8-15-2024 10:00 AM

The Rise of BREAD: Leveraging Open-Source Process Control

Finn Keith Hafting, *Western University*

Supervisor: Pearce, Joshua M., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

Abstract

Industrial pilot projects and the automation of experiments often rely on expensive and proprietary electronic hardware, known as supervisory control and data acquisition (SCADA) systems, to control and monitor areas of their processes. The Broadly Reconfigurable and Expandable Automation Device (BREAD) framework was created to address the need for inexpensive SCADA systems. New BREAD Slices were designed and compared to commercial heating, motor, pump, and pH controllers, exceeding all in functionality and cost. In each case, the decreased accuracy of the BREAD controller had little impact on the final products but greatly reduced the cost of the system. The modularity of BREAD was also desirable in situations where control zones were constantly changing, like the heating zones of a pyrolysis reactor. After improving the design to better meet the needs of a SCADA system, BREADv2 was integrated into an enclosure demonstrating its potential as a backbone for pH controllers.

Keywords

open hardware; open-source hardware; open-source electronics; Arduino; automation; data acquisition; controls; monitoring; supervisory control; SCADA

Summary for Lay Audience

When automating experiments or processes, researchers and technicians will choose products like the National Instruments CompactRIO or Opto 22 groov EPIC systems. These devices can be configured to meet a variety of requirements like controlling heating zones, monitoring pH, and actuating pumps. Unfortunately, these devices are also proprietary and incompatible with other systems. By limiting the choices of researchers, companies can then charge a premium to use their equipment which makes these devices inaccessible in low-resource settings. The Broadly Reconfigurable and Expandable Automation Device (BREAD) framework was created to address the need for inexpensive devices for researchers used in conducting experiments. This work improves upon the BREAD framework by designing devices within it to meet new needs like heating control and pH control. These new devices are compared to commercial equivalents in cost and functionality. In each comparison, the BREAD system was both cheaper and provided more functionality than the commercial equivalents without sacrificing performance. Furthermore, because BREAD is an open-source project, its performance and scope will only grow as users improve the design and add additional functionality.

Co-Authorship Statement

This paper is based on three articles:

Chapter 2:

Hafting, F.K., Kulas, D., Michels, E., Chipkar, S., Wisniewski, S., Shonnard, D. and Pearce, J.M., 2023. Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics. *Electronics*, *12*(24), p.4893. https://doi.org/10.3390/electronics12244893

Published under CC-by

Finn K. Hafting: designed BREAD Loaf for iteration 1; revised and tested RLHT and DCMT Slices; built BREAD system and integrated board; customized software for BREAD system and integrated board; programmed Slices; additionally, methodology, validation, formal analysis, investigation, data curation, writing - original draft preparation, writing - review and editing, visualization

Daniel Kulas: methodology, validation, formal analysis, investigation, writing - original draft preparation, writing - review and editing, visualization

Etienne Michels: designed OpenReactor software; methodology, writing - review and editing

Sarvada Chipkar: methodology, writing - review and editing

Stefan Wisniewski: methodology, resources, writing - review and editing

David Shonnard: writing - review and editing, supervision, project administration, funding acquisition

Joshua M. Pearce: conceptualization, methodology, validation, resources, data curation, writing - original draft preparation, writing - review and editing, supervision, project administration, funding acquisition

Chapter 3:

F. K. Hafting, S. Yeung, H. Al-Aribe, E. Michels, and J. M. Pearce, Parametric Low-Cost Open Source Bioreactor Designed for Distributed Manufacturing, (In Review)

Finn K. Hafting: designed BREAD Loaf for iteration 2 and 3; revised and tested RLHT and DCMT Slices; designed PHDO Slice; built BREAD system; designed new cases for BREAD iteration 2 and 3; customized software for BREAD system; programmed Slices; additionally, methodology, data curation, visualization, validation, writing- original draft, writing-reviewing and editing

Stephen Yeung: designed bioreactor; additionally, methodology, data curation, visualization, validation, visualization

Hamida Al-Aribe: writing- reviewing and editing

Etienne Michels: OpenReactor software design, data curation, visualization, writing-reviewing and editing

Joshua M. Pearce: conceptualization, supervision. methodology, writing- original draft, writing- reviewing and editing

Chapter 4:

Finn K. Hafting, Xander Chan, Jeff T. Hafting, and Joshua M. Pearce. Converting the Open Source Broadly Reconfigurable and Expandable Automation Device (BREAD) into a Full Supervisory Control and Data Acquisition System (to be published).

Finn K. Hafting: designed BREAD Loaf backplane, Loaf controller, RLHT, DCMT, PHCL, and PHDO Slices for iteration 4; designed cases for iteration 4; programmed Slices; adapted Butter software for pH control applications; built BREAD pH control enclosure; additionally, conceptualization, methodology, visualization, validation, writing- original draft, writing-reviewing and editing

Xander Chan: Butter software design

Jeffery T. Hafting: conceptualization, data curation, methodology

Joshua M. Pearce: conceptualization, supervision. methodology, writing- original draft, writing- reviewing and editing

Acknowledgments

I would first like to thank my friends and family for their unwavering support.

To my supervisor Dr. Pearce, your passion for open-source hardware has connected me with an amazing community of innovators. I'm truly thankful for the opportunities you have given me to present my work and explore the world of open-source hardware.

To my Free Appropriate Sustainability Technology (FAST) lab friends, thank you for any assistance you provided with my research from questions answered to projects we collaborated on. I'm thankful to have been surrounded by such a smart and passionate group of people.

To my friends at Michigan Technological University (MTU), collaborating whether over Zoom or in-person has been a truly rewarding experience. Thank you for your hospitality and working with me through numerous software bugs.

To my loving girlfriend, Abi, I could not have done this without your support. Thank you for cultivating my passion for this work even when I found it difficult.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

Acronym	Abbreviation
ADC	Analog-to-digital converter
BOM	Bill of materials
BREAD	Broadly Reconfigurable and Expandable Automation Device
CAD	Computer aided design
CAN	Controller Area Network
CO2	Carbon dioxide
CSA	Canadian Standards Association
DAQ	Data acquisition
DCMT	DC motor controller Slice
DO	Dissolved oxygen
FOSH	Free and open source hardware
FOSS	Free and open source software
GC-MS	gas chromatography–mass spectroscopy
GUI	Graphical user interface
I/O	Input output
I2C	Inter-integrated circuit protocol
IDE	Integrated development environment
JSON	JavaScript Object Notation
MOSFET	metal-oxide-semiconductor field-effect transistor
MRE	Meals-ready-to-eat
NI	National Instruments
OS	Open source
PCB	Printed circuit board
PHCL	pH controller Slice
PHDO	pH and dissolved oxygen sensing Slice
PID	Proportional - Integral - Derivative
PPE	Personal protective equipment
PWM	Pulse width modulation
RLHT	Relay heating element controller Slice
ROS	Robot operation system
RPi	Raspberry Pi computer
SCADA	Supervisory Control and Data Acquisition
UART	Universal asynchronous receiver / transmitter protocol
YPD	Yeast peptone dextrose

Chapter 1

1 Introduction

When choosing a system to automate industrial pilot projects or lab experiments, scientists and researchers use supervisory control and data acquisition (SCADA) systems to monitor critical parameters like temperature and control actuators like heaters to meet desired setpoints. Existing SCADA systems are proprietary and expensive like the National Instruments' (NI) CompactRIO system [1] and Opto 22's groov EPIC system [2]. Due to their high costs, they are often difficult to access in low-resource settings. Additionally, the internal circuitry and software of these devices are often hidden from the user which makes them fully reliant on the customer service and repair policies of the company to repair their devices or fix software bugs. In some cases, users will need to purchase an entirely new device. The alternatives to these devices are open source systems that give users free access to the circuit schematics, software, and other documents that could be useful to build their device. Beyond the significant cost savings that often motivate the switch to an open source alternative, there are three other benefits/pillars that are underscored in this work:

- **Repairability**: with full knowledge of the internal circuitry and software, users can replace broken components, fix software bugs, and learn from the community of users.
- **Customizability**: additional functionality can be added, and I/O can be repurposed for new applications.
- **Interoperability**: support for new devices, both open source and proprietary, can be added through hardware or software modifications.

The goal of this research was to design, test, and validate an open source SCADA system that meets the pillars of open source in science, previously mentioned, and explore the applications of this system by implementing it in industrial pilot projects and the automation of experiments.

The Broadly Reconfigurable and Expandable Automation Device (BREAD) system was created to as an open source project to address the needs of researchers with plug-and-play functionality at a fraction of the cost of its proprietary equivalents [3]. At the start of this research, BREAD had been tested as a data acquisition (DAQ) system; however, there were a limited number of Slices that could be used for SCADA applications. This work expanded on the existing BREAD framework and transformed it into a functioning SCADA system for scientific applications. In each of the three applications explored, the BREAD system was improved leading to the latest design outlined in Chapter 4.

In Chapter 2, the potential of BREAD as a small-scale SCADA system was first demonstrated by comparing it to a commercial heating controller used to control 6 heating zones in a pyrolysis reactor [4]. While the BREAD controller was slightly less accurate, it still produced equivalent products both in yield and quality. Additionally, BREAD provided data logging functionality which was not present on the commercial heating controller and was significantly less expensive. This first iteration of BREAD (Figure 1.1) was intended to test the communication and software and identify any limitations. Often Slices would disconnect due to loose connections and exposed circuitry was susceptible to debris. These issues were improved in the next iteration.

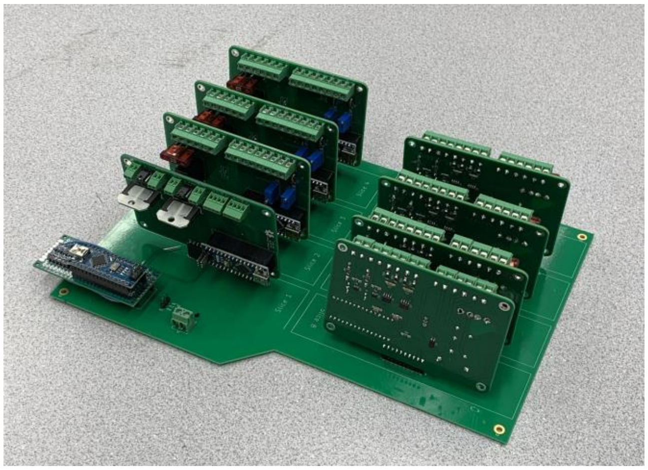
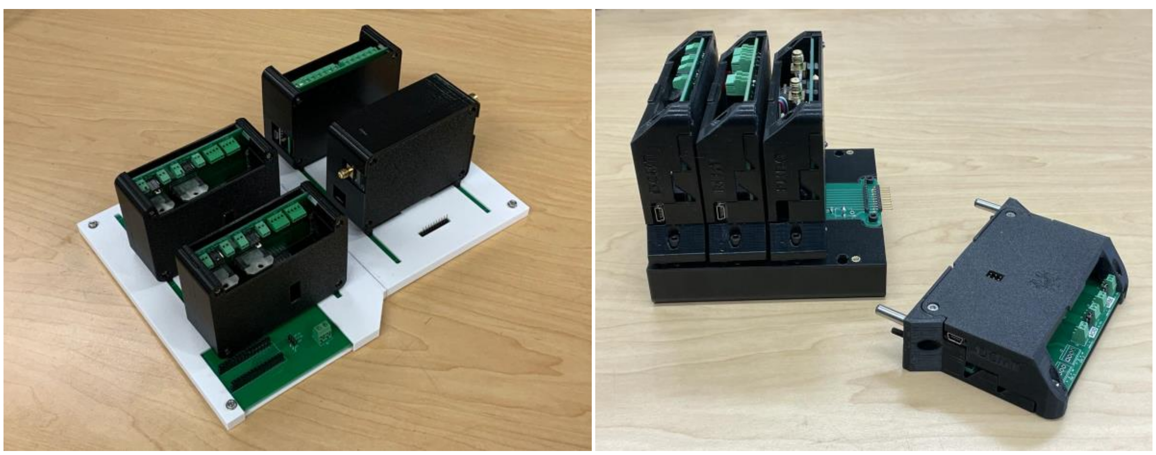


Figure 1.1: BREAD Iteration 1: Pyrolysis Reactor Controller

In Chapter 3, a new BREAD system was made for an open source bioreactor at a significant decrease in cost when compared to commercial alternatives. Accurate pH and heating control were among some of the functions added to BREAD. The heating and pH control was validated through growing yeast. A new Slice was designed to monitor pH and dissolved oxygen by integrating circuits from Atlas Scientific. This new Slice highlighted the interoperability of BREAD by demonstrating how existing devices can be integrated. The full BREAD system, paired with the OpenReactor software, provided a powerful tool for researchers to control and monitor each Slice and perform biological growth experiments. Two iterations of BREAD were made during this project (Figure 1.2). These designs improved the robustness of the Slice connections, introduced a new mounting method for the Loaf, and allowed multiple Loafs to be connected. However, there were still some drawbacks. The OpenReactor software setup was difficult for users with no Linux terminal experience; The Raspberry Pi and BeagleBone used to run the software were in limited supply within Canada at the time; and the Loaf mounting method was not standardized for a typical electrical enclosure.



(a) (b)

Figure 1.2: BREAD Iteration 2 (a) & 3 (b): Bioreactor Controller

In Chapter 4, the design of BREAD was changed to have the typical features of a SCADA system. DIN rail mounting and a new Loaf Controller, were among some of the changes that were made. Additionally, with a new Loaf Controller, new software, named "Butter", was created which was simpler to use than the OpenReactor software. This new software provided greater flexibility on how sensor data was displayed and gave feedback on control points with informative gauges. The new design was integrated into a pH control system that could be easily set up by scientists and, potentially, sold as a product. This pH controller used a new Slice design and used a modified version of Butter specifically for pH control and calibration. The controller was compared with a commercial alternative and provided better precision and data logging functionality at a 63% reduced cost. The latest iteration of BREAD is shown in Figure 1.3.

Figure 1.3: BREAD Iteration 4

1.1 References

[1] "CompactRIO Systems (cRIO)." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en/shop/compactrio.html

[2] "Opto22 - groov EPIC System." Accessed: May 01, 2024. [Online]. Available: https://www.opto22.com/products/groov-epic-system

[3] S. Oberloier, N. G. Whisman, F. Hafting, and J. M. Pearce, "Open source framework for a Broadly Expandable and Reconfigurable data acquisition and automation device (BREAD)," HardwareX, vol. 15, p. e00467, Sep. 2023, doi: 10.1016/j.ohx.2023.e00467.

[4] F. K. Hafting et al., "Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics," Electronics, vol. 12, no. 24, Art. no. 24, Jan. 2023, doi: 10.3390/electronics12244893.

Chapter 2

2 Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics

Industrial pilot projects often rely on proprietary and expensive electronic hardware to control and monitor experiments. This raises costs and retards innovation. Open-source hardware tools exist for implementing these processes individually; however, they are not easily integrated with other designs. The Broadly Reconfigurable and Expandable Automation Device (BREAD) is a framework that provides many open-source devices which can be connected to create more complex data acquisition and control systems. BREAD is plug-and-play hardware because users can reconfigure their control system by swapping Slices with little reprogramming needed. This article explores the feasibility of using BREAD plug-and-play open hardware to quickly design and test monitoring and control electronics for an industrial materials processing prototype pyrolysis reactor. Generally, pilot-scale pyrolysis plants are expensive custom designed systems. The plug-and-play prototype approach was first tested by connecting it to the pyrolysis reactor and ensuring that it can measure temperature and actuate heaters and a stirring motor. Next, a single circuit board system was created and tested using the designs from the BREAD prototype to reduce the number of microcontrollers required. Both open-source control systems were capable of reliably running the pyrolysis reactor continuously, achieving equivalent performance to a state-of-the-art commercial controller with a ten-fold reduction in the overall cost of control. Open-source, plug-and-play hardware provides a reliable avenue for researchers to quickly develop data acquisition and control electronics for industrial-scale experiments.

2.1 Introduction

Following the trend of accelerated innovation [1] that drove the success [2] of free and open-source software [3], free and open-source hardware (FOSH) [4,5] appears to be trailing adoption velocity by roughly 15 years [6]. An area of FOSH that has expanded rapidly is open-source electronics [7]. This is readily observed by a rapidly expanding global library of low-cost and high-quality libre electronic designs for science equipment

[8]. For example, outside of the expected open electronics for teaching electrical engineers [9,10], there are devices used to aid in electrical engineering research, such as power monitoring [11] and phasor measurement [12]. Open-source electronics systems have been developed to accurately measure gas pressures [13] and properties [14]. In addition, the approach has been used for such diverse and complex fields as smart converters and cloud connectivity [15], medical devices [16] like those for neuroscience [17,18], conventional [19,20] and indoor agriculture [21], electrophoresis [22], nuclear physics [23] and environmental monitoring [24,25]. The latter in particular has enabled open-source platforms for the undertaking of research-grade weather monitoring [26,27] and citizen environmental science [28,29]. In general, although open-source data acquisition (DAQ) systems have been developed for specific applications, including wire arc additive manufacturing [30], more general systems [31–33] and the systems in [13–29], the majority of science is still accomplished with closed source, proprietary systems. For example, National Instruments cDAQ [34] systems that are flexible, modular and operate as plug-and-play devices are widely popular but can be prohibitively expensive (~$1000 USD for a chassis and from $138 USD to $2846 USD per function card). Complex systems that need many actuators or sensors can cost tens of thousands of dollars. Such costs limit access to high-quality DAQ for those working in science and engineering in low-resource settings [35]. In many (maybe even most) cases the functions executed by the cDAQ cards could be carried out by an open-source alternative; however, as desired function count increases, the simplicity of integrating the designs decreases substantially. To overcome this challenge a new open-source electronics platform with plug-and-play functionality has been developed called the Broadly Reconfigurable and Expandable Automation Device (BREAD) [36].

