubing	8	Idex	P-668	OTS
12			Various 22-24 AWG Wire for Pi Protoboard and I2C Signal				OTS
13			Various 18 AWG Wire for Power Distribution				OTS
14			Flangeless Fittings	50	Cole Parmer	P-201	OTS
15			T-tap wire connectors	16	Amazon	B07114RK67	OTS
16			Bottle Storage Container	1	McMaster-Carr	6686T42	OTS
17			M2, M3, M4, M5 stainless steel socket head screws		McMaster-Carr		OTS
18			GL45 Bottle Cap with dual 1/4-28 Ports	12	Cole Parmer	EW-12018-00	OTS
19			Polycarbonate Storage Container	1	McMaster-Carr	6686T42	OTS

Appendix C.
Microfluidic Controller Build Guide



C.1 Overview

The following guide is for assembling and configuring the mechanical, pneumatic, electrical, and software subsystems of the microfluidic controller described by **Chapter 3**. Information regarding system operation can be found in an additional supplementary info from **"Appendix D. Microfluidic Controller Operation Guide"**. The system is modular and allows for alternative hardware or configurations to be used. Sections that are most likely to be modified by others will be denoted with **[*]** and additional guidance for modifications is provided. This appendix makes extensive references to component designators from the **"Appendix E. Bill-of-Materials for Microfluidic Controller"** and should be read in conjunction with that appendix. Figure C-1 is used to provide an abstract view of how the components of the BOM interface with one another. For assembly of the custom solenoid driver PCB PLRD1, please reference the KiCAD

documentation listed in the Data Availability statement of Chapter 3. The custom *plfluidic* Python 3 module and the configuration files for automating the compilation of the optional Linux image listed in the Data Availability statement of **Chapter 3**.

Required tools for assembly:

- Large build volume FDM 3D printer
- Soldering iron of heat-set inserts
- PTFE tape (1/4”) or thread sealant
- Plastic tubing cutter
- Adjustable wrench (up to 1”)
- A set of metric allen keys
- A set of miniature screw drivers
- Wire strippers
- Lab-proof permanent marker
- MicroSD USB adapter

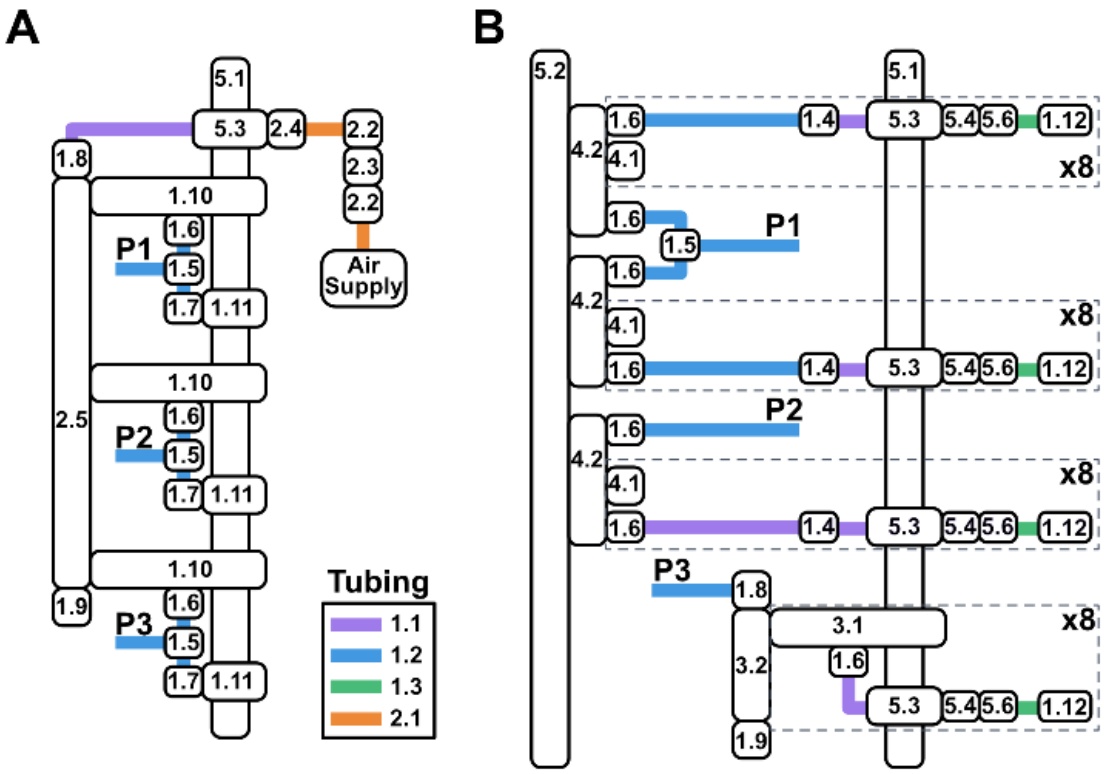


Figure C-1. Bill of materials component diagram.
An outline of the mechanical connections between the pneumatic components with reference designators obtained from the BOM. Tubing is indicated by colored lines, mechanical connections are indicated with tangential edges, and panel mounts are represented with overlaid objects. Subfigure A describes how the pressure source is split and down regulated to three separate pressures. Subfigure B describes how those pressures are routed to manifolds which are then passed to the interface. The dashed grey bounding boxes indicate that the assemblies are repeated multiple times on the associated manifold but are not represented for clarity.

C.2 Enclosure Fabrication

Prior to assembling the pneumatic controller, the enclosure body (BOM 5.1) and rear panel (BOM 5.2) should be 3D printed. The source files for creating the enclosure can be found in the Data Availability statement in **Chapter 3**. The materials and parameters for printing are flexible as long as a thermoplastic is used so that heat-set threaded inserts can be inlaid into the chassis. The system described here used an Ultimaker S5 FDM printer for its large build volume with PLA filament, 1.2mm wall thickness and a 20% infill. After printing was completed and supports were stripped, heat-set threaded inserts (BOM 5.6) were heated with a soldering iron set to 200 °C and

pressed into the body and rear panel. The body requires (4) inserts where the rear panel mates with the body. The rear panel requires (8) inserts: (2) for each of the (3) solenoids and an additional (2) for the relay controller PCB mount.

[*] The enclosure can be produced by on-demand manufacturing service such as Protolabs or for those who do not have large build volume 3D printers. All that is required is the enclosure design files. If solenoids valves other than SMC S070 or Pneumadyne 10 mm are used, then the rear panel of the enclosure will need to be modified to support the valves being used.

C.3 Pneumatic Assemblies

The pneumatic connections used in this system include threaded components in addition to push-to-connect and barbed fittings for tubing. Threaded components do not create a perfect seal, so it is recommended to either wrap PTFE tape around the male threaded component or lightly apply liquid sealant to the threads. When joining tubing to push-to-connect fittings, it is also recommended to use tubing that has been cut with a blunt end to ensure a secure seal. An outline of the pneumatic system is provided in Figure C-2.

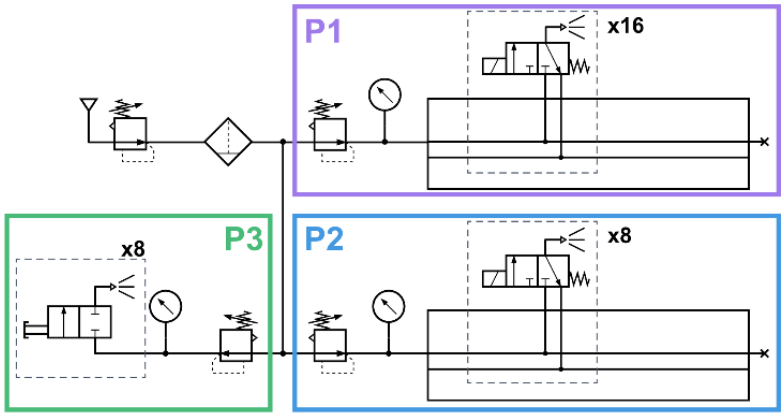


Figure C-2. Outline of the pneumatic subsystem.
The microfluidic controller accepts regulated and filtered compressed air from an external source, feeds it to a distribution manifold where it is split into 3 separate regulated pneumatic circuits. P1 is a circuit supporting set of 16 solenoid valves, P2 is a circuit supporting an additional 8 solenoid valves, and P3 is a circuit supporting 8 toggle switches for reagent pressurization.

C.3.1 Regulated Distribution Manifold

This manifold serves to split the filtered and regulated input pressure into three separate pressures for operating the downstream pneumatic functions. Begin by threading the panel mount regulators (BOM 1.10) into the 3-port manifold (BOM 2.5). Once all regulators have been seated, slightly adjust their orientation such that the output port of the regulator is perpendicular to the long edge of the manifold. Thread the barbed tubing adapters (BOM 1.6) into each of the regulators. Thread the manifold plug (BOM 1.9) into one end of the manifold and the manifold barbed fitting (BOM 1.8) into the other end. Slightly adjust the orientation of the barbed fitting so that it is pointing towards the regulators.

Cut (3) pieces of flexible PVC tubing to roughly 15 cm and place them on the barbed regulators. For each of the pressure gauges (BOM 1.11) thread on the 1/8 NPT to barb adapters (BOM 1.7). Cut (3) pieces of flexible PVC tubing to roughly 3 cm in length and place the tubing on each of the NPT to barb adapters. Connect the other end of the tubing with the short arm of the barbed T connectors (BOM 1.3).

Mount the push-to-connect to 1/4 NPT fitting (BOM 5.3) to the port on the enclosure body labeled “Pin”. Next, mount each of the pressure gauges to the enclosure body. The 10-32 nuts and mounting bracket on the rear of the gauge will need to be removed and then replaced once the face of the gauge has been passed through the panel. Afterwards, mount the manifold with the panel mount regulators to the enclosure body. The assembly should be oriented such that the barbs on the regulators point towards the top of the enclosure body.