The BREAD framework has potential to be integrated into a pyrolysis reactor control system. The performance and economic feasibility of such an approach, however, has yet to be measured. Pyrolysis is a chemical process capable of converting waste plastic into their hydrocarbon components that can then be sold as feedstock for new plastics production [37]. The use of pyrolysis to upgrade plastic waste to fuels or value-added product is well established [38,39]. During pyrolysis, the plastic is thermally degraded at temperatures between 400–700 °C in an inert environment [40]. By controlling key operating parameters, such as temperature and vapor residence time, the reaction can be tuned to produce the desired hydrocarbon products ranging from gases (ethylene and propylene) to oils (gasoline and diesel range hydrocarbons) to waxes [41]. Both academic and industrial interest in plastic pyrolysis has increased in the past decade as a solution to the plastic waste crisis. Several pyrolysis pilot plants have been built across the world, ranging from scales of 1 kg/h (small research lab [42]) to scales of over 100 kg/h (industrial [43,44]).

To test the capabilities of BREAD, an open-source (OS) pyrolysis reactor control system was first prototyped with Slices to control heating and stirring and to monitor temperature. After validating its performance, a single circuit board was designed and tested with the same pyrolysis reactor using the readily available circuit board designs from BREAD. Validation experiments included the conducting of a pilot-scale pyrolysis experiment converting waste military polyolefin plastic into wax, oil, and gas product. BREAD was used to control the primary pyrolysis reaction for this experiment, which controls the product distribution by regulating the reaction temperature. Results were compared with a control experiment using a commercially available and proprietary controller with similar functionality in order to validate the OS controller performance. The goal of this work was to first create an inexpensive BREAD-based controller with equivalent functionality to the commercial controller, then further reduce costs by integrating all electronics into a single PCB.

2.2 Materials and Methods

2.2.1 Commercial Methods of Pyrolysis Reactor Control

The pilot pyrolysis reactor system (see Figure 2.1) was conventionally controlled with a commercial 1/8 DIN 7 Channel Universal Process Ramp and Soak Controllers, available at Omega Engineering for US$1123.50 [45]. This controller has the ability to provide ON/OFF and full PID control to 7 independent zones. Each zone can be programmed with a ramp/soak profile for heating or cooling outputs. Limitations of this controller include the ability to see only one zone at a time, the lack of automatic data recording, and the inability to reprogram the controller during an experiment without temporarily turning it off. The first two limitations make it extremely challenging to monitor temperature trends during an experiment, while the third makes it unsafe to adjust the setpoint of a control zone during an experiment. Monitoring temperature is important and has been done in several ways using open hardware [46].

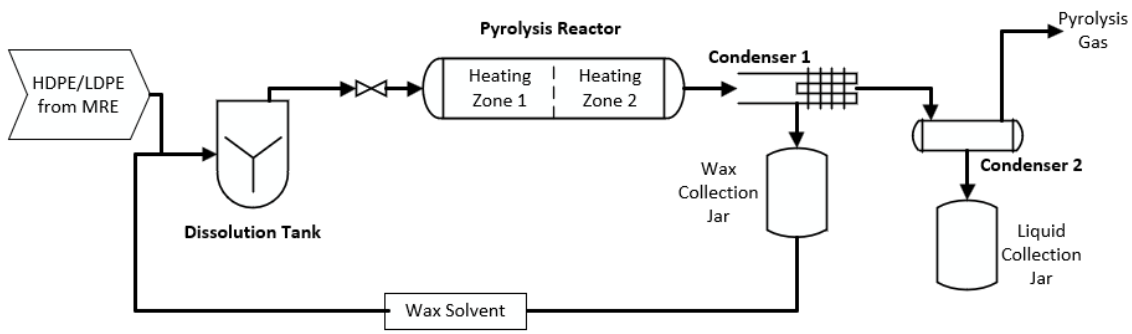


Figure 2.1: Process flow diagram of liquid-fed plastic pyrolysis. Waste polyethylene plastic enters the dissolution tank and is broken down in the primary and secondary pyrolysis reactors. A dual condenser system connects the wax, liquid, and gas product. MRE means meals-ready-to-eat bags.

2.2.2 Pyrolysis Reactor

The novel liquid-fed plastic pyrolysis system (invention disclosure at Michigan Technological University, Office of Innovation and Commercialization, Houghton, MI, USA) used in this work contains three major unit operations: a dissolution tank, a pyrolysis reactor, and a series of condensers (see Figure 2.1). The dissolution tank uses a novel solvent to dissolve waste polyolefin plastic at 240 °C to produce a homogenous liquid feed [42]. This liquid feed is fed into the pyrolysis reactor at a maximum rate of 1 kg/h, where primary pyrolysis occurs at 460 °C within heating zone 1. During primary pyrolysis, the polyolefin plastic is broken down into hydrocarbons of various chain lengths via a random scission reaction mechanism [47]. These hydrocarbon vapors, which are primarily high molecular weight waxes, flow into heating zone 2 where they are further broken down into liquid and gas range hydrocarbons at 575 °C with a residence time of 1–3 s. The hydrocarbon waxes, liquids, and gases are collected in a dual condenser system, with the first condensing waxes at 150 °C and the second collecting liquid pyrolysis oil at 25 °C. Both condensers use compressed air as the cooling agent. Any inorganic fillers in the waste plastic (e.g., from U.S. military meals-ready-to-eat (MRE) polyethylene bags) that do not react are collected as char, where they are removed at the end of the experiment. The design, fabrication and operation of this system has been previously described in detail in Kulas et al. 2022 [8,42].

2.2.3 Open-Source Pyrolysis Reactor Control System with BREAD

To be successful for a wide range of pyrolysis systems, the control system needed to control seven heaters and a stirring motor and be able to monitor and log temperatures at 11 different locations. At the time of testing, however, the control system requirements changed and only 6 heaters, 6 thermocouples, and 1 motor were required (Figure 2.2). Logging temperature was useful during experimentation for data analysis and troubleshooting. To implement this control system, two Slices from the BREAD framework were chosen: the DC motor Slice (DCMT) and the relay heater Slice (RLHT). The BREAD framework contains many open hardware designs which use Arduino Nano microcontrollers to communicate with an embedded Linux board like a Raspberry Pi running the OpenReactor software v1 [48] (Figure 2.3). Each Slice is connected over an I2C bus, so the system can be expanded for more complex reactors by simply connecting more Slices and assigning them a unique I2C address. It should be noted that the I2C bus is able to support multiple devices as both leader/follower, but that every I2C device on the I2C bus must have a unique address, which creates a limitation due to the address space limit of 128 unique addresses.

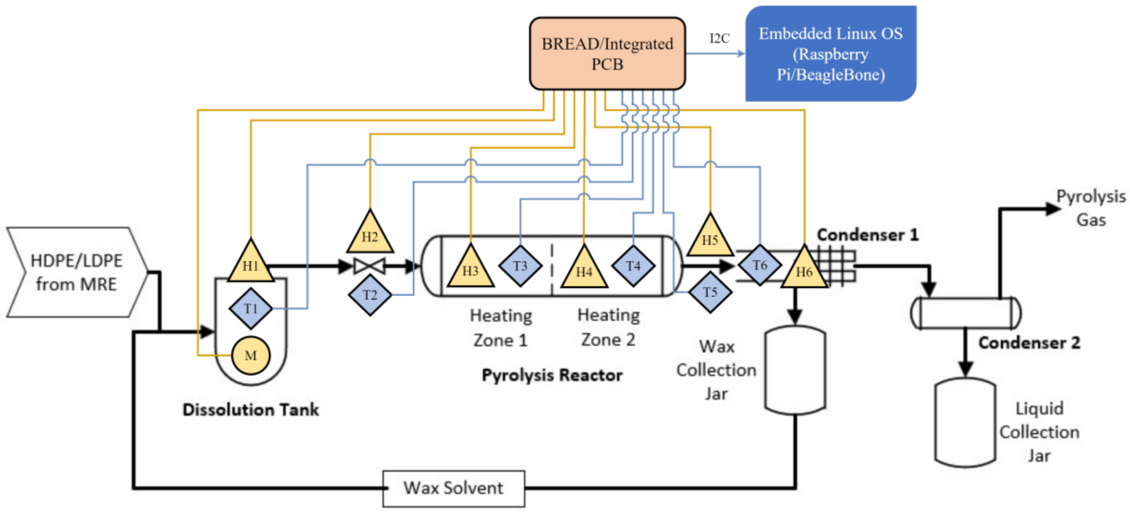


Figure 2.2: Pyrolysis system diagram (M: motor, T: thermocouple, H: heater). The arrows indicate the material flow paths.

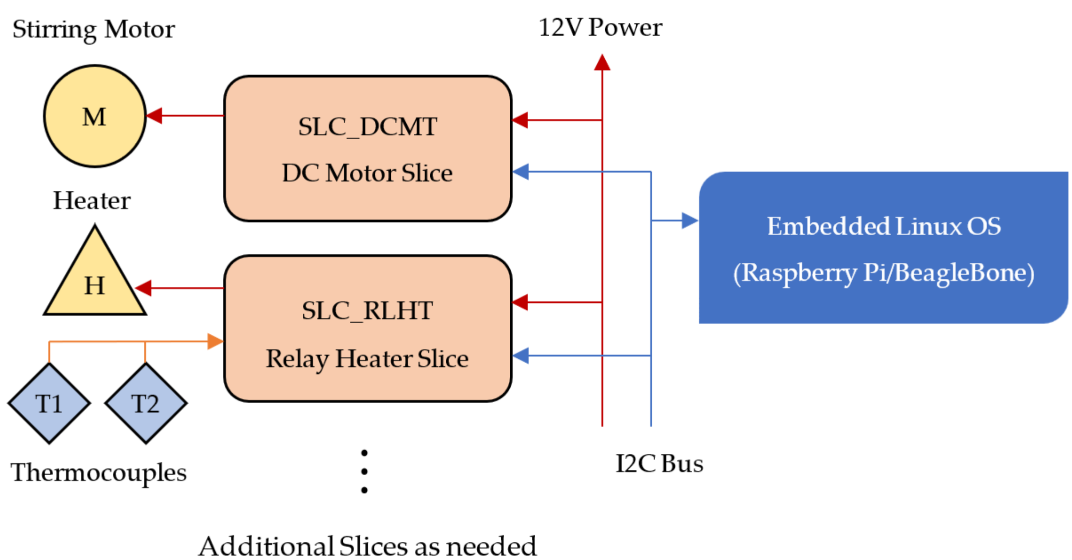


Figure 2.3: System connection diagram. Arrows indicate the direction of information (blue) and power (red).

Each relay heater Slice can control a single heater (maximum of 10A @ 250VAC or 5A @ 30VDC) and monitor 2 k-type thermocouples and 2 thermistors (Figure 2.4). The DC motor Slice can control the speed and direction of two DC motors (maximum of 3A @

12V) (Figure 2.5). The 10-pin connector linked each Slice to the Loaf backplane, provided 12V of power, and enabled I2C communication to the Raspberry Pi. Connection of the Loaf to the Raspberry Pi was undertaken via the I2C port (Figure 2.6 and Figure 2.7). For more technical details on the BREAD framework, one should review the original BREAD publication [36].

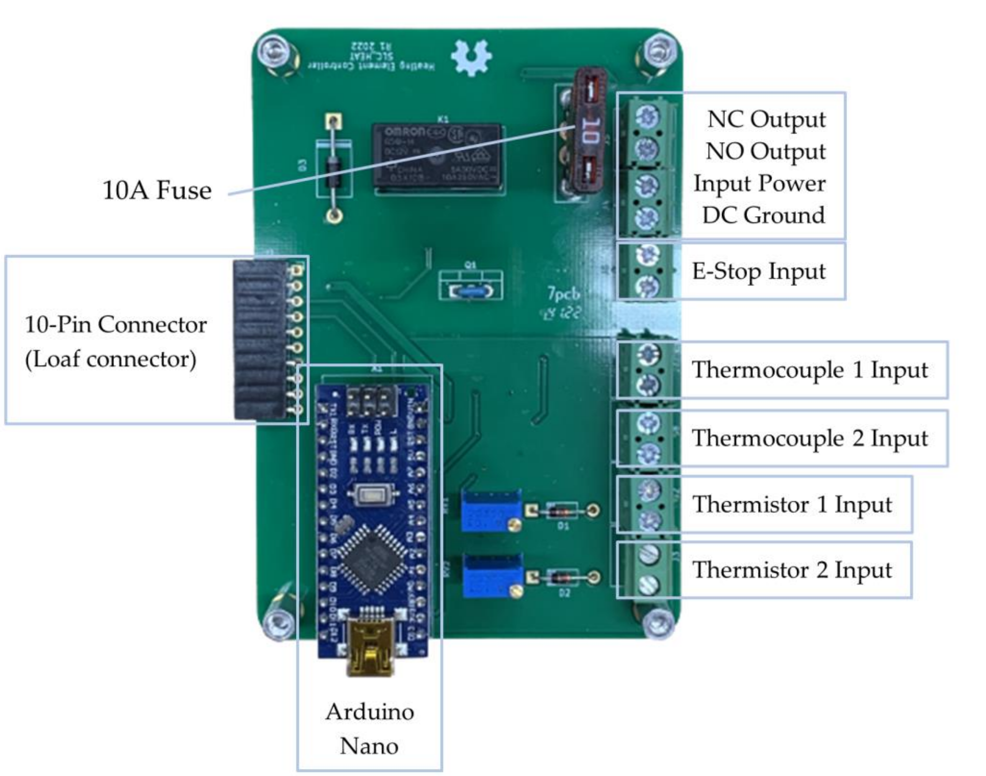


Figure 2.4: SLC_RLHT connection diagram (when using the relay as a switch, only use input power and the normally closed (NC) output or the normally open (NO) output).

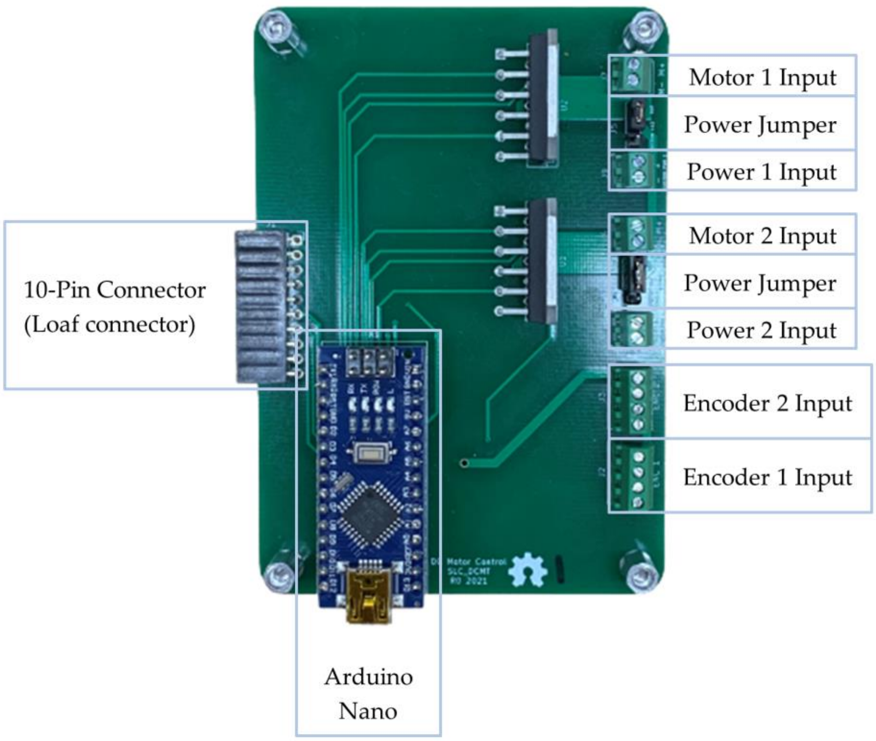


Figure 2.5: SLC_DCMT connection diagram

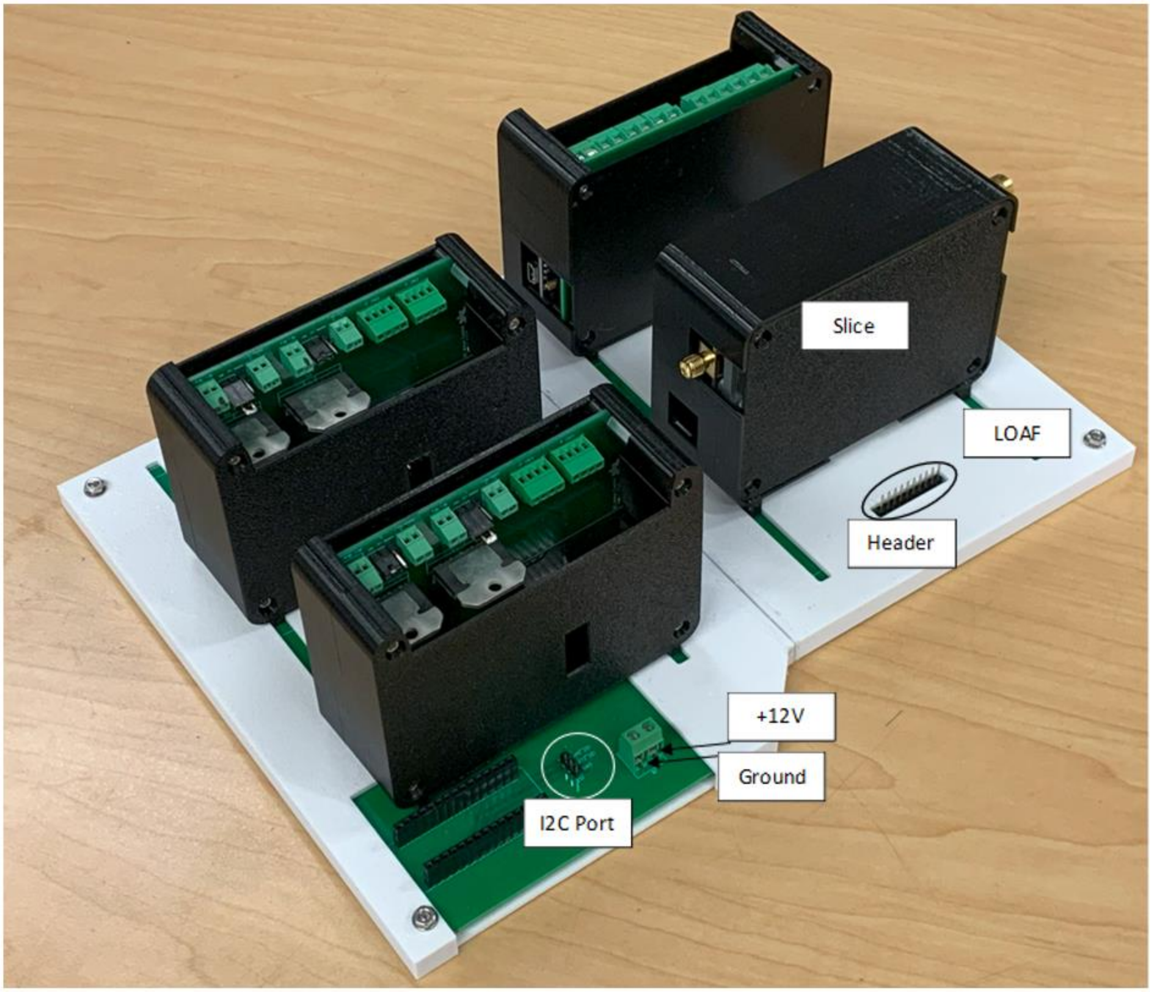
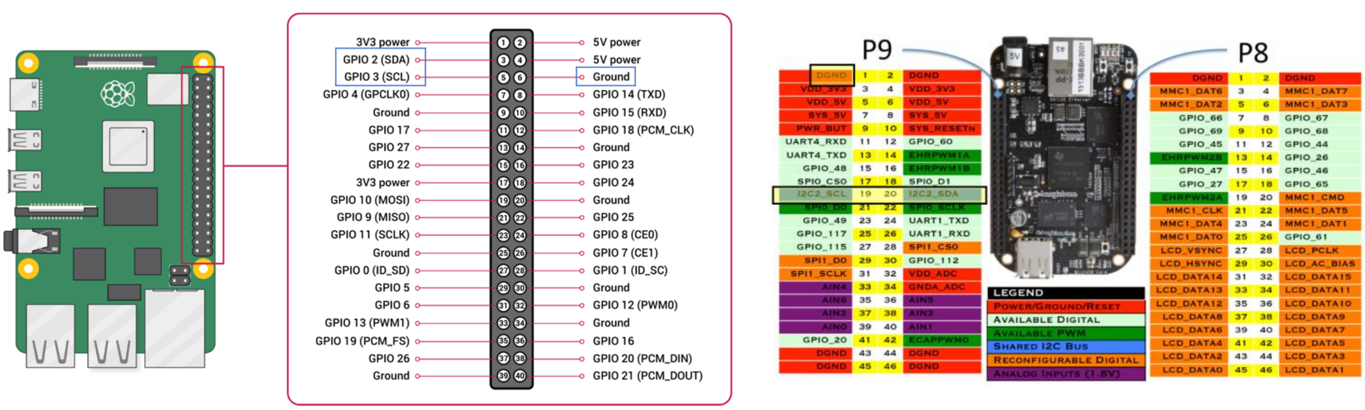


Figure 2.6: Loaf connection diagram



(a) (b)

Figure 2.7: Linux board connection pins (highlighted): (a) Raspberry Pi; (b) Beaglebone Black.

All PCB design files as well as the case models for the Slices and the Loaf can be found in Table 2.1. All electrical components for each Slice and Loaf were sourced from Digikey and can be found in Appendix A, with the total cost of the BREAD system in Table 2.2. The procedure for setting up a new Slice is detailed in Appendix B.

Table 2.1: Design file links

Files	URL (accessed Dec. 2, 2023)
BREAD SLC_RLHT	https://osf.io/pf6gy/
BREAD SLC_DCMT	https://osf.io/6aw9m/
Integrated pyrolysis board	https://osf.io/3ugbn/

Table 2.2: BREAD pyrolysis system cost (in CAD).

Component	Number	Cost Per Unit	Total Cost
SLC_RLHT	7	$68.26	$477.82
SLC_DCMT	1	$84.55	$84.55
LOAF	1	$16.69	$16.69
		Total	$579.06

2.2.4 Integrated Single-Board Design

To reduce the number of components needed to control the heating zones and monitor temperature, the thermocouple and relay components from the RLHT Slices were incorporated onto a single PCB along with a single motor driver (Figure 2.8). This reduced costs by eliminating the thermistor components and the additional motor control components. The new board can also control up to 10 heaters as it was not restricted to a maximum of 8 Slices, in contrast with the BREAD system which uses 7 RLHT and 1 DCMT. Arduinos 1–3 control three sets of heaters and monitor three thermocouples each. Arduino 4 controls heater 10 and thermocouples 10 and 11. Arduino 5 controls the DC motor. Each Arduino was treated as an individual Slice with additional I/O and a unique I2C address. Thus, only minor additions to their firmware were needed (added additional heaters and thermocouples or removed additional motors). The Slice definitions in the software were changed in a similar fashion. The design file links should be consulted for more information.

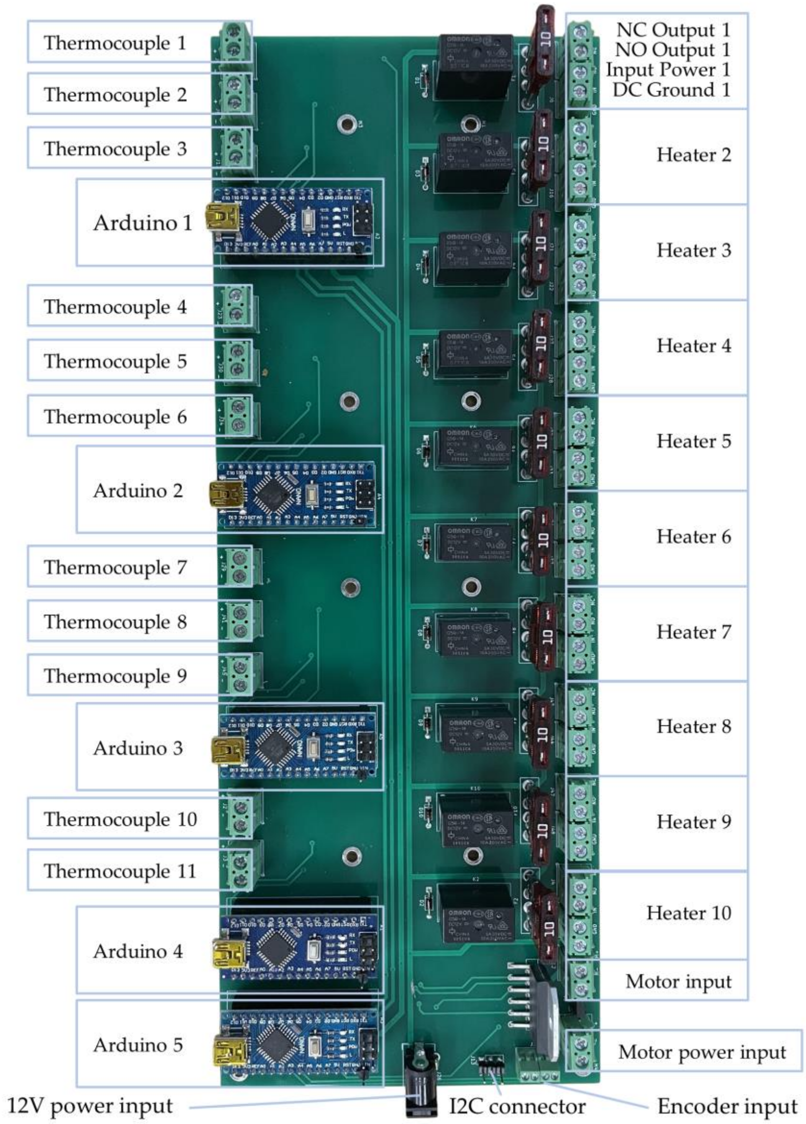


Figure 2.8: Integrated pyrolysis board connection diagram.

As shown in Table 2.3, the cost of the single board design was greatly reduced when compared with the BREAD system.

Table 2.3: Integrated board bill of materials (Cost in CAD, sourced from Digikey).

Component	Number	Cost Per Unit	Total Cost
Arduino Nano	5	$11.33	$56.65
Barrel Jack (12V 5A)	1	$1.18	$1.18
Capacitor 10nF	14	$0.27	$3.78
Capacitor 10uF	3	$0.81	$2.43

Diode	10	$0.14	$1.40
Fuse Holder 10A	20	$0.31	$6.20
01 × 03 Male Header	2	$0.13	$0.26
Resistor 2.2k	1	$0.15	$0.15
Ferrite Bead	22	$0.40	$8.80
Relay SPDT 12V G5Q-1	10	$2.30	$23.00
MAX6675ISA+	11	$15.85	$174.35
TCMT1100	10	$0.95	$9.50
AP1117-50	1	$0.57	$0.57
LMD18200	1	$29.71	$29.71
Screw Terminal 01x02	33	$0.81	$26.73
Screw Terminal 01x04	1	$2.00	$2.00
Automotive Fuse 10A	1	$1.64	$1.64
		Total	$348.35

2.2.5 Validation Tests

The OS controller was used to control the temperature of heating zone 1 within the pyrolysis reactor during a pyrolysis experiment at 460 °C. The pyrolysis reaction was run for 80 min at a feed flow rate of 730 g per hour. The feed composition for the reaction was 25% HDPE, 25% LDPE, and 50% pyrolysis wax solvent. The HDPE and LDPE were sourced from meals-ready-to-eat (MRE) plastic bags, a complex waste plastic that is not normally recycled. Results from the pyrolysis experiment were compared with an identical experiment conducted with the commercial controller used to control the primary pyrolysis reactor and all other process units, with the goal of seeing equivalent performance be-tween the two controllers and experiments.

2.3 Results

Three criteria were used to analyze the pyrolysis experiment and performance of the two controllers (proprietary vs. OS): temperature control, product yields, and product quality.

2.3.1 Pyrolysis Temperature Control

Both OS systems were controlled using different versions of the OpenReactor software linked in Table 2.4.

Table 2.4: Open Source Pyrolysis Software

System	Software URL
BREAD	https://gitlab.com/mtu-most/most_openreactor/-/tree/pyrolysis
Integrated board	https://gitlab.com/mtu-most/most_openreactor/-/tree/integrated_pyrolysis2

Details on how to set up the software are outlined in the repository. During testing it was observed that the software would slow down after accumulating many data points. The temperature readings were sampled at 10 second intervals to reduce the number of data points logged by the software.

In order to reach a performance comparable to the commercial controller, the OS controller was manually tuned using the Ziegler–Nichols method at 460 °C in order to determine the tuning constants for PI control [49]. This was accomplished by tuning the ultimate gain, K_u, until the temperature reached a periodic oscillation with period T_u. The gains can then be calculated for PI control as:

$$K_P = 0.45K_u \tag{1}$$

$$K_I = 0.54\frac{K_u}{T_u} \tag{2}$$

The measured controller parameters are shown in Table 2.5.

Table 2.5: PID Tuning Parameters

Parameter	Value
K_u	64
T_u	150s
K_P	28.8
K_I	0.23

After tuning the system, the temperature of heating zone 1 (see Figure 2.1) was controlled at a setpoint of 460 °C during the pyrolysis reaction. A custom immersion cartridge heater (BriskHeat) was located inside a cylindrical stainless-steel reactor and the temperature was measured in the center of the chamber with a type K thermocouple from Omega Engineering. The internal temperature of the immersion heater was also monitored during the reaction as a safety precaution to ensure the control system was working as designed. Figure 2.9 compares the measured temperature for the two experiments to the setpoint. As expected, the commercial controller always kept the temperature within ±1–2 °C of the setpoint. The OS controller was comparable with a variation of ±3 °C from the setpoint and a slight bias of −1 °C. To understand if this is an acceptable margin, the yield and compositional quality of the pyrolysis products were also compared.

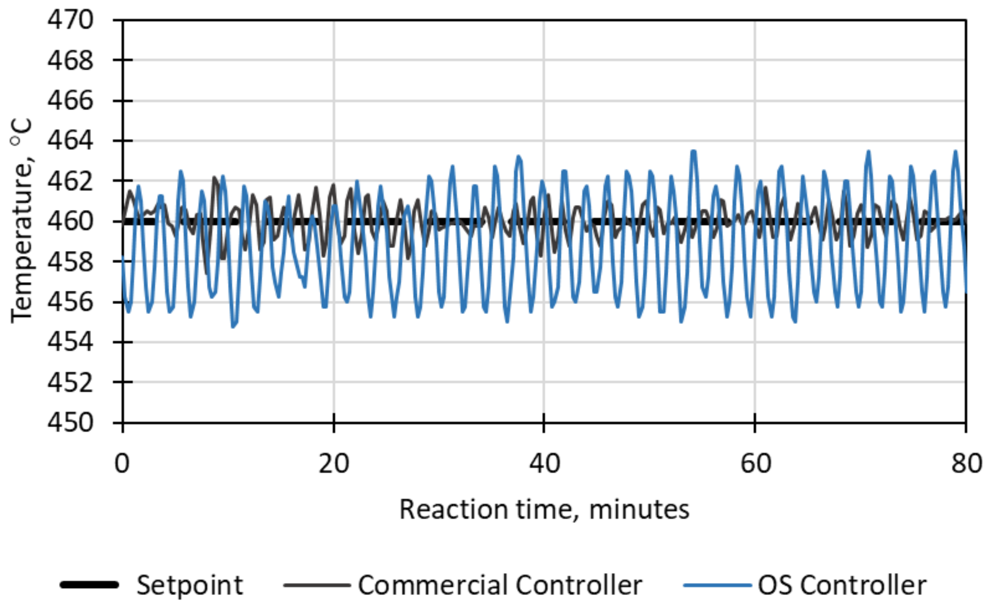


Figure 2.9: Temperature traces for the inside of heating zone 1 of the pyrolysis reactor over an 80 min pyrolysis reaction for both a commercial controller and the OS controller at identical operating conditions and at a temperature setpoint of 460°C and feed flow rate of 0.7 kg/h. The average absolute error is 0.49% for the OS controller and 0.14% for the commercial controller.

2.3.2 Product Yields

After performing the experiment, three products were produced: hydrocarbon wax, oil, and gas. The wax and oil products were collected using a dual condenser system and are shown in Figure 2.10B. The oil product consists of primarily C6–C15 alkenes and is a yellow liquid at room temperature. The wax product consists primarily of C15–C30 alkenes and alkanes and is a tan solid at room temperature. A char residue is formed from the inorganic nanoclay filler material in the feed plastic (Figure 2.10A). Overall, the product distribution for the two pyrolysis experiments is remarkedly consistent (Figure 2.11). The OS controller produced 30.8 wt.% gas, 5.1 wt% liquid, 64.1 wt.% wax, and 1.6 wt.% char while the commercial controller produced 31.3 wt.% gas, 5.1 wt% liquid, 63.6 wt.% wax, and 1.6 wt.% char. This product distribution, shown in Figure 2.10, seems to validate the performance of the OS controller, however, the product quality must also be tested.