Cut an appropriate amount of firm EVA tubing (BOM 1.1) to join the push-to-connect fitting placed at “Pin” to the barb on the regulated distribution manifold. Thread the 1/4 NPT push-

to-connect fitting (BOM 2.4) onto the other side of the panel mount fitting placed at PIn. Join the tubing at the output of each regulator to the barbed T connector of the closest pressure gauge.

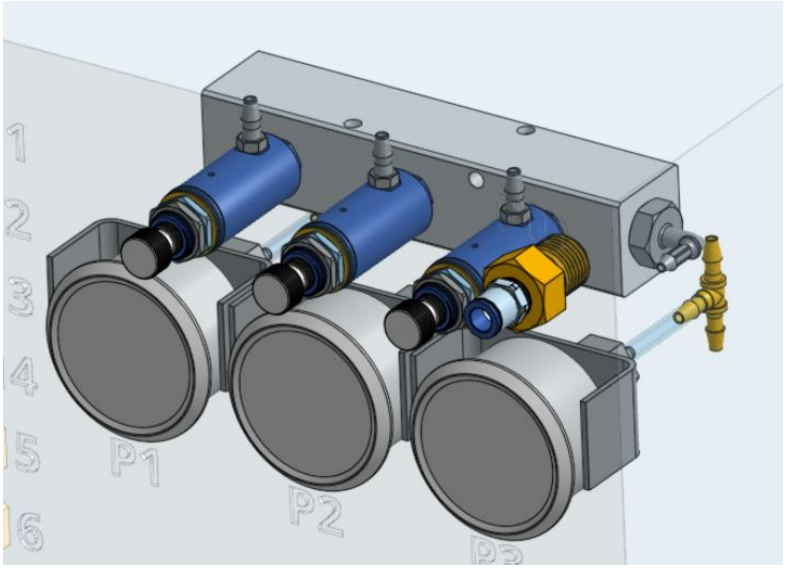


Figure C-3. Regulated distribution manifold with panel mount push to connect fitting and pressure gauges.

Tubing routed from regulator outputs to gauges not shown.

C.3.2 Input Reagent Manifold

This manifold is described in Figure C-2. It includes the components encompassed identified as P3 in the figure and is used to pressurize reagents that are used in microfluidic experiments. Begin by threading the 2-way toggle switches (BOM 3.1) onto the 8-port manifold (BOM 3.2). Slightly adjust the orientation of the toggle switches such that the outlet ports are perpendicular to the same long edge of the manifold. Thread the manifold plug (BOM 1.9) into one end of the manifold and the manifold barbed fitting (BOM 1.8) into the other end. Attach 15 cm of firm EVA tubing (BOM 1.1) to each of the barbs.

Mount (8) push-to-connect to 1/4 NPT fittings (BOM 5.3) in the row of cutout holes near the bottom of the enclosure body. Mount the manifold to the enclosure body just above the push-

to-connect fittings. The manifold is mounted using the panel mount toggle switches. Orient the manifold so that barbs and tubing are pointing towards the top of the enclosure. After mounting, connect each of the switches with the push-to-connect fitting directly below the manifold. Cut an appropriate length of EVA tubing to connect the manifold with the T barbed fitting attached to the P3 pressure gauge from the regulated pressure manifold.

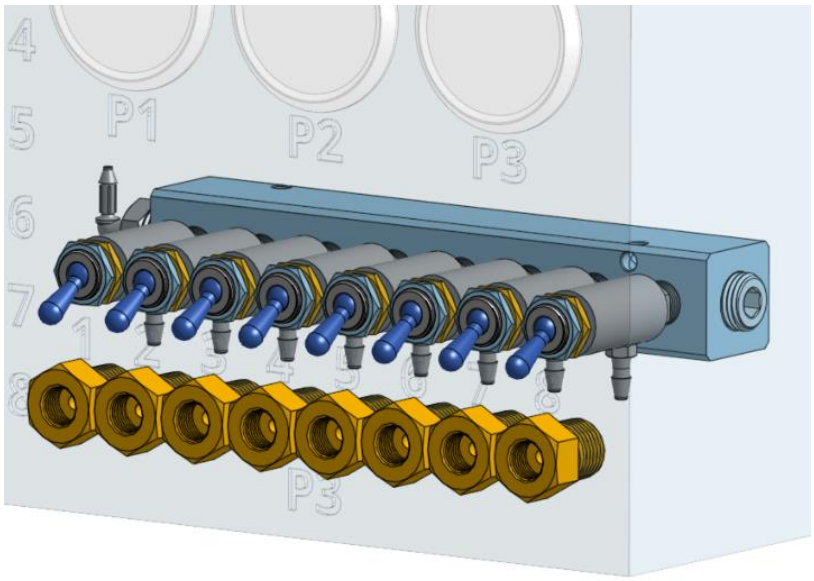


Figure C-4. Input reagent manifold.
Tubing routed from the toggle switch outputs to the panel mount push-to-connect fittings not shown.

C.3.3 Solenoid Manifold Assemblies

There are (3) separate solenoid manifolds that should be assembled identically. Attach (8) solenoid valves (BOM 4.1) to the solenoid manifold (BOM 4.2) using the included hardware. Thread 10-32 barbed tube fittings (BOM 1.6) to each solenoid output port on the manifold. Thread an additional barbed tube fitting to the input port of the manifold. Cut (8) pieces of flexible PVC tubing (BOM 1.2) to 45 cm in length and fit each one onto a solenoid output barb. On the other

end of each tubing, insert a straight barbed tubing fitting (BOM 1.4) and attach a piece of firm EVA tubing (BOM 1.3) that is 3 cm in length.

Mount solenoid manifold assemblies to the top left, top right, and bottom right of the enclosure rear panel (BOM 5.2) using 4-40 screws (BOM 5.7). Join the top solenoid manifold assemblies by connecting the input barbed fittings to a barbed T fitting (BOM 1.5) with flexible PVC tubing. Connect the barbed T fitting joining the manifolds with the barbed T fitting connected to the pressure gauge on the regulated distribution manifold labeled P1 with flexible PVC tubing. Connect the input barbed fitting of the lower right solenoid manifold assembly to the barbed T fitting connected to the pressure gauge on the regulated distribution manifold labeled P2.

Mount (24) push-to-connect to 1/4 NPT fittings (BOM 5.3) to the (3) columns of cutouts on the left side of the enclosure body. Starting with the top left solenoid manifold assembly, connect each solenoid output left to right to the left most column of push-to-connect fittings on the enclosure body from top to bottom. Repeat the process with the solenoid manifold assembly on the top right or the rear panel and the center column of the push-to-connect fittings. Repeat a final time with the lower right solenoid manifold assembly and the right-most column of push-to-connect fittings.

[*] Different solenoid manifolds will require slightly different implementations. Regardless, the supply port should have a barbed fitting and the exhaust port can be left unpopulated. Different solenoid manifolds may require minor modifications to the rear enclosure panel.

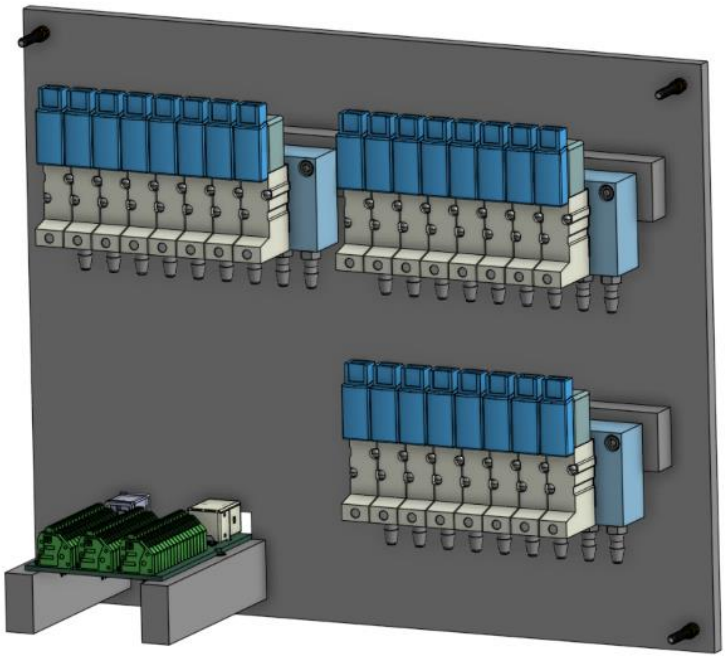


Figure C-5. Rear panel with Pneumadyne solenoid manifolds and custom solenoid driver PCB.

C.4 Solenoid Driver PCB

There are several integrated circuit components used in the assembly of the relay control PCB that are sensitive to electrostatic shock. It is important to work on an electrostatic dissipative (ESD) surface that is properly grounded while also handling the parts with ESD tooling. If a component is damaged, it will not be obvious until assembly and reflow has been completed and it is time to power up the board. This design uses small surface mount components with fine pitch leads that are tightly packed to minimize the footprint of the board. As a result, it is also difficult to rework the board and replace damaged components. Extra precaution early can save a lot of time.

There are numerous components used to assemble this PCB. Refer to the PCB bill-of-materials for specific parts and refer to the PCB schematic and layout files for specific placement of the components.

Begin by securing the PCB to a surface and securely overlay the 4 mil stainless steel stencil on top of the PCB such that the apertures of the stencil are aligned with the pads of PCB and there is no gap between the PCB and the stencil. Dispense solder paste along one edge of the stencil such that it spans the full length of the board. Using a straight edged paste spreader angled at 30 degrees, sweep the paste across the apertures while applying a steady downward force. Ensure that all pads have been covered with paste. Carefully lift the stencil vertically and begin populating the board with components as shown in the PCB layout file. Reflow the PCB following the thermal profile card included with the solder paste. Once the board has cooled manually solder the DC barrel jack connector and the wire-to-board spring connectors.

Inspect the solder joints and ensure that all components are bonded to the board and there are no solder bridges between the pins of components. Temporarily connect the board to the wall mounted DC power supply. If smoke is observed, disconnect immediately. If both the VDC and 3V3 LEDs are not emitting light, there is an issue with the board that should be diagnosed. If the issue was not trivial and correctable, a new board should be assembled. If both LEDs emit light, connect the board to a PC with a USB cable. If the USB LED emits light and the PC detects a new device, the PCB is ready for use.