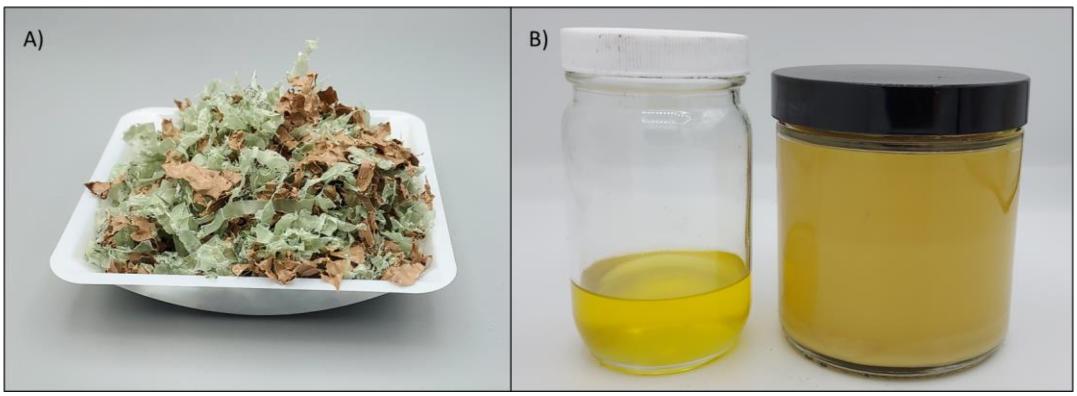


Figure 2.10: Shredded waste MRE plastic (A) is broken down into oil (left (B)), wax (right (B)), and gas (not pictured).

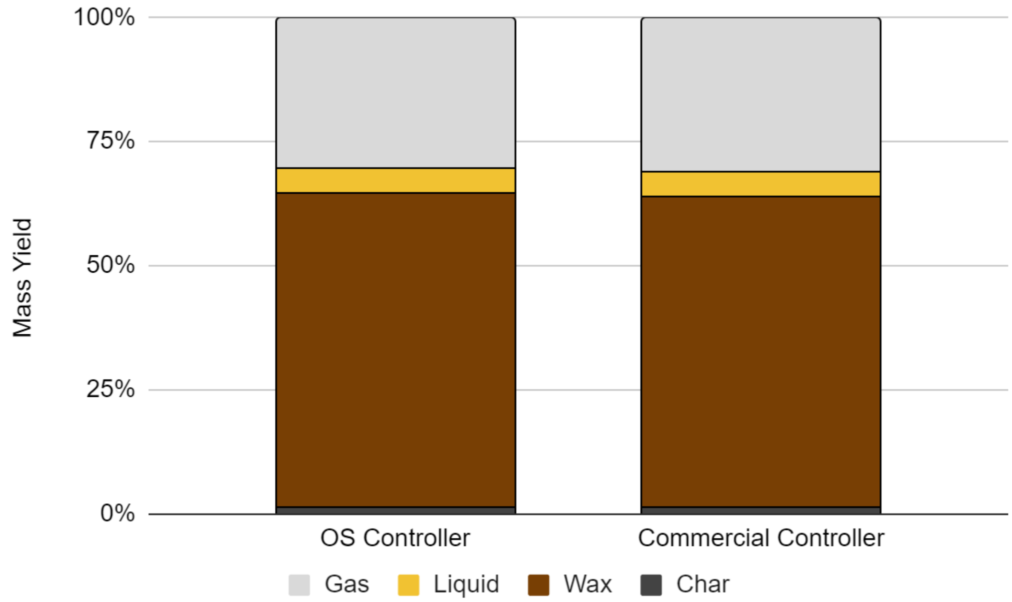


Figure 2.11: Pyrolysis mass yields for experiments using OS and commercial controller.

2.3.3 Pyrolysis Product Quality

The quality of the three collected products—wax, oil, and gas—was measured using gas chromatography–mass spectroscopy (GC–MS) (Thermo Scientific TRACE 1310 Gas Chromatograph in sequence with ITQ 110 Ion Trap MS). The methods for the GC–MS analyses have been previously published [42] and are capable of detecting alkanes, alkenes, and alkadienes from C6–C30. In this work, the GC–MS chromatograms for pyrolysis oil (Figure 2.12) and wax (Figure 2.13) are qualitatively and quantitatively compared for each controller. The pyrolysis oil produced by both controllers contains primarily alkenes from C6 to C15 (Figure 2.12) while the wax contains a mix of alkenes and alkanes from C15 to C30 (Figure 2.13). For both products, the chromatograms from each experiment are very similar, proving that the OS controller is capable of producing products with the same compositional quality as the commercial controller. These qualitative results were confirmed quantitatively by comparing the peak areas of the identified compounds (see Table 2.11 in Appendix B). The average absolute error for the identified peak areas is 6.4% for wax, 9.8% for oil, and 11.6% for the gas product, confirming that the products produced in the two experiments are equivalent in composition.

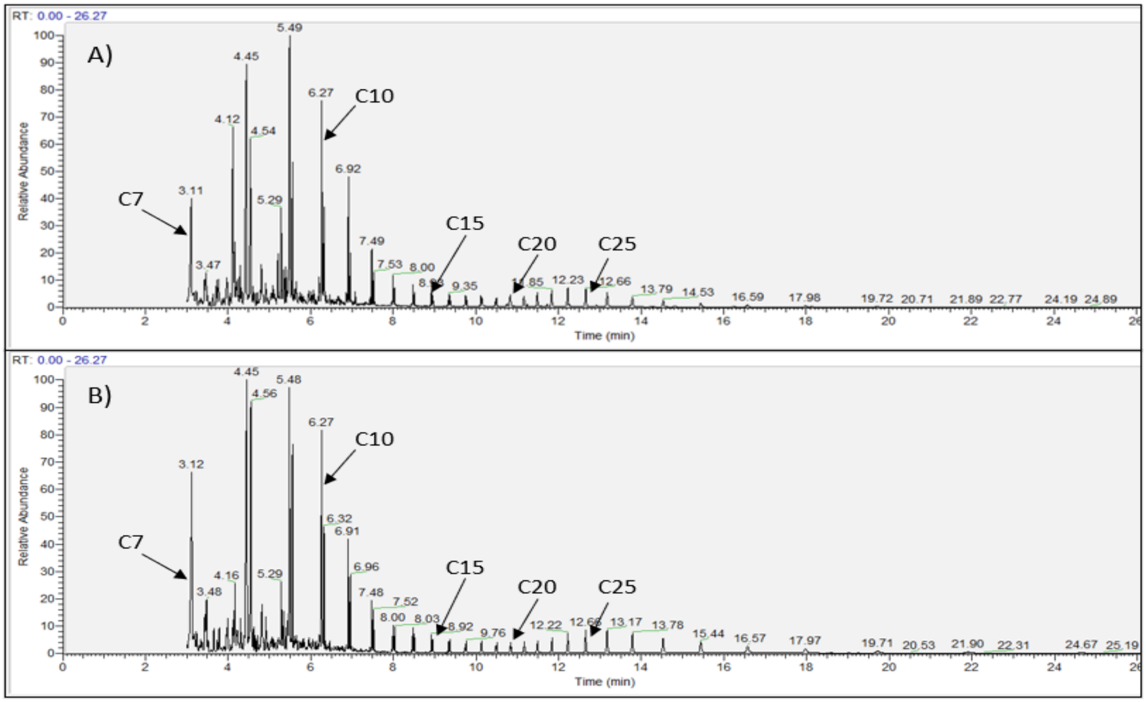


Figure 2.12: GC–MS chromatogram of pyrolysis oil produced using the OS control system (A) and the commercial control system (B). Key peaks of interest are labeled, while unlabeled peaks are one carbon number apart from each other.

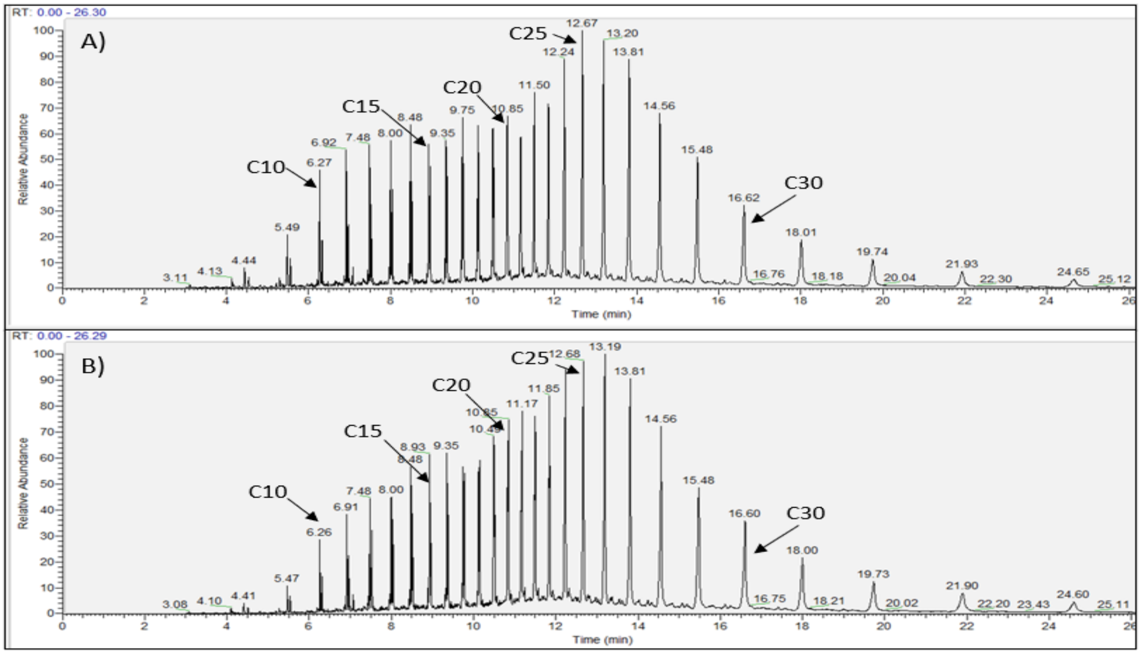


Figure 2.13: GC–MS chromatogram of pyrolysis wax produced using the OS control system (A) and the commercial control system (B). Key peaks of interest are labeled while unlabeled peaks are one carbon number apart from each other.

2.4 Discussion

2.4.1 Implications

The approximately equivalent commercial pyrolysis systems cost either $1123.50 for seven control channels (less functionality than the OS system) or $6000 for eight control channels (the same functionality as the OS system) [45,50]. The OS BREAD-based system that can be built for under $350 integrated or under $580 as separated Slices offers savings of more than a factor of ten and clearly makes pyrolysis control more accessible. There are other commercial controllers available that are reprogrammable and have temperature re-cording capabilities at costs of US$3041–US$3714 for four control zones [50]. This Model Quad Controller from KEM Scientific has accompanying computer software that is capable of recording temperature data and changing the setpoint during an experiment without having to turn off the controller. Again, the OS system is significantly less expensive, but does require fabrication and assembly.

It should be noted that the costs of the OS system shown here only include the material costs. Labor costs need to be taken into account for assembling the BREAD system to make a complete comparison; however, zero labor costs are appropriate for several situations including: (1) where the assembly of the OS system is used as an educational tool providing students with experience in fabrication of open-source scientific hardware; (2) when the labor is provided by anyone not salaried or paid direction (e.g., interns or volunteers); or, (3) where the opportunity cost is zero to use an existing salaried employee. This latter is true for individuals (e.g., citizen scientists) that normally do not calculate their opportunity costs for fabricating their own equipment. In other cases, these opportunity costs will need to be calculated for decision makers in their own context. Overall, it is clear that the economic savings for the materials provide a much greater accessibility to the device that is currently available from proprietary systems.

This is consistent with the literature, as the use of open hardware is often related to cost savings when it replaces proprietary electronics for DAQ and control applications [5,7,51]. While BREAD-based DAQ systems are already significantly less expensive than proprietary systems like National Instruments cDAQ [34], custom open hardware like the pyrolysis system developed in this report further increase this cost difference when compared with proprietary pyrolysis reactor control systems. This is consistent with the open scientific hardware literature in general [8,52–55], in open-source electronics [56–58], and in electronics for other chemical processes [59–62]. Compared with high-cost proprietary controllers, the ease of BREAD and the functionality are clear. The programmable and adaptable nature of BREAD allow it to overcome the limitations of commercial controllers, such as the lack of automatic recording of data and the inability to reprogram the controller during an experiment without temporarily turning it off. Finally, the system's ability to log data and monitor temperature trends in real time allows researchers and students to better understand the pyrolysis process. The low cost of manufacturing for the OS BREAD system enables it to be used in education, which is also consistent with other open-source electronics devices in the literature [63,64].

2.4.2 Limitations and Future Work

While a single circuit board decreases costs and improves reliability by permanently connecting all peripherals, it lacks ease of assembly. The integrated control system covered in this report took days to assemble by hand and had some design issues which needed to be fixed. Unfortunately, simple mistakes can be common when designing PCBs and they are not easily diagnosed until the board is fully assembled. Design iterations increase the cost and time to develop electronic hardware, so a BREAD-based design is more cost effective for experimentation and small-scale industrial applications. Nevertheless, both solutions have demonstrated that open-source hardware can provide comparable performance to expensive commercial systems at a fraction of the cost while also being more customizable, serviceable, and modular. In addition, because of the open-source license of the system, anyone in the world may commercialize it and, with a substantial profit margin, still provide lower-cost, fully assembled systems to the scientific community following an open-source business model [65,66].

There are several areas of future work that could improve the system. First, the connection reliability of BREAD Slices could be improved and made more stable with 3D printed supports for the electrical connections. This could be done by improving the ease of connection between Slices and Loafs by implementing some system to guide the 10-pin connector to the correct location. In addition, future work could focus on implementing a connection between Loafs with additional 10 pin connectors to deliver power and communication. To make the systems easier to assemble the name of each Slice could be integrated into the CAD of the case so they can be easily identified. These cases could also be improved to ensure the hex standoff does not loosen during assembly. Future work could also implement a mounting solution so any BREAD system can be sturdily fixed to a surface and wires organized to improve safety. Multiple versions of the same Slice could also be developed with different components to aid in the component selection process during prototyping, particularly during supply chain disruptions. This would also eliminate delays due to any form of part shortages.

Finally, significant software adaptations could be made to improve ease of installation and provide an auto-tuning feature for PID control. Currently, the heaters are actuated with PID controls which must be tuned by hand. It takes many hours to tune a heating system by hand and often the results are suboptimal. Having an auto-tune feature would improve ease of integration and could potentially lead to a more accurate controller when compared with commercial alternatives. Additionally, understanding the heating and sensing latency of the RLHT Slice would improve the accuracy of the PID control algorithm. The software could also be augmented to make the GUI easier for non-experts to use so that there would be a user operator screen and a research operator screen, with the former using default settings for standard production and the latter having complete control of the system.

2.5 Conclusions

This study assessed the performance of an open-source pyrolysis control system using plug-and-play hardware from the BREAD framework and compared this with a seven channel Universal Process Ramp and Soak Controller from Omega Engineering. When testing the heating control of both systems at a constant 460 °C, the proprietary system had an average absolute error of 0.14% while the BREAD system was 0.49%. After performing a pyrolysis experiment and by measuring the yield, the results indicate that the BREAD framework can be used to make comparable control hardware at a fraction of the cost of a commercial proprietary system. In addition, BREAD provides functionality such as data logging, the ability to modify the temperature profile in real time, and the ability to expand the system to, for example, accommodate additional thermocouples and heaters. This is especially useful with experimental systems, such as the pyrolysis reactor explored in this paper, where control requirements, like the number of heaters and thermocouples, are constantly changing. Like many other rapid prototyping technologies where small batches can be made more efficiently than with mass production processes, BREAD can also be used as a rapid prototyping technology for electronic hardware.

The potential for BREAD to aid in PCB development was also explored by integrating the designs from BREAD Slices onto a single circuit board. While the final design further reduced the costs of the open-source controller, it took substantial time to assemble, which increased the overall costs.

While a BREAD-based controller can provide similar performance and additional functionality compared with a commercial system, there are still some aspects which make BREAD more difficult to use. Improving these limitations, as outlined in future work, would make the BREAD framework a more competitive and reliable choice for researchers. Ultimately, the BREAD framework has the potential to serve as a rapid prototyping platform for control electronics and a starting point for researchers designing their own control systems.

Appendix A: BREAD System Price Breakdown

Table 2.6: SLC_RLHT bill of materials (cost in CAD, sourced from Digikey)

Component	Number	Cost Per Unit	Total Cost
Arduino Nano	1	$11.33	$11.33
Capacitor 10uF	2	$0.81	$1.62
Capacitor 10nF	2	$0.27	$0.54
Zener Diode 5.1V	2	$0.42	$0.84
Diode	1	$0.14	$0.14
Fuse Holder 10A	2	$0.31	$0.62
Ferrite Bead	4	$0.40	$1.60
Female Header 01 × 10	1	$1.14	$1.14
Screw Terminal 01 × 02	6	$0.81	$4.86
Relay SPDT 12V G5Q-1	1	$2.30	$2.30
Resistor 10k	4	$0.15	$0.60
Potentiometer 10k	2	$3.73	$7.46
MCP6001U	2	$0.46	$0.92
MAX6675ISA+	2	$15.85	$31.70
TCMT1100	1	$0.95	$0.95
Automotive Fuse 10A	1	$1.64	$1.64
		Total	$68.26

Table 2.7: SLC_DCMT bill of materials (cost in CAD, sourced from Digikey).

Component	Number	Cost Per Unit	Total Cost
Capacitor 10nF	6	$0.27	$1.62
LMD18200	2	$29.71	$59.42
Arduino Nano	1	$11.33	$11.33
Resistor 2.2k	2	$0.15	$0.30
Screw Terminal 01 × 02	4	$0.81	$3.24

01 × 03 Male Header	2	$0.13	$0.26
Screw Terminal 01 × 04	2	$2.00	$4.00
Capacitor 10uF	4	$0.81	$3.24
01 × 10 Female Header	1	$1.14	$1.14
		Total	$84.55

Table 2.8: Loaf bill of materials (cost in CAD, sourced from Digikey).

Component	Number	Cost Per Unit	Total Cost
10uF	2	$0.81	$1.62
Arduino_Nano_v3.x	1	$11.33	$11.33
Conn_01 × 10_Male	8	$0.35	$2.80
Screw_Terminal_01 × 02	1	$0.81	$0.81
Conn_01 × 03_Male	1	$0.13	$0.13
		Total	$16.69

Appendix B: Setting Up a New Slice

When adding a new Slice to a network, the I2C address needs to be updated and the software v1 must be told how to handle the new Slice (i.e., specific commands, control program, etc.). Users must first familiarize themselves with the Arduino IDE [67]. Then, follow these steps to set up a new Slice:

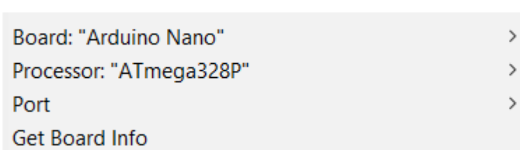
1. Open the .ino file included in the Firmware folder for the Slice's specific repository.

2. At the top of the program, change the I2C address to a number not used by the other Slices in the network.

```
#define I2C_ADR <new_number>
```

3. Connect the Arduino Nano via mini-USB cable to the computer.

4. Ensure the proper board, processor, and port are selected under "Tools".



5. Verify and upload the code.

6. Connect the Slice to the Loaf backplane.

For each Slice with a unique address, the software v1 must be told how to handle both the sensors and actuators that may be connected to a Slice. The relay heater Slice is used as an example below:

7. On the Linux board, open “devices.json”.

8. Each thermocouple and thermistor can be added by defining their specific parameters in the DEVICES section (Table 2.9).

Table 2.9: Sensor setup parameters.

Parameter	Value	Description
enabled	false	Starting configuration. Always set to false
input	0	Current sensor input
lastInput	0	Last sensor input
setPoint	0	Desired sensor input
kp	0.1	Proportional gain
ki	0.03	Integral gain
kd	0.03	Derivative gain
er	0	Difference between setPoint and input
thermocouple	1	Thermocouple monitoring heater temperature
control	“control.BREADheaterPID”	Python program for controlling heater

9. The heater actuator is added by defining its parameters in the CONTROL section (Table 2.10).

Table 2.10: Actuator setup parameters.

Parameter	Value	Description
enabled	false	Starting configuration. Always set to false
input	0	Current sensor input
lastInput	0	Last sensor input
setPoint	0	Desired sensor input

kp	0.1	Proportional gain
ki	0.03	Integral gain
kd	0.03	Derivative gain
er	0	Difference between setPoint and input
thermocouple	1	Thermocouple monitoring heater temperature
control	"control.BREADheaterPID"	Python program for controlling heater

10. Save "devices.json" after adding all sensors and actuators.

Table 2.11: GC–MS peak areas for all pyrolysis products from open source and commercial controller experiments. Each carbon number is primarily composed of alkenes with minor amounts of alkanes and alkadienes also present. The average absolute error between the product composition for the two experiments is 6.4% for the wax product, 9.8% for the oil product, and 11.6% for the gas product. Carbon numbers with peak areas below 3% were ignored when calculating the error due to instrument noise.

	Wax			Oil			Gas		
Carbon Number	Commercial	Open Source	% Absolute Error	Commercial	Open Source	% Absolute Error	Commercial	Open Source	% Absolute Error
2	0.0%	0.0%	-	0.0%	0.0%	-	12.0%	10.4%	14.4%
3	0.0%	0.0%	-	0.0%	0.0%	-	28.8%	28.5%	1.2%
4	0.0%	0.0%	-	0.0%	0.0%	-	34.7%	31.6%	9.3%
5	0.0%	0.0%	-	0.0%	0.0%	-	14.0%	19.6%	33.1%
6	0.0%	0.0%	-	0.0%	0.0%	-	9.3%	9.3%	0.2%
7	0.0%	0.0%	-	19.5%	19.4%	0.6%	1.2%	0.7%	-
8	0.3%	0.5%	-	27.5%	23.9%	14.2%	0.0%	0.0%	-
9	0.6%	1.0%	-	19.9%	22.8%	13.8%	0.0%	0.0%	-
10	1.2%	2.0%	-	9.7%	10.9%	11.0%	0.0%	0.0%	-
11	1.9%	2.6%	-	4.7%	5.2%	9.3%	0.0%	0.0%	-
12	2.3%	2.9%	-	2.2%	2.7%	-	0.0%	0.0%	-
13	2.6%	3.0%	-	1.5%	1.6%	-	0.0%	0.0%	-
14	2.9%	3.3%	12.2%	1.2%	1.2%	-	0.0%	0.0%	-
15	3.1%	3.4%	10.4%	0.9%	0.9%	-	0.0%	0.0%	-
16	3.2%	3.5%	8.6%	0.7%	0.8%	-	0.0%	0.0%	-
17	3.4%	3.6%	5.8%	0.7%	0.6%	-	0.0%	0.0%	-
18	3.5%	3.6%	2.0%	0.5%	0.7%	-	0.0%	0.0%	-
19	3.7%	3.8%	2.6%	0.5%	0.5%	-	0.0%	0.0%	-
20	4.2%	4.1%	3.4%	0.5%	0.9%	-	0.0%	0.0%	-

21	4.5%	4.1%	8.6%	0.5%	0.7%	-	0.0%	0.0%	-
22	4.8%	4.5%	6.3%	0.5%	0.7%	-	0.0%	0.0%	-
23	5.2%	4.8%	8.0%	0.7%	0.8%	-	0.0%	0.0%	-
24	6.3%	5.2%	19.6%	0.8%	1.0%	-	0.0%	0.0%	-
25	6.2%	5.9%	4.1%	0.9%	1.0%	-	0.0%	0.0%	-
26	6.9%	6.8%	2.2%	1.1%	0.9%	-	0.0%	0.0%	-
27	7.0%	6.9%	1.4%	1.2%	0.9%	-	0.0%	0.0%	-
28	6.5%	6.5%	0.8%	1.1%	0.5%	-	0.0%	0.0%	-
29	5.5%	5.5%	0.9%	1.0%	0.4%	-	0.0%	0.0%	-
30	4.9%	4.3%	13.6%	0.8%	0.3%	-	0.0%	0.0%	-
31	3.3%	3.2%	3.7%	0.6%	0.2%	-	0.0%	0.0%	-
32	2.3%	2.3%	-	0.5%	0.3%	-	0.0%	0.0%	-
33	2.2%	1.7%	-	0.2%	0.1%	-	0.0%	0.0%	-
34	1.5%	1.2%	-	0.2%	0.1%	-	0.0%	0.0%	-

2.6 References

[1] Kogut, B.; Metiu, A. Open-Source Software Development and Distributed Innovation. *Oxf. Rev. Econ. Policy* **2001**, *17*, 248–264. https://doi.org/10.1093/oxrep/17.2.248.

[2] Weber, S. The Success of Open Source. In *The Success of Open Source*; Harvard University Press: Cambridge, MA, USA, 2005; ISBN 978-0-674-04499-9.

[3] Robles, G.; Amor, J.J.; Gonzalez-Barahona, J.M.; Herraiz, I. Evolution and Growth in Large Libre Software Projects. In Proceedings of the Eighth International Workshop on Principles of Software Evolution (IWPSE'05), Lisbon, Portugal, 5–6 September 2005; pp. 165–174.

[4] Powell, A. Democratizing Production through Open Source Knowledge: From Open Software to Open Hardware. *Media Cult. Soc.* **2012**, *34*, 691–708.

[5] Oberloier, S.; Pearce, J.M. General Design Procedure for Free and Open-Source Hardware for Scientific Equipment. *Designs* **2018**, *2*, 2. https://doi.org/10.3390/designs2010002.

[6] Pearce, J.M. Sponsored Libre Research Agreements to Create Free and Open Source Software and Hardware. *Inventions* **2018**, *3*, 44. https://doi.org/10.3390/inventions3030044.

[7] Oellermann, M.; Jolles, J.W.; Ortiz, D.; Seabra, R.; Wenzel, T.; Wilson, H.; Tanner, R.L. Open Hardware in Science: The Benefits of Open Electronics. *Integr. Comp. Biol.* **2022**, *62*, 1061–1075. https://doi.org/10.1093/icb/icac043.

[8] Pearce, J.M. Economic Savings for Scientific Free and Open Source Technology: A Review. *HardwareX* **2020**, *8*, e00139. https://doi.org/10.1016/j.ohx.2020.e00139.

[9] Pozo-Ruz, A.; Aguilera, F.D.T.; Moron, M.J.; Rivas, E. Power Electronics Open-Source Educational Platform. *J. Power Electron.* **2012**, *12*, 842–850. https://doi.org/10.6113/JPE.2012.12.5.842.

[10] Balid, W.; Abdulwahed, M.; Alrouh, I. Development of an Educationally Oriented Open-Source Embedded Systems Laboratory Kit: A Hybrid Hands-on and Virtual Experimentation Approach. *Int. J. Electr. Eng. Educ.* **2014**, *51*, 340–353. https://doi.org/10.7227/ijeee.0006.

[11] Oberloier, S.; Pearce, J.M. Open Source Low-Cost Power Monitoring System. *HardwareX* **2018**, *4*, e00044. https://doi.org/10.1016/j.ohx.2018.e00044.

[12] Laverty, D.M.; Best, R.J.; Brogan, P.; Al Khatib, I.; Vanfretti, L.; Morrow, D.J. The OpenPMU Platform for Open-Source Phasor Measurements. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 701–709. https://doi.org/10.1109/TIM.2013.2240920.

[13] Goertzen, L.; Mehr, N.; Lopez, M.; Udell, C.; Selker, J.S. Low-Cost and Precise Inline Pressure Sensor Housing and DAQ for Use in Laboratory Experiments. *HardwareX* **2020**, *8*, e00112. https://doi.org/10.1016/j.ohx.2020.e00112.

[14] Suriano, D. A Portable Air Quality Monitoring Unit and a Modular, Flexible Tool for on-Field Evaluation and Calibration of Low-Cost Gas Sensors. *HardwareX* **2021**, *9*, e00198. https://doi.org/10.1016/j.ohx.2021.e00198.

[15] Merenda, M.; Iero, D.; Pangallo, G.; Falduto, P.; Adinolfi, G.; Merola, A.; Graditi, G.; Della Corte, F.G. Open-Source Hardware Platforms for Smart Converters with Cloud Connectivity. *Electronics* **2019**, *8*, 367. https://doi.org/10.3390/electronics8030367.

[16] Pandey, G.; Vora, A. Open Electronics for Medical Devices: State-of-Art and Unique Advantages. *Electronics* **2019**, *8*, 1256. https://doi.org/10.3390/electronics8111256.

[17] Zimmermann, M.J.Y.; Maia Chagas, A.; Bartel, P.; Pop, S.; Prieto-Godino, L.L.; Baden, T. LED Zappelin': An Open Source LED Controller for Arbitrary Spectrum Visual Stimulation and Optogenetics during 2-Photon Imaging. *HardwareX* **2020**, *8*, e00127. https://doi.org/10.1016/j.ohx.2020.e00127.

[18] Kimchi, E.Y.; Coughlin, B.F.; Shanahan, B.E.; Piantoni, G.; Pezaris, J.; Cash, S.S. OpBox: Open Source Tools for Simultaneous EEG and EMG Acquisition from Multiple Subjects. *eNeuro* **2020**, *7*, . https://doi.org/10.1523/ENEURO.0212-20.2020.

[19] Mesas-Carrascosa, F.J.; Verdú Santano, D.; Meroño, J.E.; Sánchez de la Orden, M.; García-Ferrer, A. Open Source Hardware to Monitor Environmental Parameters in Precision Agriculture. *Biosyst. Eng.* **2015**, *137*, 73–83. https://doi.org/10.1016/j.biosystemseng.2015.07.005.

[20] Botero-Valencia, J.S.; Mejia-Herrera, M.; Pearce, J.M. Low Cost Climate Station for Smart Agriculture Applications with Photovoltaic Energy and Wireless Communication. *HardwareX* **2022**, *11*, e00296. https://doi.org/10.1016/j.ohx.2022.e00296.

[21] Wishkerman, A.; Wishkerman, E. Application Note: A Novel Low-Cost Open-Source LED System for Microalgae Cultivation. *Comput. Electron. Agric.* **2017**, *132*, 56–62. https://doi.org/10.1016/j.compag.2016.11.015.

[22] Lagos-Susaeta, D.; Salazar, O.; Asenjo, J.A. openPFGE: An Open Source and Low Cost Pulsed-Field Gel Electrophoresis Equipment. *HardwareX* **2020**, *8*, e00128. https://doi.org/10.1016/j.ohx.2020.e00128.

[23] Carena, F.; Carena, W.; Chapeland, S.; Chibante Barroso, V.; Costa, F.; Dénes, E.; Divià, R.; Fuchs, U.; Grigore, A.; Kiss, T.; et al. The ALICE Data Acquisition System. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2014**, *741*, 130–162. https://doi.org/10.1016/j.nima.2013.12.015.

[24] Ferdoush, S.; Li, X. Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications. *Procedia Comput. Sci.* **2014**, *34*, 103–110. https://doi.org/10.1016/j.procs.2014.07.059.

[25] Poquita-Du, R.C.; Du, I.P.M.; Todd, P.A. EmerSense: A Low-Cost Multiparameter Logger to Monitor Occurrence and Duration of Emersion Events within Intertidal Zones. *HardwareX* **2023**, *14*, e00410. https://doi.org/10.1016/j.ohx.2023.e00410.

[26] Netto, G.T.; Arigony-Neto, J. Open-Source Automatic Weather Station and Electronic Ablation Station for Measuring the Impacts of Climate Change on Glaciers. *HardwareX* **2019**, *5*, e00053. https://doi.org/10.1016/j.ohx.2019.e00053.

[27] Woo, W.; Richards, W.; Selker, J.; Udell, C. WeatherChimes: An Open IoT Weather Station and Data Sonification System. *HardwareX* **2023**, *13*, e00402. https://doi.org/10.1016/j.ohx.2023.e00402.

[28] Camprodon, G.; González, Ó.; Barberán, V.; Pérez, M.; Smári, V.; de Heras, M.Á.; Bizzotto, A. Smart Citizen Kit and Station: An Open Environmental Monitoring System for Citizen Participation and Scientific Experimentation. *HardwareX* **2019**, *6*, e00070. https://doi.org/10.1016/j.ohx.2019.e00070.

[29] Nguyen, B.; Goto, B.; Selker, J.S.; Udell, C. Hypnos Board: A Low-Cost All-in-One Solution for Environment Sensor Power Management, Data Storage, and Task

Scheduling. *HardwareX* **2021**, *10*, e00213. https://doi.org/10.1016/j.ohx.2021.e00213.

[30] Pringle, A.M.; Oberloier, S.; Petsiuk, A.L.; Sanders, P.G.; Pearce, J.M. Open Source Arc Analyzer: Multi-Sensor Monitoring of Wire Arc Additive Manufacturing. *HardwareX* **2020**, *8*, e00137. https://doi.org/10.1016/j.ohx.2020.e00137.

[31] Herbst, R.; Claus, R.; Freytag, M.; Haller, G.; Huffer, M.; Maldonado, S.; Nishimura, K.; O'Grady, C.; Panetta, J.; Perazzo, A.; et al. Design of the SLAC RCE Platform: A General Purpose ATCA Based Data Acquisition System. In Proceedings of the 2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Seattle, WA, USA, 8–15 November 2014; pp. 1–4.

[32] Ferrero Martín, F.J.; Valledor Llopis, M.; Campo Rodríguez, J.C.; Blanco González, J.R.; Menéndez Blanco, J. Low-Cost Open-Source Multifunction Data Acquisition System for Accurate Measurements. *Measurement* **2014**, *55*, 265–271. https://doi.org/10.1016/j.measurement.2014.05.010.