Mount the PCB to the enclosure rear panel (BOM 5.2) using 4-40 screws (BOM 5.7). Progressing left to right and top to bottom one solenoid at a time, route the power leads through the space behind the manifold and down to the PCB, cut the wires such that there is an extra 5 cm of length form where they reach the connector, strip the ends of the wires and insert into the connector. Progress along the connector positions from A1 to C8.

[*] The solenoid driver PCB can be replaced with alternatives such as generic FT245R relay boards (ie a set of 3 1982AY-USB boards from SainSmart). It will be difficult to fit them into

the enclosure without making it significantly larger. Exploring alternatives may require writing Python 3 drivers if not already provided in the custom plfluidic module. The SainSmart board requires each relay to be individually powered by running a wire from the V+ terminal of the power terminal to the center terminal of each relay. The V+ wire of the solenoid valve is connected to either the left or right terminal and the V- wire is connected to the GND terminal of the power connector.

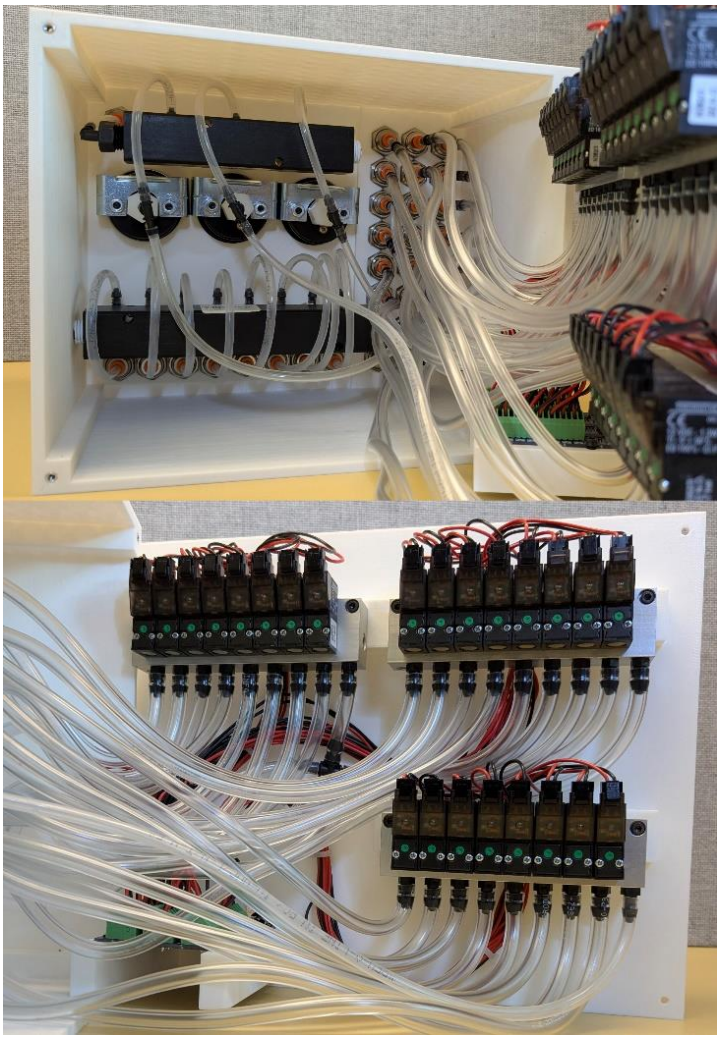


Figure C-6. Photos of internal system layout for additional reference.

C.5 External Connections

Connect the NPT to push-to-connect adapter (BOM 2.2) to both the in and out ports of the regulator with an air filter (BOM 2.3). Route polyurethane tubing (BOM 2.1) from the output regulator to the “PIn” port on the front face of the enclosure. Route polyurethane tubing from the input of the regulator to the air supply. Adjust the regulator so that it is fully closed. Feed pressurized air from the air supply to the regulator and gradually open the regulator until it is between 45 and 60 PSI.

Set P1 to 15 PSI and listen for potential air leaks. If there is a potential leak, search it by feeling for pressurized air around fittings by hand. Set P1 to 0 PSI and set P2 to 15 PSI. Search for leaks once more. Set P2 to 0 PSI and set P3 to 15 PSI. Search for leaks once more. If there are no leaks, individually toggle the input reagent two-way valves to confirm that air passes to the ports on the front face. Set P3 to 0 PSI. Fix leaks as needed. Seal the enclosure body (BOM 5.1) and enclosure rear panel (BOM 5.2) with 4-40 screws (BOM 5.7).

Populate all remaining 1/4 NPT fittings (BOM 5.3) on the enclosure with NPT to luer-lock adapters (BOM 5.4) and 23G luer-lock blunt end needles (BOM 5.5). Afterwards, cut sufficient Tygon microbore tubing (BOM 1.3) to reach from the microfluidic controller to the microfluidic device experimental workspace. Terminate the tubing by inserting a piece of steel tubing (BOM 1.12). A length of 3 to 4 feet is likely sufficient. Populate all the needles for P1 and P2 with microbore tubing. The input reagent manifold will only be populated during experiments and subsequently discarded. Mark the end of each tubing with an indexing system similar to that shown in Figure C-7. The indexing system helps users identify which tubing should be connected to which port on microfluidic devices. Code sets with increments of 4 were used in the system presented

here. After marking the microbore tubing, consider braiding the tubing for neatness and organization.

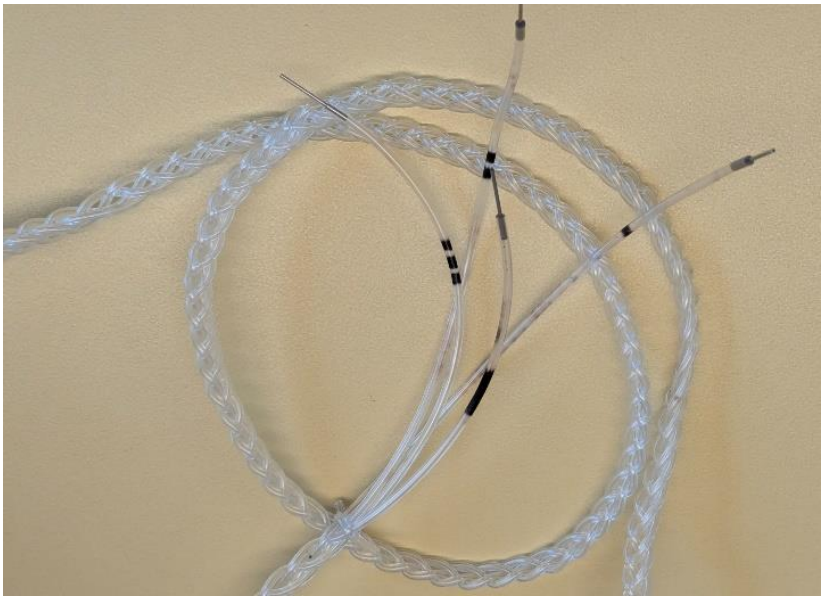


Figure C-7. Photo of Tygon microbore tubing with coding scheme in a 4-strand braid. It is recommended to code each column of the controller from 1 to 8 using small lines for increments of 1 and a solid bar for increments of 4. The braiding and coding scheme help with correctly pairing tubing to microfluidic ports.

C.6 Software Installation

Installing software to control hardware can be problematic due to deprecation of dependencies and the diversity of possible computer configurations. Two options for installing software for this microfluidic control platform are presented for this purpose. Yocto is an automated operating software (OS) build platform that helps facilitate dependency management with a set of configuration files and version management. The Yocto method discussed below generates a custom, minimal OS that can be used to control the microfluidic pressure controller. It is strongly recommended to compile the image if you have a computer that is powerful enough to do so. If not, an alternative build method that includes features on top of the default Raspberry Pi OS is also described. Given that Raspberry Pi OS is actively managed, it is subject to change and the methods described below may no longer be applicable. If the control system is not a Raspberry

Pi, roughly follow the Raspberry Pi instructions for installing the Python module, installing the drivers, and launching the program.

C.6.1 Yocto Build (Option A)

A Debian machine with 32GB of DDR3 RAM and a Ryzen 3700X processor was used to build the image described here. The building process is a compute and memory intensive process that will take 4.5 hours to complete on the system described above. A system with less than 32 GB is likely to run out of memory during the build process.

Install the bitbake utility called kas to populate Yocto build directory with all the required layers. Kas can be installed either via a Docker container (https://github.com/siemens/kas) or a Python 3 package (https://kas.readthedocs.io). Once kas is installed, create a directory and clone software repository listed in the Data Availability section of **Chapter 3**.

Edit parameters in the image configuration file located at "kas-project.yml" to match desired system behavior. Optional parameters include user credentials, wireless network credentials, a Tailscale authentication key, and system timezone. The wireless credentials allow for a wireless connection if ethernet is not available and a remote connection is desired, including a Tailscale authentication key greatly facilitates connecting to the device remotely without managing network properties, and the timezone information is useful for the logging utilities of the system. A wired ethernet connection is recommended. To create a Tailscale authentication code, create an account and follow their published instructions. In addition, install Tailscale on your personal computer so that it can communicate with the Raspberry Pi once it boots. Once all parameters have been set, follow the kas documentation to begin building the image.

Once the image has finished building, flash the microSD card with Balena Etcher or similar software. The image will be located at build/temp/deploy/images/raspberry-pi/plfluidic-rpi.XXX.wic.gz, where XXX is the timestamp of the build process. It should be one of the largest files in the directory.

Insert the SD card into the Raspberry Pi, connect to the device with a wired ethernet cable, and then power on the system. The first time the Raspberry Pi is powered on, it may take up to one minute for the device to appear on the Tailscale admin dashboard. Once it is powered and networked, access the server by either typing the hostname or IP address of the device and port 5454 into the address bar of a web browser with web socket functionality like Firefox or Chrome. The address should look similar to X.X.X.X:5454 or plfluidic:5454. The device can be accessed via SSH using the credentials found in the local.conf file used for building the image.