[33] Niehaus, K.; Westhoff, A. An Open-Source Data Acquisition System for Laboratory and Industrial Scale Applications. *Meas. Sci. Technol.* **2022**, *34*, 027001. https://doi.org/10.1088/1361-6501/ac9994.

[34] CompactDAQ Systems. Available online: https://www.ni.com/en-ca/shop/compactdaq.html (accessed on 15 June 2023).

[35] Maia Chagas, A. Haves and Have Nots Must Find a Better Way: The Case for Open Scientific Hardware. *PLoS Biol.* **2018**, *16*, e3000014. https://doi.org/10.1371/journal.pbio.3000014.

[36] Oberloier, S.; Whisman, N.G.; Hafting, F.; Pearce, J.M. Open Source Framework for a Broadly Expandable and Reconfigurable Data Acquisition and Automation Device (BREAD). *HardwareX* **2023**, *15*, e00467. https://doi.org/10.1016/j.ohx.2023.e00467.

[37] Li, H.; Aguirre-Villegas, H.A.; Allen, R.D.; Bai, X.; Benson, C.H.; Beckham, G.T.; Bradshaw, S.L.; Brown, J.L.; Brown, R.C.; Cecon, V.S.; et al. Expanding Plastics Recycling Technologies: Chemical Aspects, Technology Status and Challenges. *Green Chem.* **2022**, *24*, 8899–9002. https://doi.org/10.1039/D2GC02588D.

[38] Nanda, S.; Berruti, F. Thermochemical Conversion of Plastic Waste to Fuels: A Review. *Environ. Chem. Lett.* **2021**, *19*, 123–148. https://doi.org/10.1007/s10311-020-01094-7.

[39] Papari, S.; Bamdad, H.; Berruti, F. Pyrolytic Conversion of Plastic Waste to Value-Added Products and Fuels: A Review. *Materials* **2021**, *14*, 2586. https://doi.org/10.3390/ma14102586.

[40] Zhao, D.; Wang, X.; Miller, J.B.; Huber, G.W. The Chemistry and Kinetics of Polyethylene Pyrolysis: A Process to Produce Fuels and Chemicals. ChemSusChem 2020, 13, 1764–1774, doi:10.1002/cssc.201903434. Yadav, G.; Singh, A.; Nicholson, S.R.; Beckham, G.T. *Techno-Economic Analysis and Life Cycle Assessment for Pyrolysis of Mixed Waste Plastics*; National Renewable Energy Lab. (NREL): Golden, CO, USA, 2022.

[41] Yadav, G.; Singh, A.; Nicholson, S.R.; Beckham, G.T. *Techno-Economic Analysis and Life Cycle Assessment for Pyrolysis of Mixed Waste Plastics*; National Renewable Energy Lab. (NREL): Golden, CO, USA, 2022.

[42] Kulas, D.G.; Zolghadr, A.; Shonnard, D.R. Liquid-Fed Waste Plastic Pyrolysis Pilot Plant: Effect of Reactor Volume on Product Yields. *J. Anal. Appl. Pyrolysis* **2022**, *166*, 105601. https://doi.org/10.1016/j.jaap.2022.105601.

[43] Quantafuel Expands into the UK to Help Solve Plastic Waste Problem. Available online: https://interplasinsights.com/api/content/36fca3c0-1d0d-11ec-a329-1244d5f7c7c6/ (accessed on 19 October 2023).

[44] How Waste Solutions Company Brightmark Is Converting Plastic Waste into Fuel. Available online: https://brightmark.com/newsroom/business-insider-ashley/ (accessed on 19 October 2023).

[45] 1/8 DIN 4 and 7 Channel Universal Process Ramp & Soak Controllers. Available online: https://www.omega.com/en-us/control-monitoring/controllers/multiloop-controllers/cn1504-series/p/CN1507TC (accessed on 18 September 2023).L

[46] Lewis, A.J.; Campbell, M.; Stavroulakis, P. Performance Evaluation of a Cheap, Open Source, Digital Environmental Monitor Based on the Raspberry Pi. *Measurement* **2016**, *87*, 228–235, doi:10.1016/j.measurement.2016.03.023.

[47] FakhrHoseini, S.M.; Dastanian, M. Predicting pyrolysis products of PE, PP, and PET using NRTL activity coefficient model. *J. Chem.* **2013**, *2013*, 487676.

[48] MTU MOST/MOST_OpenReactor GitLab. Available online: https://gitlab.com/mtu-most/most_openreactor (accessed on 17 November 2023).

[49] Åström, K.J.; Murray, R.M. *Feedback Systems: An Introduction for Scientists and Engineers*; Princeton University Press: Princeton, NJ, USA, 2008; ISBN 978-0-691-13576-2.

[50] *Model Quad Four Independent Controllers|J-KEM*; J-Kem Scientific: St. Louis, MO, USA.

[51] Karami, M.; McMorrow, G.V.; Wang, L. Continuous Monitoring of Indoor Environmental Quality Using an Arduino-Based Data Acquisition System. *J. Build. Eng.* **2018**, *19*, 412–419. https://doi.org/10.1016/j.jobe.2018.05.014.

[52] Li, Z.; Seering, W.; Ramos, J.D.; Yang, M.; Wallace, D.R. Why open source?: Exploring the motivations of using an open model for hardware development. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*; American Society of Mechanical Engineers: New York, NY, USA, 2017; Volume 58110, p. V001T02A059.

[53] Daniel, K.F.; Peter, J.G. Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research. *Mod. Instrum.* **2012**, *2012*, 18950. https://doi.org/10.4236/mi.2012.12002.

[54] Pearce, J.M. *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*; Elsevier: Amsterdam, The Netherlands, 2013; ISBN 978-0-12-410486-0.

[55] Wenzel, T. Open Hardware: From DIY Trend to Global Transformation in Access to Laboratory Equipment. *PLoS Biol.* **2023**, *21*, e3001931. https://doi.org/10.1371/journal.pbio.3001931.

[56] Costa, D.G.; Duran-Faundez, C. Open-Source Electronics Platforms as Enabling Technologies for Smart Cities: Recent Developments and Perspectives. *Electronics* **2018**, *7*, 404. https://doi.org/10.3390/electronics7120404.

[57] Rocha Ribeiro, R.; Bauer, E.; Lameiras, R. HIGROTERM: An Open-Source and Low-Cost Temperature and Humidity Monitoring System for Laboratory Applications. *Inventions* **2021**, *6*, 84. https://doi.org/10.3390/inventions6040084.
[58] Toldrà, A.; Ainla, A.; Khaliliazar, S.; Landin, R.; Chondrogiannis, G.; Hanze, M.; Réu, P.; Hamedi, M.M. Portable Electroanalytical Nucleic Acid Amplification Tests Using Printed Circuit Boards and Open-Source Electronics. *Analyst* **2022**, *147*, 4249–4256. https://doi.org/10.1039/D2AN00923D.
[59] Grinias, J.P.; Whitfield, J.T.; Guetschow, E.D.; Kennedy, R.T. An Inexpensive, Open-Source USB Arduino Data Acquisition Device for Chemical Instrumentation. *J. Chem. Educ.* **2016**, *93*, 1316–1319. https://doi.org/10.1021/acs.jchemed.6b00262.
[60] Davis, J.J.; Foster, S.W.; Grinias, J.P. Low-Cost and Open-Source Strategies for Chemical Separations. *J. Chromatogr. A* **2021**, *1638*, 461820. https://doi.org/10.1016/j.chroma.2020.461820.
[61] Urban, P.L. Universal Electronics for Miniature and Automated Chemical Assays. *Analyst* **2015**, *140*, 963–975. https://doi.org/10.1039/C4AN02013H.
[62] Urban, P.L. Prototyping Instruments for the Chemical Laboratory Using Inexpensive Electronic Modules. *Angew. Chem. Int. Ed.* **2018**, *57*, 11074–11077. https://doi.org/10.1002/anie.201803878.
[63] Urban, P.L. Open-Source Electronics As a Technological Aid in Chemical Education. *J. Chem. Educ.* **2014**, *91*, 751–752. https://doi.org/10.1021/ed4009073.
[64] Ngo, T.D. Open-Source Electronics Platforms: Development and Applications. *Electronics* **2019**, *8*, 428. https://doi.org/10.3390/electronics8040428.
[65] Thomas, L. Business Models for Open Source Hardware. Ph.D. Thesis, Université Grenoble Alpes, Grenoble, France, 2019.
[66] Viseur, R. From Open Source Software to Open Source Hardware. In *Open Source Systems: Long-Term Sustainability*; Hammouda, I., Lundell, B., Mikkonen, T., Scacchi, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 286–291.
[67] Getting Started with Arduino Products. Available online: https://www.arduino.cc/en/Guide (accessed on 22 November 2023).

Chapter 3

3 Parametric Low-Cost Open-Source Bioreactor Designed for Distributed Manufacturing

Bioreactors are critical for biotechnology, but costs of proprietary systems limit accessibility and open-source bioreactors are generally focused on a specific application. A customizable design is provided in this study, allowing scientists from many disciplines to develop bespoke open-source bioreactors for use in a variety of applications, including in lab and scientific settings as well as for appropriate technology distributed manufacturing applications. The design is built, each control sub-system (heating, oxygen and pH control) was tested with water and a lactic acid solution to understand the bioreactor’s precision and the entire design was validated with yeast production. The parametric 3D printed components are designed to be easily modified to adapt to specific parts (e.g. to adapt to different types of sensors, pumps, motors or containers). Thus, it is also easier to recreate and maintain because the necessary components are readily available and the mechanical, electrical and software components are all customizable. The bioreactor can be fabricated with less than $1000 in components -- a ~ 90% savings from commercial systems. The bioreactor can be used for a wide range of applications including medical applications, cultivation of bacteria, algae, and other similar species for use in laboratories or as food.

3.1 Hardware in Context

Bioreactors are a critical technology for successful design and operation of biotechnical processes [1]. Bioreactors are continuous culture devices that simultaneously pass media into and remove volume from active cultures to maintain volume homeostasis. The rate of media inflow and outflow is called the dilution rate; various types of bioreactors dynamically adjust the dilution rate to keep a desired parameter constant. Throughout the process, various parameters are measured and used in feedback to control the system such as the pH, the dissolved oxygen, and the temperature. Bioreactors have a vast array of applications including degradation of pollutants or to produce beneficial substances such as human food, animal feed, chemicals and pharmaceuticals, and tissues or even organs for use in biomedicine [2]. Commercial bioreactors are expensive as seen in Table 3.1, which limits accessibility in low-resource and underfunded laboratories [3].

Table 3.1: Commercial Bioreactor Comparison

Commercial Bioreactors	Temp. Range ° C	Dissolved Oxygen Range	Stirring Type and Speed	Vessel Material	PH Range	Vessel Size	Starting Cost (USD)	Source
Biostat B	0-60	0-100% saturation	Direct Drive: 1L, 2L: 20-2000 5L: 20-1500 10L: 20-800	glass	2-12	1,2,5,10L	$8,499.99	[4]
Bioflo 120	0-70	0-100% saturation	Continuous Stirred	glass	2-12	1,2,5,10L	$14,999.00	[5]
Thermo Scientific HyPerforma Rocker Bioreactor	±0.15°C at 15°C to 40°C (±0.25°F at 59°F to 104°F)	0-100% saturation	Rocking Stirring	Stainles s Steel	2-12	5 L to 25 L	$6,500 .00	[6]
DynaDrive Single-Use Bioreactor	±0.15°C at 15°C to 40°C (±0.25°F at 59°F to 104°F)	0-100% saturation	Rocking Stirring	Stainles s Steel	2-12	50L, 500L	$4,495.00	[7]
The Mobius®	0-40	0-100% saturation	Continuous Stirred	Glass	2-12	3L	$1,511.29	[8]
Allegro STR 200	4 ºC to 37 ºC	0-100% saturation	Continuous Stirred	Stainles s Steel	2-12	20L	$5,500.00	[9]
CSTR-5G	0-40	0-100% saturation	Continuous Stirred	Glass	2-12	5L	$5,393.26	[10]
Labfors	Up to 95	0-100% saturation	Magnetic Drive: 20-300 rpm	Glass	2-12	2, 3.6, 7.5, 13L	$14,250.00	[11]

The demand for these open-source bioreactors is directly impacted by several constraints brought on by current open source and commercial models. The main drawback is the expense; as bioreactors require a large initial outlay and recurring expenses, they are not feasible for many smaller-scale laboratories. This high cost is due to multiple reasons. Materials unnecessary for most laboratory tasks, such as stainless steel, are often used instead of more economic materials like glass or plastics. Additionally, the capabilities, precision, and quality of sensors, pumps, motors, and heating in commercial bioreactors is often unnecessary, however no lower-cost alternatives are offered. Over-engineering of vessels, stands, connections, sealing mechanisms, and other subcomponents continue to contribute to cost. Lastly, with premium components and proprietary hardware, repairability and reusability also become expensive, leading to long term costs. High production costs are ultimately passed on to the consumer in the form of a higher price. As a result of this, accessibility to resource-constrained laboratories without the need for a premium bioreactor becomes difficult without a lower cost substitute. The open hardware model of technical development [12,13] has been widely shown to reduce the cost of scientific hardware [14-19]. This approach has been applied to bioreactor technology as summarized in Table 3.2. As can be seen from Table 3.2, for those systems which have available capital costs the ranges are lower than in Table 3.1.

Table 3.2: Open-Source Bioreactors

Reactor (set up, name)	Temp (°C)	Dissolved Oxygen Range	Type of Stirring	Vessel Material	PH Control	Size (Pilot Scale volume)	Cost (USD$)	Source
Nitrogen removal from wastewater	27	6.0–7.0 mg/L	constant	Glass	7.5 to 8.3	N/A	NA	[20]
Woodchip Bioreactor	17	N/A	constant	Metal	7	2.21m	NA	[21]
Typical Lab Scale Bioreactor	20-40	N/A	constant	Stainless Steel	7	0.3-10	NA	[22]
Food Production Bioreactor	30	2.6 mg/L	constant	Perspex	7.5	1.2L	NA	[23]

Pilot Scale Nutrient Removal Bioreactor	23	10.4 mg/L	constant	Polypropylene	7.4	4m area	NA	[24]
Stirred Tank Reactor	35	N/A	constant	Glass	7	5-7m area	NA	[25]
Pharmaceutical Reactor	37	3.48 mg/l	constant	Hollow fiber	7	1m surface area	NA	[26]
MFC	30	6.0 mg/L	constant	Stainless Steel	7	1.0 L	NA	[26]
Municipal Wastewater	27	N/A	constant	Hollow Fibre	7	N/A	NA	[26]
Petroleum	37	N/A	constant	Silver	7	N/A	NA	[27]
Wastewater	35.7	4.6	constant	Stainless Steel	7	N/A	NA	[26]
Pioreactor	Ambient +20	N/A	Magnetic: 100-1000RPM	Glass	N/A	20ml	$260.29	[28]
EVE	Ambient +62	N/A	Magnetic, 5000rpm	Glass	N/A	20ml	$819.00	[29]
Igem 2015 AACHEN	Variable	N/A	Magnetic 100-1400RPM	Glass	N/A	20ml	$123.79	[30]
Hackuarium Bioreactor	Variable	N/A	Direct Drive 600 RPM	Glass or Polycarbonate	1-13; 0.1	1L	$1,790	[31]

Scalability is yet another issue in earlier efforts. The sizing of this type of reactor is extremely important to the outcome, as certain models examined in earlier studies are challenging to scale up or scale down. Another crucial element is contamination; many of the studied bioreactors are single use, making it unwise to autoclave or recycle them. It would be extremely undesirable to work with a one-use bioreactor due to the high expense of bioreactor production. These limitations were all taken into consideration when designing the open-source bioreactor described here.

To overcome these limitations, this study describes a parametric, open-source continuous bioreactor system that can be quickly built from readily accessible components, adapted and customized for different types of studies, and used in academic and pedagogical settings. The primary goal of the design is to enable all the functions of the most common scientific bioreactors while radically reducing the costs from the proprietary systems shown in Table 3.1. A recent review showed that, for a wide range of scientific tools, open-source technologies provide economic savings of 87% compared to equivalent or lesser proprietary tools [32]. These economic savings increased to 94% when the designs used Arduino open-source microcontroller technology [33] and RepRap-class 3D printing [34-36]. To capitalize on this trend the open-source bioreactor here uses Arduino controllers inside of a BREAD electronics prototyping system for monitoring and control [37]. In addition, as open source 3D printing is now widely used in the scientific enterprise to make custom components at low cost [38,39], it is used here to ensure that the bioreactor is accessible and as versatile as possible.

A customizable design is provided in this study, allowing scientists from a wide range of disciplines to develop bespoke open-source bioreactors for use in a variety of applications, including in lab and scientific settings as well as for appropriate technology distributed manufacturing applications. The design is built, each control sub-system (heating, oxygen and pH control) was tested with water and a lactic acid solution to understand the bioreactor's precision and the entire design was validated with yeast production.

3.2 Hardware Description

To overcome the high costs of proprietary bioreactors and lack of flexibility of existing open-source bioreactors to allow anyone to fabricate a scalable customizable bioreactor, this study develops a distributed manufacturing solution using only an open-source manufacturing tool chain and provides a parametric fully free design of the components needed for control of temperature, pH and oxygen. Customizable components that can be 3D printed on readily accessible fused filament fabrication-based 3D printers were developed in parametric FreeCAD [40].

The open-source bioreactor (Figure 3.1) consists of a housing made up of a 1.5-liter food canister with a stainless-steel cover., readily available mechanical components, electronics that can be fabricated with an open-source PCB mill, and custom 3D printable components. The open-source bioreactor is designed to be as modular and customizable as possible. It uses the BREAD electronics prototyping platform and the software interface of the Open Reactor so other functionalities for monitoring or control can be readily added.

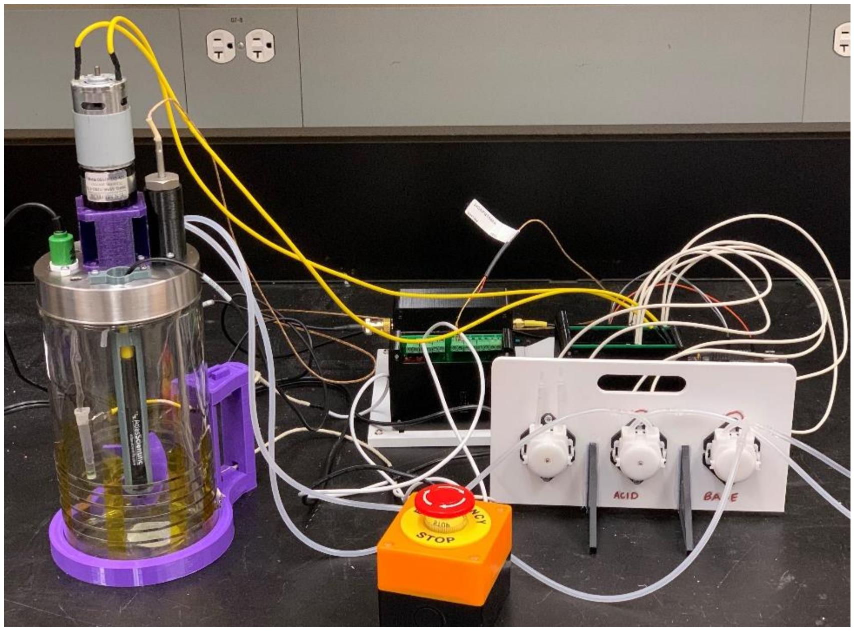


Figure 3.1: Fully assembled open-source bioreactor prototype

3D printed components are designed to be easily modified to adapt to specific parts used in a particular build (e.g. to adapt to a different type of sensor, or container). As a result, the device is also easier to recreate because the necessary components are readily available and the mechanical, electrical and software components are all readily customizable.

This approach has several advantages over purchasing a proprietary bioreactor:

- <USD$925 device replaces commercial bioreactors saving 90%
- Parametric open-source electronics designs allow for adaptation to other core components (e.g. existing pumps, motors, sensors)
- Controllable temperature, pH, O_2.

- Parametric open source 3D printable files enable customizing size, shape and material compatibility.
- Flexibility to be used for a wide range of applications including medical applications (e.g. for pharmaceuticals), cultivation of bacteria, algae, and other similar species for use in laboratories, additionally, the bioreactor can aid in the production of food goods like yeast.

3.3 Design Files Summary

Design file name	**File type**	**Open source license**	**Location of the file**
Impeller.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
MotorMountPrint.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
O2MountPrint.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
PHMountPrint.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
ThermocoupleMountPrint.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
LidMachiningStencil.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
Pump Stand Side.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
Pump Stand Top.stl	STL	CERN OHL v2	https://osf.io/yxfj6/
Bioreactor Pump Stand	Onshape	CERN OHL v2	https://cad.onshape.com/documents/0f3f9a273edf747f2ffb205e/w/46d2e4aaae25fe9bdb749342/e/3cafe07e46a83fcf52532aaf?renderMode=0&uiState=63f11de5e716fe7dc03b4d34

BOM_Bioreactor.ods	ODS	CERN OHL v2	https://osf.io/yxfj6/
Documentation/BOM_Slice.ods	ODS	CERN OHL v2	https://osf.io/u2h4g/
Firmware/firmware.ino	.ino Arduino Code	GNU GPL v3	https://osf.io/u2h4g/
Gerbers	Gerber board files	CERN OHL v2	https://osf.io/u2h4g/
Hardware/Bread_Slice.pro	KiCAD Project	CERN OHL v2	https://osf.io/u2h4g/
Hardware/Bread_Slice.sch	KiCAD Schematic	CERN OHL v2	https://osf.io/u2h4g/
Hardware/Bread_Slice.kicad_pcb	KiCAD PCB	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/Top.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/Bottom.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/ LOAF - Top 1.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/ LOAF - Top 2.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/ LOAF - Bottom 1.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
Mechanical/ LOAF - Bottom 1.stl	STL	CERN OHL v2	https://osf.io/u2h4g/
OpenReactor	Repository	GNU GPL v3	https://gitlab.com/mtu-most/most_openreactor

For all 3D printable components the CAD files are provided to allow for easy adjustments and the STLs used for the example bioreactor system are provided for immediate 3D printing.

- Impeller.stl: 3D model of the impeller used for 3D printing
- MotorMountPrint.stl: 3D model of the mount for the motor. The STL file for printing
- O2MountPrint.stl: 3D model of the mount for the O2 sensor, used for 3D printing the STL file
- PHMountPrint.stl: 3D model of the mount for the PH sensor, used for 3D printing the STL file
- ThermocoupleMountPrint.stl: 3D model of the mount for the PH sensor, used for 3D printing
- LidMachiningStencil.stl: 3D printable stencil of the layout for the holes in the lid of the jar
- BOM_Bioreactor.ods: Bill of materials spreadsheet for bioreactor
- Documentation/BOM_Slice.ods: Bill of materials spreadsheet for each Slice
- Firmware/firmware.ino: the default firmware to be programmed onto the Slice
- Gerbers: a folder which contains all Gerber files necessary for construction of the circuit board
- Hardware/Bread_Slice.pro: The main project file for the electrical design
- Hardware/Bread_Slice.sch: The schematic of the electrical design
- Hardware/Bread_Slice.kicad_pcb: The board routing for the slice
- Mechanical/Top.stl: 3D model of top Slice casing
- Mechanical/Bottom.stl: 3D model of bottom Slice casing
- Mechanical/ LOAF - Top 1.stl: 3D model of top Loaf casing part 1
- Mechanical/ LOAF - Top 2.stl: 3D model of top Loaf casing part 2
- Mechanical/ LOAF - Bottom 1.stl: 3D model of bottom Loaf casing part 1
- Mechanical/ LOAF - Bottom 2.stl: 3D model of bottom Loaf casing part 2

- OpenReactor: an online repository of all the files needed to run the OpenReactor software

Controlling and monitoring the bioreactor was accomplished with the RLHT, DCMT, and PHDO Slices from the BREAD system [37]. Each Slice has a standard set of hardware for communicating with other Slices and storing unique firmware on an Arduino Nano (Figure 3.2).

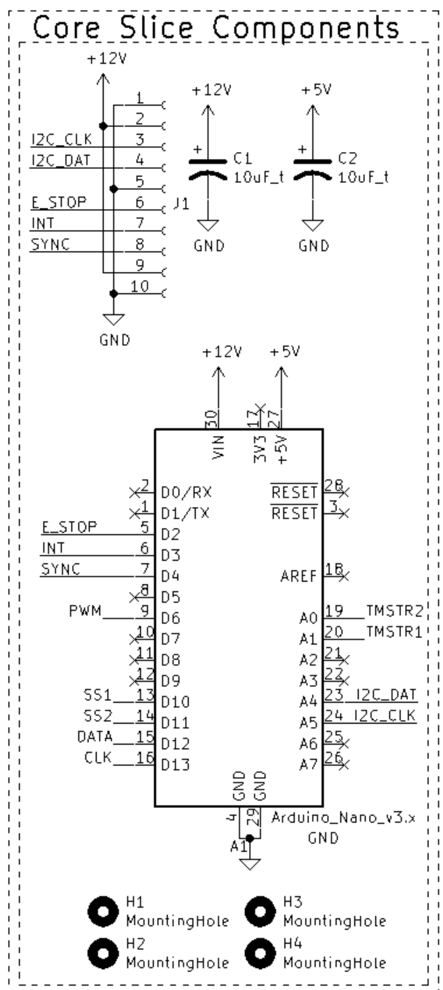


Figure 3.2: Core Slice Components

3.3.1 SLC_RLHT Relay Heater Slice

SLC_RLHT consists of a single relay control output (Figure 3.3) with two k-type thermocouple inputs (Figure 3.4) and two thermistor inputs (Figure 3.5). Each thermistor input has overvoltage protection using a Zener diode. The relay output can be used to control many high-power devices such as heaters, solenoids, pumps, or anything that requires ON/OFF or slow PWM control. This Slice is best used in heating applications where precise temperature monitoring is needed in multiple locations or specific temperature probes are required. Both the thermocouple and thermistor inputs provide flexibility on the type of sensor used. The circuit board layout and assembled Slice are shown in Figure 3.6. Note that the assembled board is an old version of the Slice. The updated version in the design files uses an optoisolator for isolation protection instead of a MOSFET.

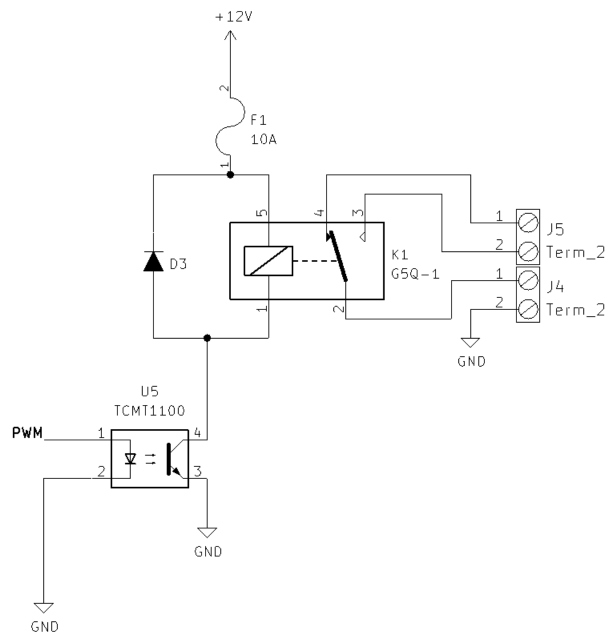


Figure 3.3: SLC_RLHT Relay Output

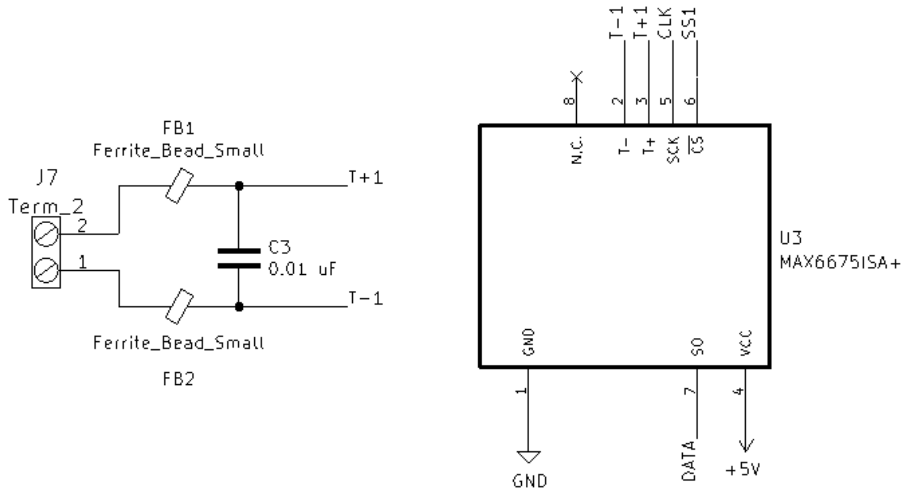


Figure 3.4: SLC_RLHT Thermocouple Input

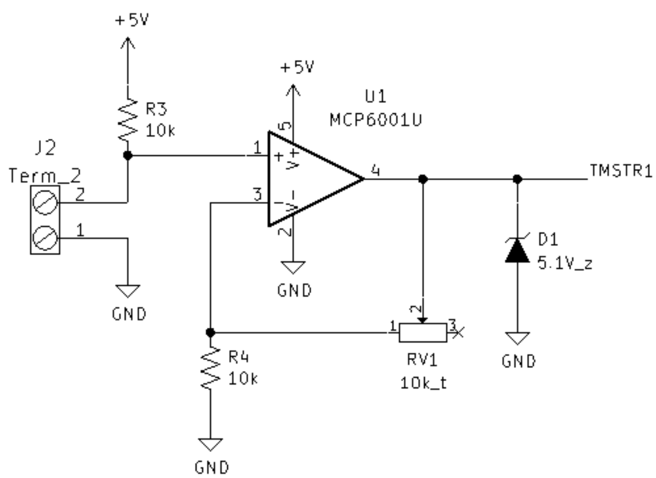


Figure 3.5: SLC_RLHT Thermistor Input

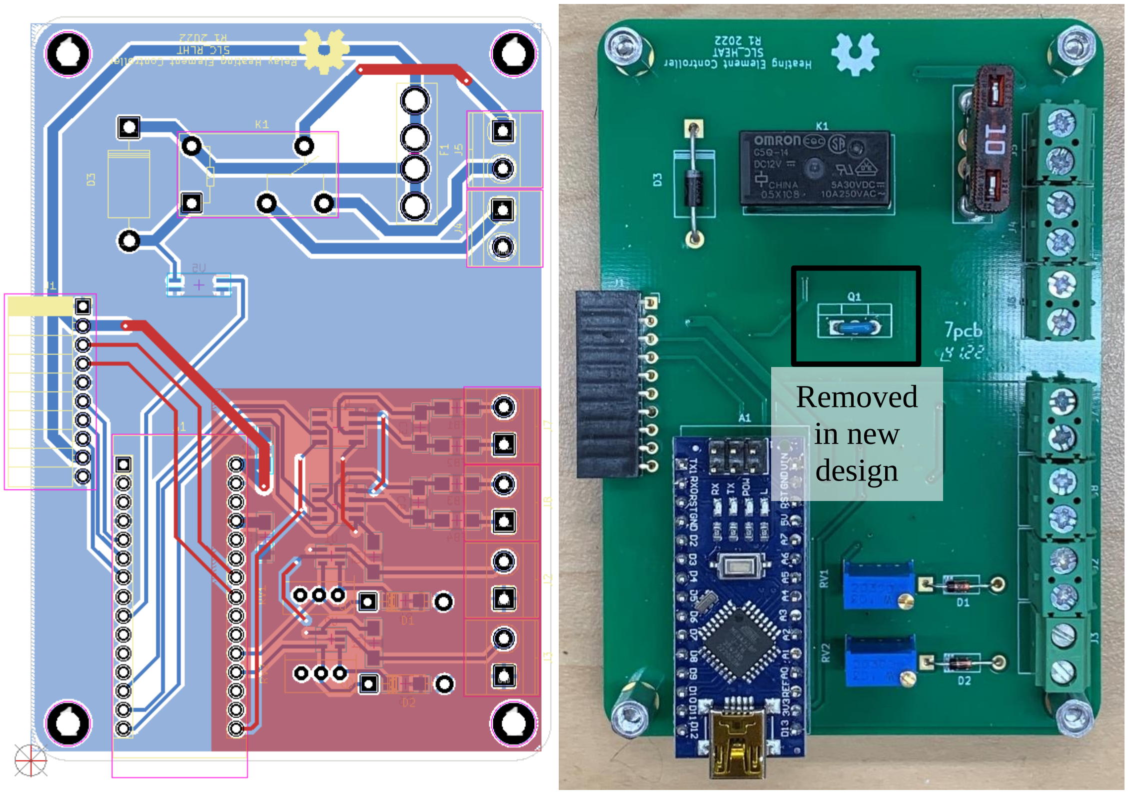


Figure 3.6: SLC_RLHT Circuit Board (left) Assembled (right)

3.3.2 SLC_DCMT DC Motor Slice

SLC_DCMT consists of two DC motor channels (Figure 3.7) with two optional encoder and external power inputs (Figure 3.8). Each motor has the direction, speed, and braking controlled with an LM18200 which can also output the current draw and internal temperature of the chip. Each motor can be powered from the Slice connector's 12V input or an external power supply via a jumper. The circuit board layout and assembled Slice are shown in Figure 3.9.