In addition, while this system was designed for remote operation over a network, it can also be used locally without a network connection. The system is designed to boot into a kiosk mode where only the server interface is accessible and the only requirements include a monitor, mouse, and keyboard.

C.6.2 Raspberry Pi OS Build (Option B)

Download the lite Raspberry Pi OS from the Raspberry Pi Foundation's website. At the time of writing, OS version 2.3 is available and compatible with this guide. Burn the OS onto an SD card using Balena Etcher or other similar software. Follow the Raspberry Pi Foundations instructions for setting default username and password in addition to enabling SSH.

Insert the SD card into the Raspberry Pi, connect the Raspberry Pi to a local router via ethernet, and power the device. Find the IP address of the device on the dashboard for the router and SSH into the system using the default credentials.

Download the software from repository listed in the Data Availability section of **Chapter 3** onto the Raspberry Pi. Create a virtual environment for Python 3 and PIP install the plfluidic Python module.

- python3 -m venv 'venv_plfluidic'
- source venv_plfluidic/bin/activate
- pip install path/to/plfluidic/directory

All the dependencies required for running the hardware control server should automatically download and install. Test if the server is functional by calling python -m plfluidic.app. The terminal should be locked while the process is running. Using a web browser with websocket functionality such as Firefox or Chrome, navigate to port 5454 of the Raspberry Pi IP address in the address bar. It should look similar to X.X.X.X:5454. Reaching the configuration page of the server indicates that there are no network restrictions. If the page cannot be accessed, it is recommended to install Tailscale to bypass all network configuration steps. Create a Tailscale account and follow their installation instructions for the Raspberry Pi as well as a personal computer. The Raspberry Pi OS is based on Debian which is a variant of Linux. Once Tailscale is running on both systems, the Tailscale admin dashboard should show two devices as being online and their IP addresses. Enter the IP address and port into the browser address bar as described above using the Tailscale IP address for the device. The configuration page should be accessible.

Next, create a systemd service so that the server can automatically run at boot. Start by killing the previously launched server process running in the SSH terminal with CTRL + C. Create a launch script for the server as described below and name it “plfluidic_launch.sh”. The ftdi_sio and usbserial driver modules are removed in the script because they can interfere with the detection of the relay driver by the Raspberry Pi.

```
---
#!/bin/bash
/sbin/rmmod ftdi_sio
/sbin/rmmod usbserial
source /home/venv_plfluidic/bin/activate
python3 -m plfluidic.app
---
```

Afterwards, change the permission of the script so that it can be executed and modified by the root and user accounts by calling chmod 0770 plfluidic_launch.sh. Next, create a systemd service by generating a new file named “/etc/systemd/system/plfluidic.service” with the contents as described below.

```
---
[Unit]
Description= plfluidic server
After=network.target

[Service]
User=root
ExecStart=/usr/bin/bash /home/plfluidic_launch.sh
Restart=always
StandardOutput=journal

[Install]
WantedBy=multi-user.target
---
```

Enable the systemd service so that it will launch at start up with the command systemctl enable plfluidic.service and confirm that it is running with the command systemctl status plfluidic.service. Power cycle the Raspberry Pi and confirm that the configuration page is

accessible by browser. If it is not, check the status of the plfluidic.service and check for error messages from the systemd service with journalctl -u plfluidic.service. If there are no errors, the server is running but there may be network issues. Confirm that Tailscale is running and that the IP has not changed if using it.

C.7 Configuration

Each system may be configured slightly differently to accommodate different experimental needs. This section covers general advice. To configure the user interface to accommodate new microfluidic devices, refer to **Appendix D.**

C.7.1 USB connections

During development, the Raspberry Pi 4B was found to have a sensitive USB interface where devices would occasionally not be detected. Connecting all peripherals to a powered USB 3 hub was found to resolve the instability of the bus. It is recommended to connect all peripherals to a powered hub if using a Raspberry Pi. This issue was not observed when using a Windows 11 laptop.

C.7.2 Supporting new hardware

Hardware drivers for the custom 24 channel solenoid driver, generic FT245R relay control boards, New Era 500 series syringe pumps, and a deprecated custom solenoid driver designed by Rafael Gómez-Sjöberg are provided in the software found in the previously described software repository. Review these validated examples.

Supporting new hardware includes:

- Developing a Python 3 hardware driver class that defines how to connect to and disconnect from the hardware, issues commands to the hardware, and provides optional quality of life methods. The “ft245r.py” file has a simple example.
- Developing a hardware unit class. The hardware unit class is used to manage the virtual state of hardware, includes methods to manipulate the hardware, and uses the hardware driver class to communicate with the physical hardware. The “valve.py” file includes a set of hardware classes for different drivers.
- Developing a hardware controller class. The controller class represents a collection of hardware being used such as a collection of FT245R relay boards. The purpose of this hardware controller is to accept commands from the program and route them to the appropriate hardware unit. The main application should be agnostic to how the hardware is physically arranged. Review “valve_controller.py” for reference.
- Updating the “driverSet()” method of the “ModelHardware” class to include the new hardware controller class. This method tells the program how to parse the configuration file to initialize the correct hardware controller class.
- Updating the “driver” entry of configuration files to match the option created in “driverSet()”.
- (Situational) If new methods are required (ie similar to the included “openValve()” or “closeValve()” methods) and do not exist, additional methods need to be developed in “createApp()”, “ModelHardware”, “ModelScript()”, “MicrofluidicController”, and HTML pages. At the time of publication, standard methods for actuating solenoids valves and

operating syringe pumps exist and additional methods are not likely needed for those types of hardware.

C.7.3 Controlling hardware via USB

The hardware drivers discussed in the previous section can be used independently of the "plfluidic" application. Fastest operation can be obtained by directly using the hardware driver class into custom Python 3 scripts. For greater convenience at the expense of slightly heavier overhead, the hardware unit or hardware controller classes can be used. The differences between these classes can be found in the preceding section. There may be hardware access issues if the "plfluidic" application is running.

C.7.4 Controlling hardware with HTTP and WebSocket APIs

The "plfluidic" application can be controlled remotely without using the web browser user interface by directly issuing HTTP and WebSocket commands to the port that the application is listening to. All interface commands that can be issued to the application are found in the "createApp()" function of the "app.py" file. WebSocket information that can be received can be found on the HTML pages. Prior to using the API, manually use the user interface to understand the order of operations required for configuring and controlling the system.

Appendix D.
Microfluidic Controller Operation Guide

D.1 Overview

The purpose of this document is to provide a reference for operating a completed microfluidic controller based on the *plfuidic* Python application. It covers navigating the browser-based user interface (UI), running experiments, and considerations for supporting new microfluidic devices. Information regarding building the system or adding new hardware capabilities can be found in an **Appendix C.** Additional notes that may be useful to the reader are demarked with **[*]**.

D.2 Overview of the user interface

The user interface is split into two separate views for selecting a hardware configuration to initialize and controlling the hardware, subsequently referred to as the config and control views.

D.2.1 Navigating the control view

Begin by navigating to the *plfluidic* server. This can either be found by using a web browser and connecting to port 5454 of the IP address of the hosting device (ie ‘xxx.xxx.xxx.xxx:5454’) or by using a keyboard, mouse, and monitor with the custom Linux image, which is in kiosk mode by default. Once there, the default configuration view is presented, as seen in Figure D-1. The configuration view contains panels for listing the existing configuration files and also previewing the contents of a selected file.

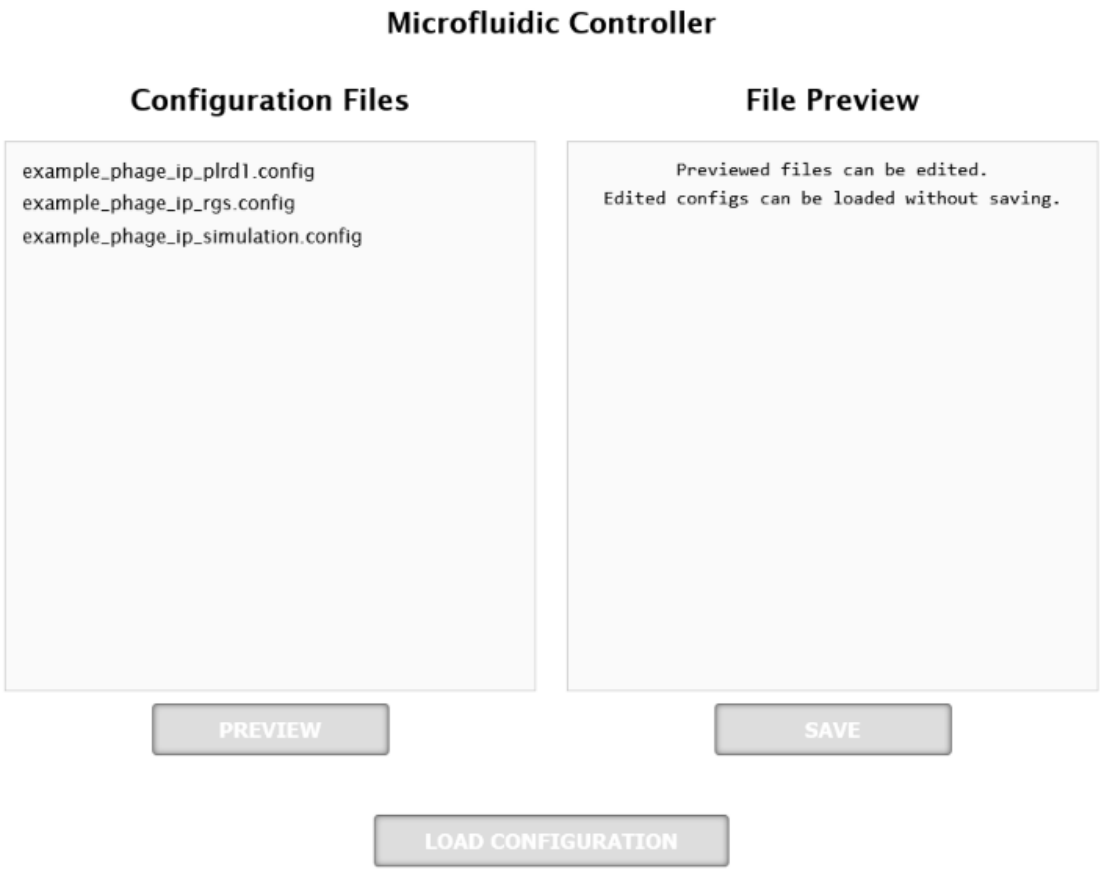


Figure D-1. Default configuration view.

Once the configuration view is accessed, the user can select a configuration file in the left panel and optionally preview the contents of the file or directly load it to initialize hardware as shown in Figure D-2.

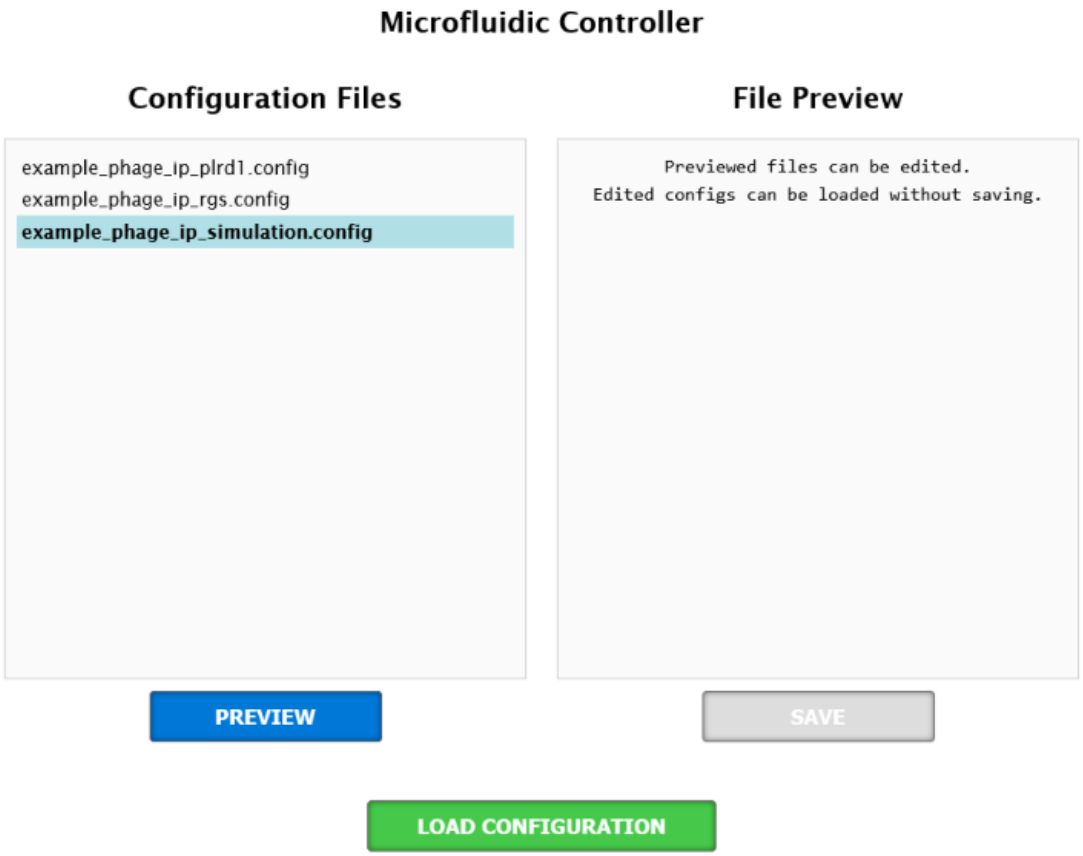


Figure D-2. A configuration file has been selected in the configuration view.

With the selection of a configuration file, the user may choose to preview the contents, which will be displayed in the right panel, as shown in Fig S10. The user may make changes to the configuration file as needed in the preview panel. Saving the contents of the preview panel will overwrite the contents of an existing file if the "config_name" property is not changed. Changes that are made in the preview panel are not permanently retained if the contents are not saved. This uniquely allows users to test changes to configurations by loading without saving.

[*] If a file has been previewed, the contents of the File Preview panel will ALWAYS override the selected configuration file! This means that if you preview config A but want to load config B, you must preview config B to change the contents of the file preview panel.

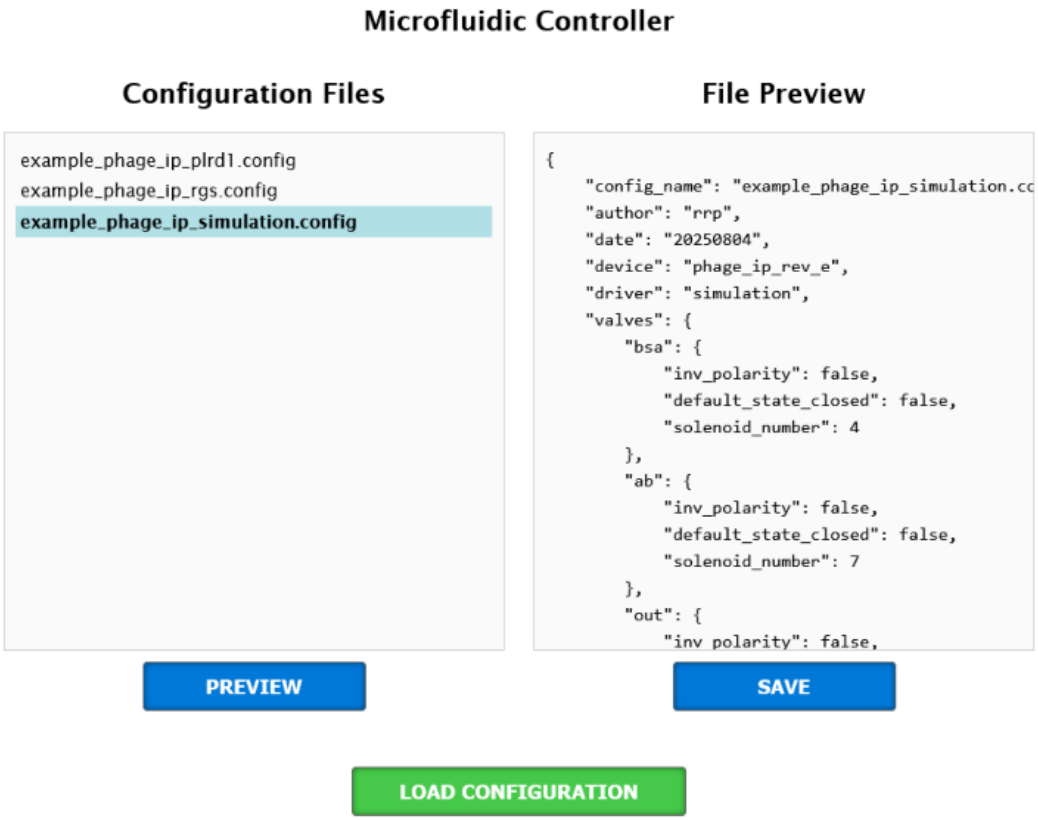


Figure D-3. The configuration view with a config file being previewed.

D.2.2 Configuration files

The configuration files are JSON formatted text files that store parameters for how the hardware should be initialized as well as file documentation. These files are located in the *plfluidic* library directory under '/server/configs'. While files can be created outside of the interface, it is strongly recommended to use the config view UI for editing and creating new files. Each time a config is saved or loaded, the program parses the file to validate structure and contents and warns the user if any issues were encountered.

D.2.3 Configuration file properties

The properties used in a file vary by the hardware being used. While documentation properties are always used, valve or pump properties are optional. Only one of each type of driver is supported. As an example, using drivers for both the custom solenoid driver board and generic FT425R boards simultaneously is not permitted. However, you can use multiple FT425R boards or you can use a one type of syringe pump and valve driver together. For full examples, refer to the configs provided in the library. Additional guidance, refer to section 3.1 Creating new configuration files.

Required properties:

```
"config_name": "example_phage_ip_simulation.config",
"author": "rrp",
"date": "20250804",
"device": "phage_ip_rev_e",
```

“device” is used to load a specific control view for operating a device. The field should match the name of a control view HTML page.

Valve properties:

```
"valves": {
        "driver": "simulation",
        "address": "abc",
        "valve_name_1": {
                "inv_polarity": false,
                "default_state_closed": false,
                "solenoid_number": 0 },
        "valve_name_2": {
                "inv_polarity": false,
                "default_state_closed": false,
                "solenoid_number": 1 }
}
```

Each valve is provided a text name that will be linked to the control view as well as physical parameters such as valve polarity for normally-closed or normally-open valves. If a valve is normally-open, that means that powering it will depressurize elastomeric microfluidic valves. In this case, “inv_polarity” should be set to “true” so that the program represents the powered state correctly. In the case of “default_state_closed”, this specifies if the microfluidic valve should be set to the closed state when the hardware is initialized. The “solenoid_number” parameter specifies the position where the named valve is located on the valve driver.

Pump properties:

```
"pumps": {
"driver": "simulation",
"address ": "abc",
"pump_name_1": {
        "dir_inject ": true,
        "mode_flow": false,
        "units ": "ul",
        "vol_rate": 50,
        "pump_number": 1 },
"pump_name_2": {
        "dir_inject ": false,
        "mode_flow": true,
        "units ": "uh",
        "vol_rate": 50,
        "pump_number": 2 }
}
```

Each pump is provided a text name that will be linked to the control view as well as physical parameters. In the case of pump 1 in example above, it is set to inject 50 μL each time it is set to run. Pump 2 differs and is set to withdraw at a rate of 50 μL/hour when it is set to run.

The valves and pumps specify which driver to use as well as a serial number or port address to use with the driver.

D.2.4 Navigating the control view

In the control view, hardware can be manually actuated, scripts can be created, ran and interrupted, a system log stream can be viewed, and hardware can be de-initialized so that a new configuration can be applied. Figure D-4 presents an initial view of the control interface for the included file titled “example_phage_ip_simulation.confg”.

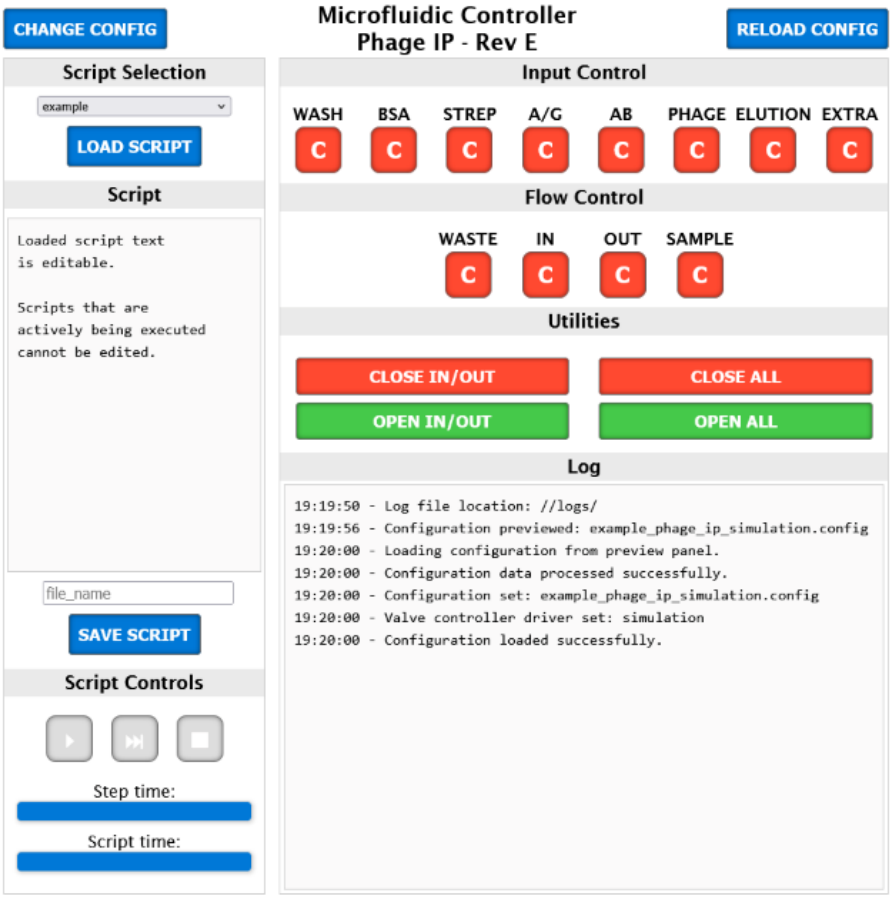


Figure D-4. The control interface for the example “phage_ip_rev_e” device.

Once the control interface is accessible, the user may choose to manually actuate the hardware or load scripts for experimental operations. Scripts for priming the control layer and displacing air is one recommended setup script to run. Figure D-5 shows the interface once a script has been loaded.

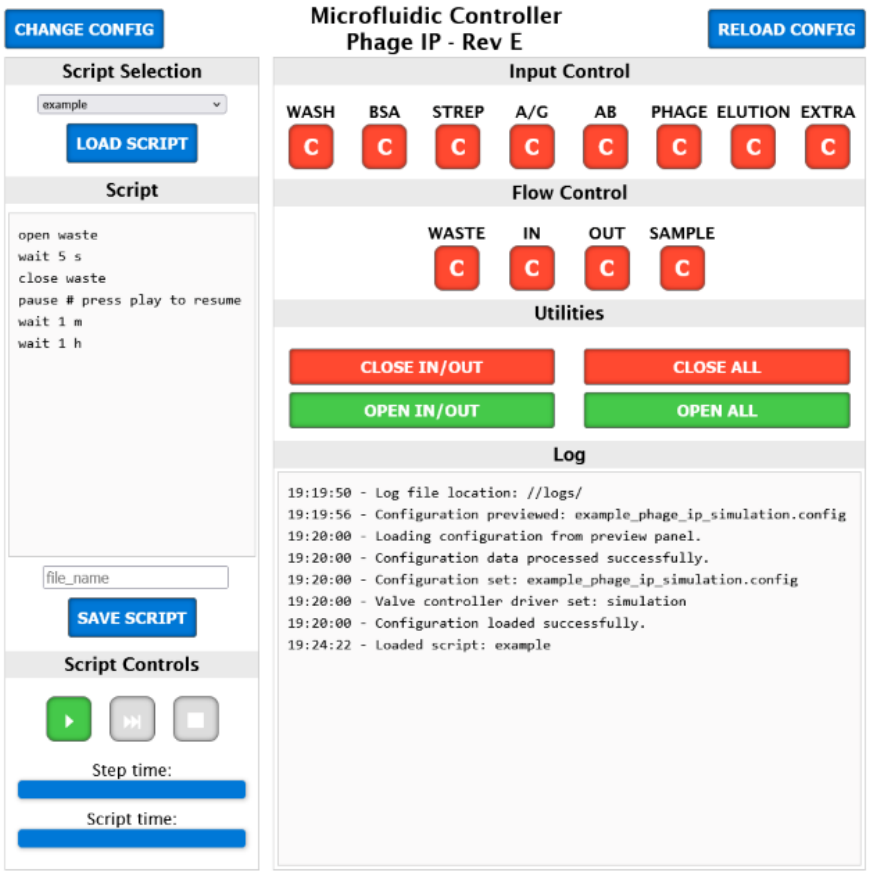


Figure D-5. The control view with a script loaded but not started.

Once a script has been loaded, the contents of the script file are populated into the editable "Script" panel. In this panel, users may choose to edit script operations, save new scripts, or execute the operations presented in the panel. Guidance for creating new scripts is presented in the following section titled "Creating scripts". When the user clicks the play button in the script controls section, the script is parsed, checked for errors, and total script time is estimated. Figure D-6 shows the control interface with a script running.

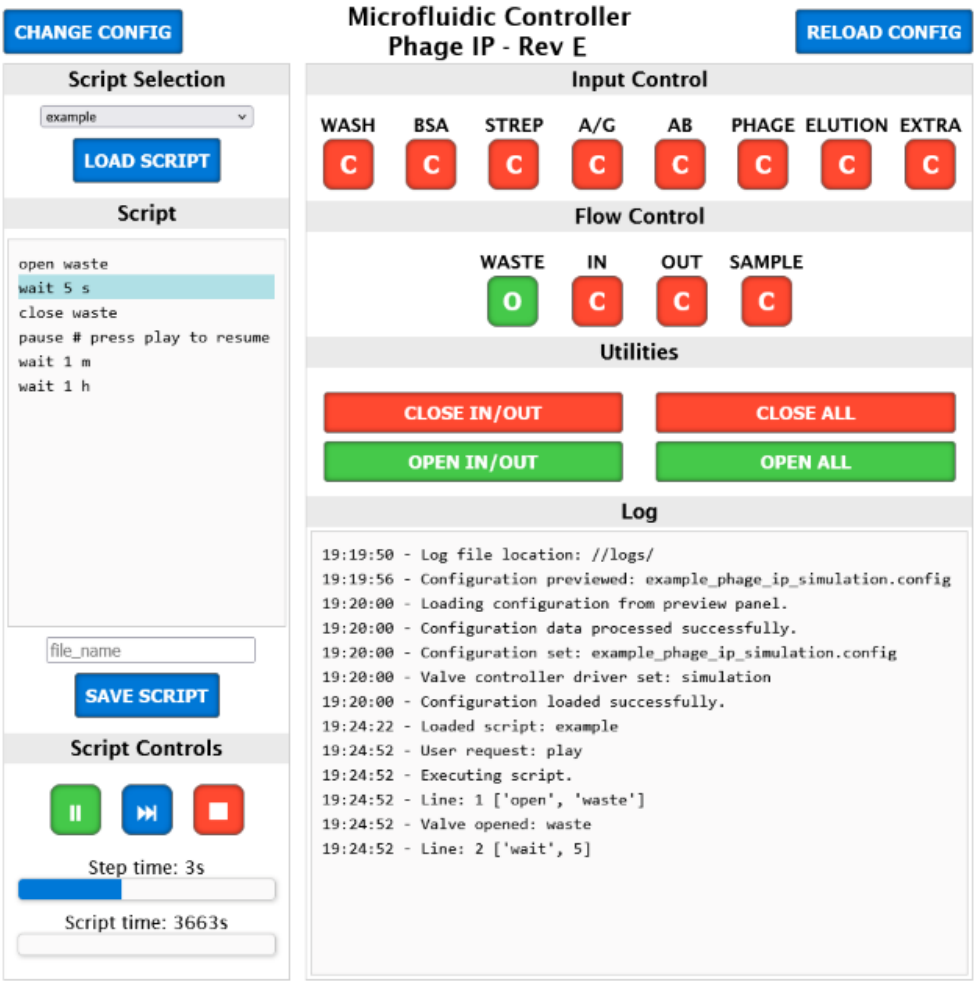


Figure D-6. The control view with a script being executed.

As the script is executed, the current line is highlighted in blue in the script panel and estimated times for step completion and total script completion time are presented below. While a script is loaded, access to control buttons also changes. The script can be paused/resumed, a step in the script can be skipped, and the script can be terminated. If the script is executing, manual control of the hardware is disabled; however, the state of the hardware on the interface continues to be updated in real time. If the script has been paused, the manual control of the hardware is re-enabled and until the script is resumed by the user.

D.2.5 Script operations

At the time of publication, the script interface is basic. It allows for commands to manipulate the hardware, but it does not currently support macros or other utilities such as for-loops. The allowable operations are listed below.

Universal operations:

- wait N ms/s/m/h
- pause

A step timer can be created with the “wait” command. This will cause the script to continually loop until the time period has expired. Time units include “ms” for milliseconds, “s” for seconds, “m” for minutes, and “h” for hour. N represents the number of units to wait. The script can also be interrupted indefinitely with a “pause” step until the operator resumes it by clicking the play button. While in this interrupted state, the user may manually control the hardware using the interface.

Valve operations:

- open valve_name
- close valve_name

As named, these operations open or close an individual valve. Actuating groups of valves simultaneously is currently not supported and data from the main publication show that each script step takes < 10 ms to complete.

Syringe pump operations:

- mode pump_name flow/vol
- dir pump_name inj/wth
- flowrate pump_name N um/mm/uh/mh
- volume pump_name N ul/ml
- start pump_name
- stop pump_name
- purge pump_name

These operations support basic pump operations. The “mode” operation allows the syringe pump to change the mode of operation between constant flow and constant volume. “dir” allows the direction of the pump to be changed between injecting and withdrawing directions. “flowrate” changes the flow rate of the pump in with “um” units for microliters per hour, “mm” units for milliliters per minute, “uh” for microliters per hour, “mh” for milliliters per hour. “volume” changes the volume to be dispensed or aspirated by the pump when in volume mode and the units are “ul” for microliters and “ml” for milliliters. “start”, “stop”, and “purge” perform their named operations. Purge will move the pump at a fast speed until it is stopped.

D.2.6 Creating scripts

Scripts can be made in the editor of the UI or with a separate file editor and they are stored in the ‘/server/scripts’ directory of the *plfluidic* Python module. Creating scripts with the UI is recommended as it parses the script when saved to ensure that it is correctly formatted and executable.

Scripts that are written in the script panel can be executed without saving. The start button extracts the contents of the window, processes, and then executes the content. If the content is to be saved, enter a name into the “file_name” the text box below the script and click the “Save Script” button.

D.2.7 Script state machine

When a script is started, the main application creates a new thread in Python that begins to execute the script. The state machine is described in Figure D-7 and is only executed on the controller hardware and not on the computer of the operator.

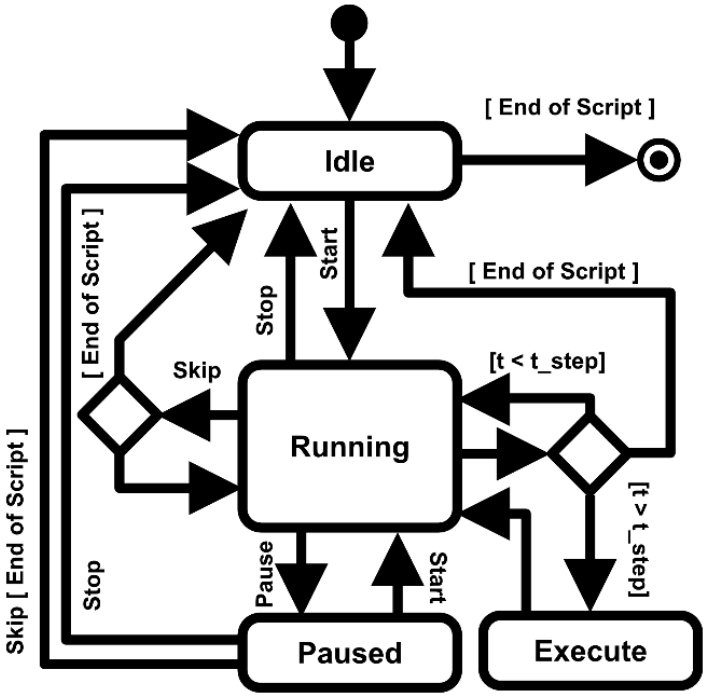


Figure D-7. A representation of the script execution state machine.

The script begins initialized in the idle state and begins running when the user clicks the start button. While running, each command will be promptly executed unless a wait command has been issued. In the wait command, the current step time is continuously being checked until it exceeds the specified wait duration. Once the wait duration has passed, the step is executed and discarded from the buffer. Commands that are not the wait command have a default step time of 0.

The execution step is responsible for setting script timers, managing the script buffer, and issuing commands to the application controller so that they can be correctly dispatched by the model-view-controller. Refer to Figure 2-4 for a generalized description of the model-view-controller framework of the application.

The user may choose to interact with the script by pausing, skipping steps, or stopping the script. When paused, the script transitions from the running state to the paused state and ceases to execute commands until it is returned to the running state by clicking the start/resume button. If the script is stopped by the user while in the running or paused state, the contents of the script are discarded and the state machine transitions to the idle state where the state machine and thread are subsequently terminated. Reaching the end of the script has an identical effect.

D.2.8 File locations

Configuration files and control views are stored where the *plfluidic* Python library is installed. To find where the library is installed, run ‘systemctl status plfluidic’ to find the service launch script. The launch script should specify which virtual environment or Python source is being used. If the custom Linux image is being used, the library can be found at ‘/usr/lib/python3.12/site-packages/plfluidics’. The configs are found in ‘/server/configs’ while the control views are found in ‘/server/templates’. Configs can be created and edited in the config view, but control views for new devices will need to be manually created. Scripts can be found in ‘/server/scripts’. Logs can be found in the directory where the application was launched from. If using the custom Linux OS, this is the root directory ‘//logs’.

D.3 Operation

Advice for general experimental procedures is provided here for solenoid valve operation as the instrument inherently supports solenoid valves. Syringe pump set up and shut down experimental processes differ and guidance for using that hardware can be found with the documentation provided by the manufacturer.

D.3.1 Experiment Set Up

D.3.1.1 Connect to the microfluidic controller

Using the IP address of the system, navigate to port 5454 with a web browser. The address should look similar to “xxx.xxx.xxx.xxx:5454”, where x represents digits from the IP address. If the device is running on the custom Linux image and has a keyboard, video, and mouse, the system will already be in kiosk mode and no further navigation is required. Select the configuration file that matches the microfluidic device to be used and the required hardware configuration. Load the configuration file.

D.3.1.2 Depressurize the system

Inspect the input pressure regulator attached to the air supply. If there is a buildup of oil or debris in the filter attached to the regulator, replace the filter as needed. If the regulator valve is currently open, turn clockwise to close the valve. Inspect the output pressure regulators on the front panel of the microfluidic controller and turn clockwise to close the valves.

D.3.1.3 Prepare solenoid valve control lines

Multilayer PDMS devices with elastomeric valves are usually actuated with water to prevent air from bleeding into the flow layer.

For each control line that will be used by the experimental microfluidic device, cut a piece of Tygon tubing to 3" in length, insert a steel pin into one side and a syringe with a 19G blunt end needle to the other. Aspirate water into the tubing, remove the syringe and replace it with the steel pin connected to the Tygon tubing that leads to the solenoid valve in the controller. Attach the unconnected end of the tubing with water to the desired control valve on the experimental microfluidic device.

Other systems such as the Brower *et al.* design use larger water reservoirs on the solenoid output instead of the 3” of Tygon tubing used here. That is also an option. 3” of tubing was found to contain enough water that is sufficient for at least a week’s worth of devices and the reduced volume also adds less pneumatic capacitance – theoretically improving actuation speeds.

Open the supply pressure valve that connects to the input of the input pressure regulator. Adjust the regulator to 60 PSI by turning the valve clockwise. The pressure regulators in the microfluidic controller cannot output more than 30 PSI and typically are set to a maximum of 10 PSI. 60 PSI is an effective set point for the input pressure.

Adjust the P1 and optionally P2 regulators to set the output pressure for the control lines. The optimal pressures for P1 and P2 will be dependent on the geometry of the device and the pressure of P3. The P1 regulator manages 16 solenoid valves and the P2 regulator manages and additional 8 solenoid valves. Some devices may require a subset of valves to operate at a different pressure as in the case of micropumps or valves with different heights, but that is not a common requirement.

D.3.1.4 Fill the control layer with water

Using the control view of the UI, pressurize all valves to fill the control layer of the microfluidic device with water. During this process, the pressurized water pushes air out of the layer through the porous PDMS. It often takes several minutes and the air-water interface can be observed with a stereoscope. Once all the lines have been filled with water, examine each set of pressurized elastomeric valves and ensure that they are fully sealed. If they are not sealed, increase the P1 or P2 pressures on the controller.

[*] Do not proceed until all valves are sealed and air has been fully purged from the control layer.

D.3.1.5 Prepare input reagent lines

Adjust the P3 regulator to set the output pressure for reagents that flow through the device. Switch the toggle valves on the front panel to the off position, if needed. The optimal input pressure is dependent on the geometry of the device and the desired experimental properties such as the elastomeric valve closing pressures or reagent flow rate.

If reagent volume is 100 uL or less, each reagent can comfortably be aspirated into 3’ of Tygon tubing using the method described in the control line preparation section. Add a steel pin to the output side of the reagent tubing, connect the steel pin to the reagent port on the microfluidic device, and connect the pneumatic input side of the tubing to the desired toggle switch valve. Reagent flow can be slow in some experiments. Consider marking the meniscus

[*] If larger volumes of reagents are needed, consider using reagent reservoirs. These can either be fabricated in lab or purchased from vendors such as Elveflow or Fluigent. Additionally,

if more than 8 reagents are needed, consider using an external manifold that can be used to split the pressure source to multiple reagents.

D.3.1.6 Fill the reagent ports of the microfluidic device

Once the reagents are connected to the controller and the device, pressurize all reagents with the toggle switches to purge the air out of the flow layer. Wait until the reagent-air interface reaches the first closed valve that manages the reagent. This process may take several minutes. If there is no flow into the device, there may be an obstruction in the microfluidic port, such as a partial PDMS plug from the punching process.

[*] If reagents are observed to leak into the device while running an experiment, the P3 regulator is set to a pressure that is too high or the P1 or P2 regulator is set to a pressure that is too low. There may also be a device defect.

D.3.2 Experiment Operation

The exact experimental methods used will be device dependent. In general, it helps to write scripts that sequentially route each output to the waste for 1 second to ensure that all bubbles and air is purged from the reagent lines. Afterwards, scripts are run to automate the microfluidic experiment.

D.3.3 Post-experiment shut down

Upon completing an experiment, begin by depressurizing P3, flipping the reagent toggle switches to the off position, and disconnecting the reagent lines from the controller. Next, depressurize P1 and P2, depower the solenoid valves using the control UI, and disconnect the control lines from the microfluidic device. In the control view, click the “CHANGE CONFIG” button to return to the device to the config view.

D.4 Supporting new microfluidic devices

Supporting new microfluidic devices includes creating a configuration file and an associated control interface. The configuration file specifies the hardware to use, labels for hardware units, the initial states, and the control interface to load. The control interface is an HTML page that consists of labeled buttons and fields that are associated with hardware. To support new hardware or control the microfluidic controller with HTTP and WebSocket interfaces, refer to **Appendix C.**

D.4.1 Creating new configuration files

The configuration files are JSON formatted text files that are processed by the ModelHardware class in the *plfluidic* Python application. The class will be the most authoritative and up-to-date source for required configuration files if the guidance provided here is not sufficient. It is recommended to make configuration files with the browser UI by previewing an existing file, making changes to it, and saving it with a new "config_name". Using the UI will automatically parse the file when saved to detect formatting errors or unacceptable key-value parameter entries. For more complete description of JSON, please refer to other resources.

The configuration files generally use the following format:

```
{
"config_name": "config_file_name.config",
"author": "rrp",
"date": "20250804",
"device": "controller_html_file_name",
"valves": {
        [A,
        B,
        C]
        },
“pumps”: {
        [D,
        E,
```

```
            F]
            },
}
```

Where A, B, and C are independent valve controller boards that are accessible with unique ports and D, E, and F are independent pumps that are accessible on unique ports. Each of these letter groups represents a required driver initialization. If multiple units are accessible on the same port, they can share the same driver and should not initialize the driver multiple times.

An example would be the FT245R USB relay control board from SainSmart. Each board can actuate 8 solenoids, so 3 boards would be needed to actuate 24 valves. Each board has a unique port address and serial number. Since 3 different USB ports are being used, 3 driver initializations are required.

As a counter example, New Era syringe pumps can be networked so that they are connected to each other and each of them can be accessed with a single USB interface. In this case, they do not require separate driver initializations, and they would all be listed together as a single unit such as D.

Section “D.2.3 Configuration file properties” shows how to group hardware units together and details the parameters that are required for each type of hardware.

D.4.2 Creating new control interfaces

The control interface is an HTML document that contains formatted HTTP requests, JavaScript and WebSocket interfaces to pass information between the operator and the application. A large majority of these features do not need to be changed when creating new interfaces and should be left untouched in order to preserve scripting, logging, and reconfiguration features. The

only features that should be modified are components that are used to manually control the hardware. This primarily includes:

- The labels for solenoid valve toggle buttons
- The layout of solenoid valve toggle buttons
- The labels for pump units
- The layout of pump units

These features already exist in the provided examples. A new interface consists of copying an example, editing it in a text editor, and saving the edited copy in '/server/templates'. The structure of the HTML files is summarized below:

- Document description
- Top navigation bar
- Left column - Scripting
- Right column
    - **Row 1 for hardware**
    - **Row 2 for hardware**
    - **Row 3 for hardware**
    - Row 4 for logger
- JavaScript and WebSocket utilities

The following example is a code snippet for a button that actuates a button that is labeled as "wash" in the configuration file. The bold words are the only text that needs to be changed to match the parameters of:

```
<div class="btn-valve-outline">
        <label for="wash" class="btn-valve-label">Wash</label>
        <buttononclick="toggleValve(this)" id="wash"  class”…”>
                {% if valves['wash']=='open' %}O{% else %}C{% endif %}
        </button>
</div>
```

It has the format of:

```
Object class
        Object label
        Object properties
```

After editing, inserting, deleting the desired hardware control objects, save the HTML template. If the template is called “my_device_revA.html”, change the config file to load this interface by modifying the “device” filed of the config with “my_device_revA”. This will find and open the newly created HTML file.

Appendix E.
Bill-of-Materials for Microfluidic Controller

OTS – Off-the-shelf
3D – 3D printed component
CO – Custom order

Item No.		Name	Qty	Unit	Vendor	Part number	Type
1		**Common Components**					
	1.1	EVA Tubing, Firm	1	25 ft	McMaster-Carr	5154K15	OTS
	1.2	PVC Tubing, Flexible	1	50 ft	McMaster-Carr	5233K52	OTS
	1.3	Tygon Microbore Tubing	1	500 ft	Fisher Scientific	502365803	OTS
	1.4	Straight barbed tube fitting	3	10 Pk	McMaster-Carr	5463K38	OTS
	1.5	T barbed tube fitting	1	10 Pk	McMaster-Carr	5463K45	OTS
	1.6	Straight tube fitting, 10-32	1	10 Pk	McMaster-Carr	5463K53	OTS
	1.7	Straight tube fitting, 1/8 NPT	3	Each	McMaster-Carr	5463K119	OTS
	1.8	Elbow tubing fitting, 1/4 NPT	1	10 Pk	McMaster-Carr	5463K133	OTS
	1.9	Plug, 1/4 NPT	2	Each	McMaster-Carr	3867T363	OTS
	1.10	Regulator, 0-30 PSI	3	Each	McMaster-Carr	41795K4	OTS
	1.11	Pressure Gauge, 0-30 PSI	3	Each	McMaster-Carr	3846K4	OTS
	1.12	Stainless steel pins, 23G	100	Each	New England Small Tube Corporation	NE-1300-05	OTS
2		**Air Input**					
	2.1	Polyurethane tubing	1	25 ft	McMaster-Carr	5648K31	OTS
	2.2	Push to connect to 1/8 NPT	2	Each	McMaster-Carr	5779K108	OTS
	2.3	Regulator + Filter	1	Each	McMaster-Carr	9891K41	OTS
	2.4	Push to connect to 1/4 NPT	1	Each	McMaster-Carr	5779K108	OTS
	2.5	Manifold, 3 port, 1/4 NPT to 1/8 NPT	1	Each	McMaster-Carr	5975K34	OTS
3		**Reagent Output**					
	3.1	Toggle Switch, 2-way	8	Each	McMaster-Carr	6791T22	OTS
	3.2	Manifold, 8 port, 1/4 NPT to 1/8 NPT	1	Each	McMaster-Carr	5469K171	OTS
4		**Solenoid Output**					
	4.1	Solenoid valve, 2/3 way	24	Each	OC Pneumatics	S070B-6CC	OTS
	4.2	Solenoid manifold	3	Each	OC Pneumatics	SS073B01-08C	OTS
	4.3	Straight tube fitting, M3	1	25 Pk	Pneumadyne	EB30-M3	OTS
	4.4	Straight tube fitting, M5	1	10 Pk	McMaster-Carr	2974K934	OTS
5		**Enclosure**					
	5.0	PLA Filament	1	1 kg	Amazon	PA02096	OTS
	5.1	Enclosure - Body	1				3D
	5.2	Enclosure - Rear Panel	1				3D

Item No.		Name	Qty	Unit	Vendor	Part number	Type
	5.3	Push to connect to 1/4 NPT, panel mount	33	Each	McMaster-Carr	7880T343	OTS
	5.4	NPT to Luer Lock	4	10 Pk	McMaster-Carr	51525K433	OTS
	5.5	Needle, 1/2" 23G	1	50 Pk	McMaster-Carr	75165A684	OTS
	5.6	Heat-set Inserts, 4-40	1	25 Pk	McMaster-Carr	94459A412	OTS
	5.7	Socket head screws, 4-40	1	100 Pk	McMaster-Carr	91251A113	OTS
6		**Electronics**					
		Bare PCB	1	3 Pk	OSH Park	PLRD1	CO
		DC Power Supply	1	Each	Digikey	16-00217	OTS
		USB Cable	1	Each	Digikey	4473	OTS
		Raspberry Pi 4 Kit	1	Each	Newark	80AH9362	OTS
		Capacitor, 4.7uF 35V	1	Each	Digikey	T491B475K035ATAUTO	OTS
		Capacitor, 100nF	13	Each	Digikey	CL05B104KB5NNNC	OTS
		Capacitor, 4.7uF	1	Each	Digikey	GRM155C61E475ME15D	OTS
		Capacitor, 22uF	2	Each	Digikey	C2012X5R1V226M125AC	OTS
		LED, SML-P11UTT86R	4	Each	Digikey	SML-P11UTT86R	OTS
		Diode, MBR0580	24	Each	Digikey	MBR0580	OTS
		Connector, PJ-082BH	1	Each	Digikey	PJ-082BH	OTS
		Connector, UJ20-C-H-G-SMT-5-P16	1	Each	Digikey	UJ20-C-H-G-SMT-5-P16-TR	OTS
		Connector, 16p Spring Clamp	3	Each	Digikey	1989887	OTS
		Inductor, 600R/1A	1	Each	Digikey	MLG1005S18NJT000	OTS
		Inductor, 10uH	1	Each	Digikey	VLS201612HBX-100M-1	OTS
		Resistor, 12K	14	Each	Digikey	ERJ-PA2F1202X	OTS
		Resistor, 5k1	2	Each	Digikey	ERJ-2GEJ512X	OTS
		Resistor, 470R	8	Each	Digikey	ERJ-2GEJ471X	OTS
		Resistor, 0R	1	Each	Digikey	CRCW06030000Z0EA	OTS
		IC, DRV81008Q	3	Each	Digikey	DRV81008QPWPRQ1	OTS
		IC, TPS560430X3	1	Each	Digikey	TPS560430X3FDBVT	OTS
		IC, FT4222HQ	1	Each	Digikey	FT4222HQ-D-T	OTS
		Oscillator, SG-210STF 12MHz	1	Each	Digikey	SG-210STF 12.0000ML5	OTS

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

Signed by:
Robert Russell Puccinelli
2A76441469E540D...
Author Signature

08/27/2026
Date