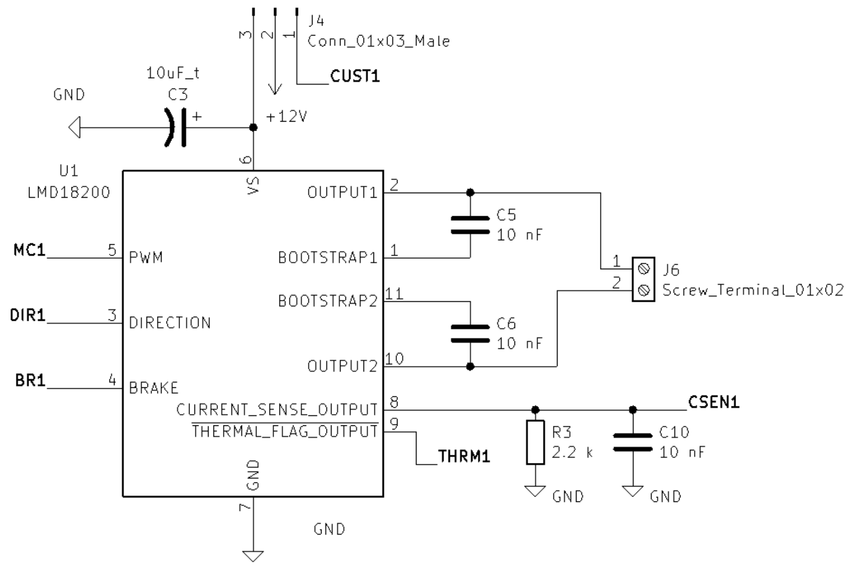
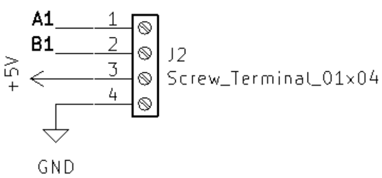


Figure 3.7: SLC_DCMT Motor Channel



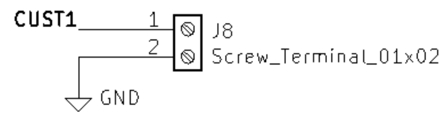


Figure 3.8: SLC_DCMT Encoder Input (left) External Power Input (right)

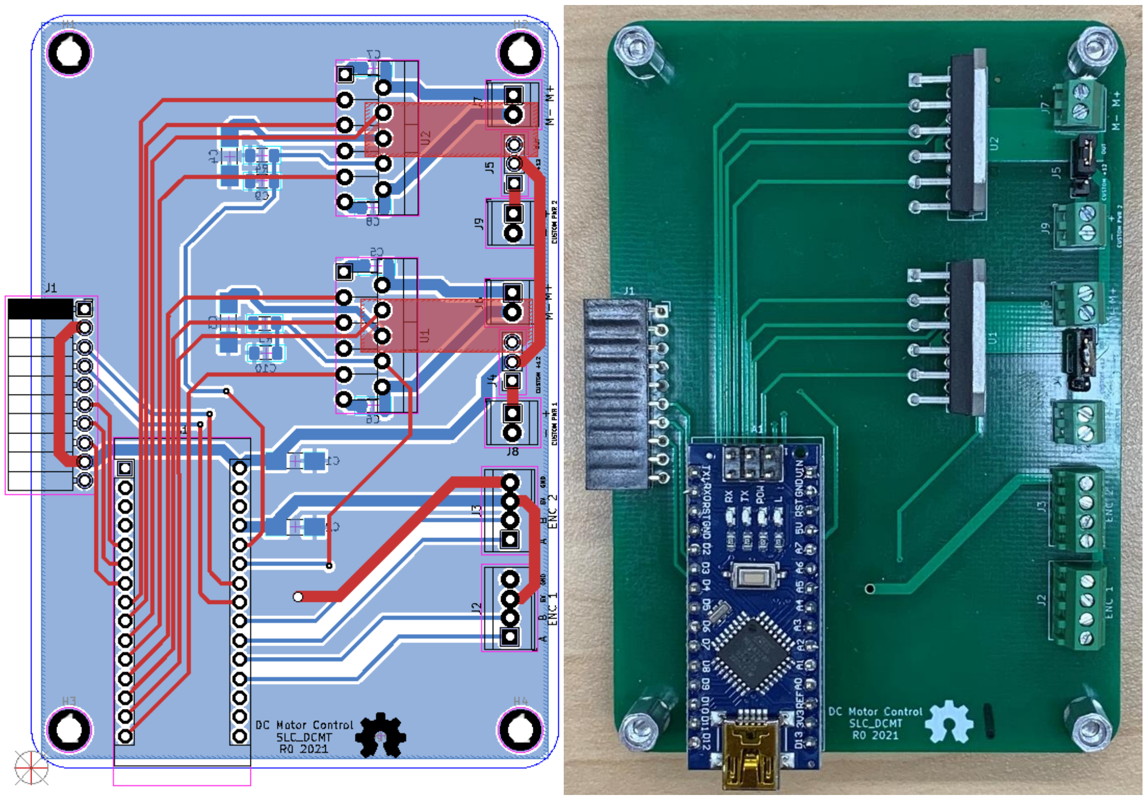


Figure 3.9: SLC_DCMT Circuit Board

3.3.3 SLC_PHDO pH and Dissolved Oxygen Slice

SLC_PHDO acts as a platform for integrating Atlas Scientific EZO-pH and EZO-DO integrated sensors into the BREAD ecosystem. Both sensors can communicate over I2C or UART; however, Slices communicate through I2C. The Arduino Nano acts as both a voltage regulator, providing the sensors with 5V power, and an I2C to UART bridge if UART is chosen as the preferred method of communication with the Atlas Scientific sensors. Otherwise, the sensors can communicate directly over BREAD's I2C bus at addresses 99 and 97 for the pH and DO sensors, respectively. The circuit board layout for the slice is shown in Figure 3.10. At the time of testing, the PCB for this Slice was unavailable due to shipping delays. Instead, the sensors were connected using a protoboard and 3D printed backplate as a temporary solution for testing purposes (Figure 3.11).

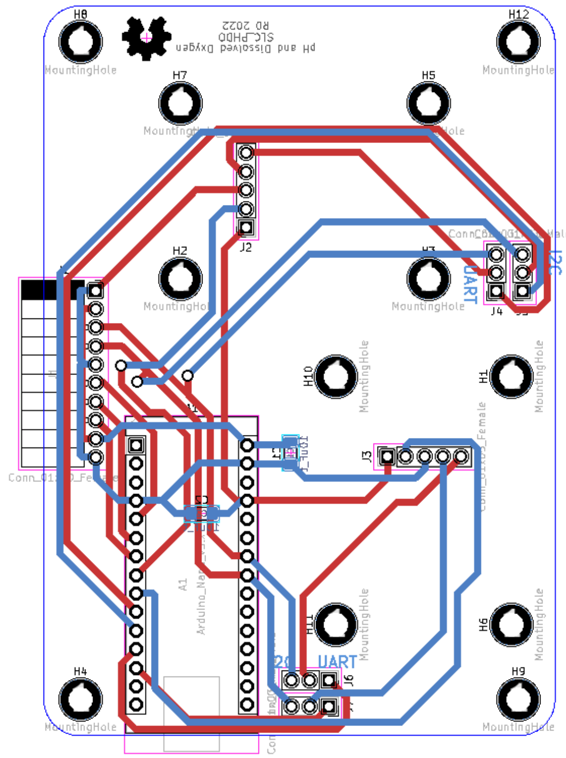


Figure 3.10: SLC_PHDO Circuit Board

Figure 3.11: Temporary SLC_PHDO Slice for Testing

3.3.4 Bioreactor Hardware

Each Slice mentioned above was performance tested with the open-source bioreactor. The hardware consisted of:

- 2x SLC_DCMT
- 1x SLC_RLHT
- 1x SLC_PHDO

These Slices were linked together with a Loaf backplate as shown in Figure 3.12. An external BeagleBone Black was used to run OpenReactor software to track experiments and translate control commands from the user to BREAD. However, many other embedded Linux boards will work for this application as well (i.e., Raspberry Pi).

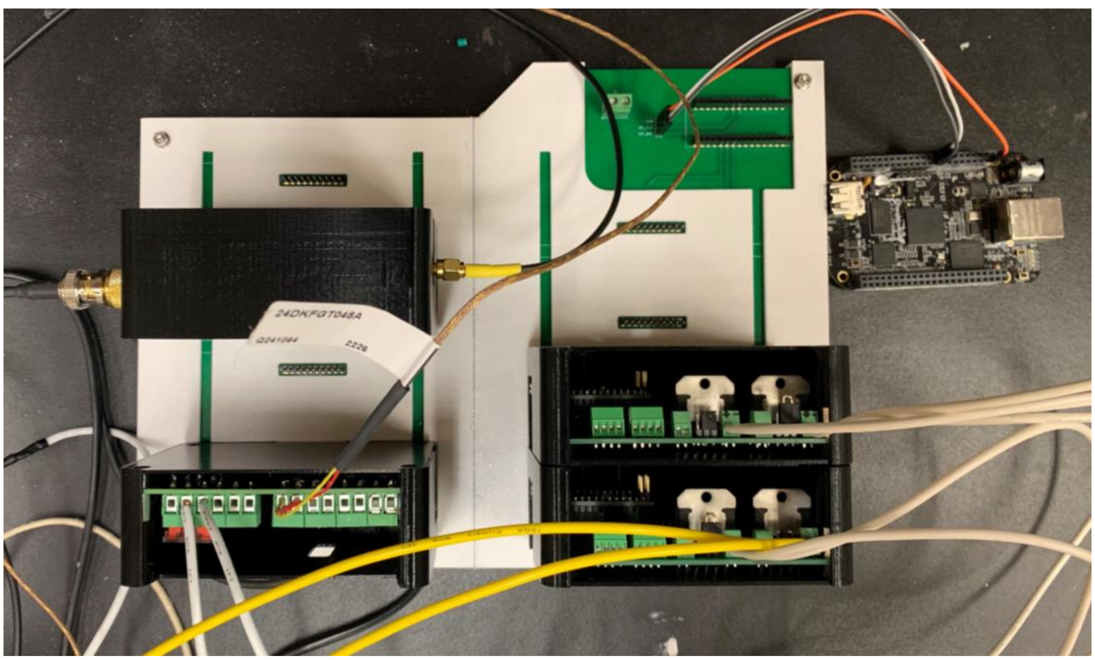


Figure 3.12: Bioreactor Hardware

3.3.5 Bioreactor Software

The OpenReactor software was used to control the bioreactor. OpenReactor is an open-source data acquisition and control system designed for use with I2C enabled devices. It can be run on any system compatible with the Adafruit busIO module such as Raspberry

Pis or Beaglebones and has a remotely accessible web interface for controlling the system. Any I2C sensor can be used for data acquisition with the ability to define custom read protocols, and by using BREAD Slices non-I2C sensors can also be implemented.

OpenReactor allows for the creation of custom control scripts in the form of Python classes, which allow for the control of I2C devices. These scripts can use both user input and sensor data and have the option of being automated via the experiment cycle system.

The experiment cycle system automatically takes measurements and triggers enabled control systems. Measured data is stored and can be exported from the web interface to any connected device for further processing and analysis.

Both sensors and control devices can easily be added to the system by editing a JSON file and creating any desired control scripts as described within the documentation. Once added, devices are automatically included within the interface and are completely usable within the system.

By following the directions on the OpenReactor repository and modifying the “devices.json” file, each Slice was incorporated into the software. The customized software interface is shown in Figure 3.13.

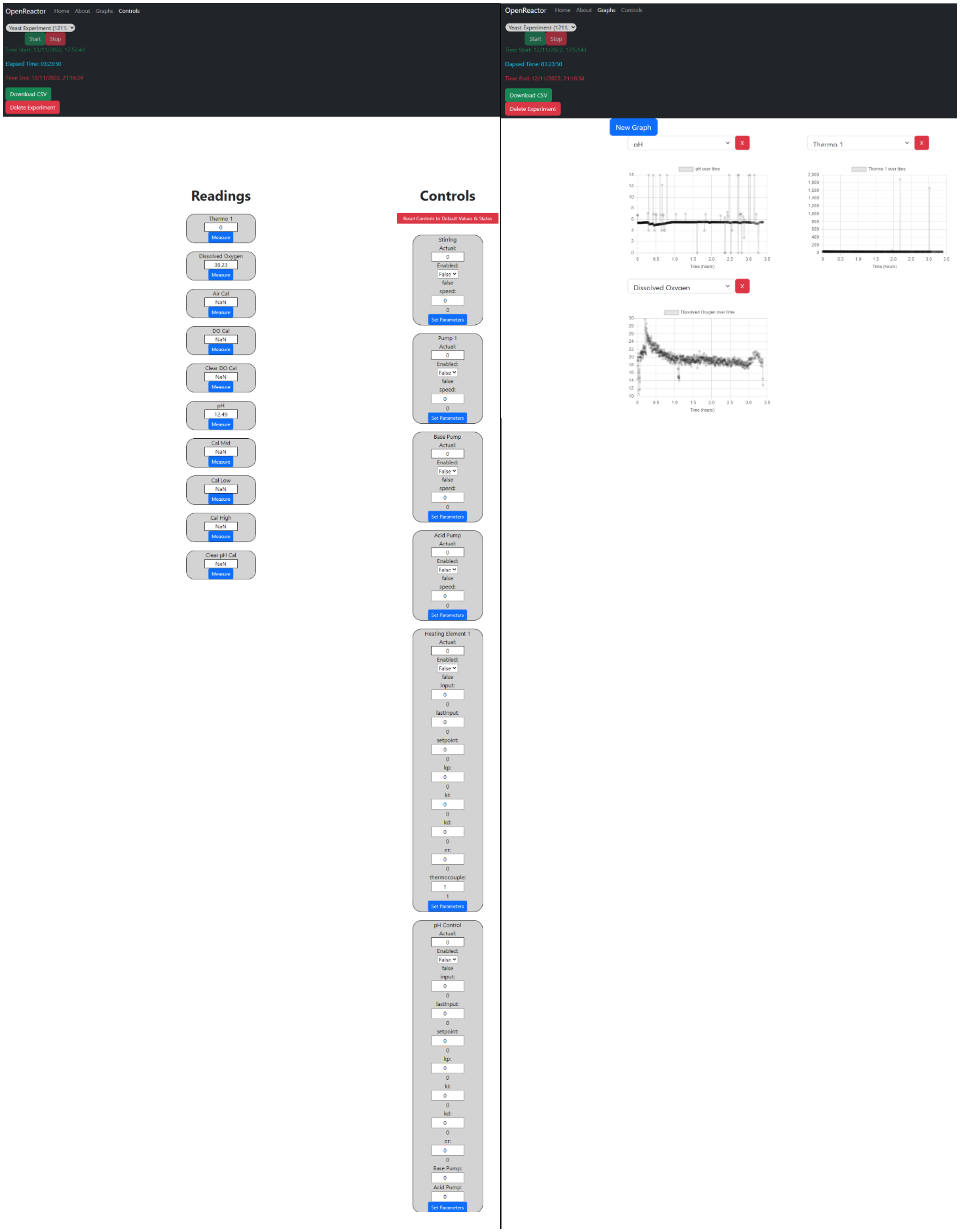


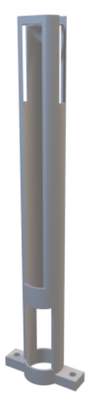
Figure 3.13: OpenReactor Control (left) Graphs (right) Interface

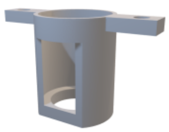
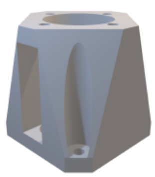
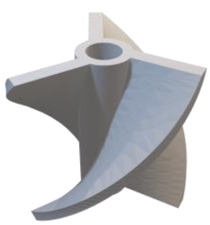
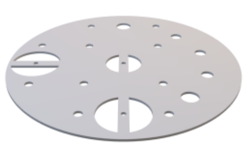
3.4 Bill of Materials Summary

For the URLs of the components listed in the BOM refer to the BOM files in the design files summary. The open-source bioreactor mechanical hardware is shown in Table 3.3, the Loaf Backplane BOM is shown in Table 3.4, SLC_RLHT Slice BOM is in Table 3.5, the SLC_DCMT Slice BOM is in Table 3.6, the SLC_PHDO Slice BOM is in Table 3.7, and the Slice and Loaf casing BOM is shown in Table 3.8.

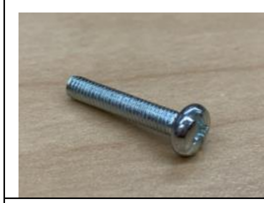
Table 3.3: Bioreactor Mechanical Hardware

Component Image	**Component**	**Number**	**Cost per unit - currency**	**Total cost - currency**	**Source of materials**	**Material type**
	1.5L Glass Jar	1	3.50 CAD	3.50 CAD	General Store	Glass, Other
	O_2 Sensor	1	310.99 USD	310.99 USD	Atlas Scientific	Other
	pH Sensor	1	250 USD	250 USD	Hannah Instruments	Other
	Thermocouple	1	88.71 CAD	88.71 CAD	McMaster-CARR	Metal, Polymer
	36mm Diameter High Torque	1	34.50 CAD	34.50 CAD	RobotShop	Other

	Planetary Gear Motor - 12V, 1150RPM DC Motor					
	8mm-8mm Aluminum Shaft Coupler	1	1.65 CAD	1.65 CAD	Amazon	Metal
	8mm Linear Steel Rod	1	12.99 CAD	12.99 CAD	Amazon	Metal
	8mm ID Shielded Bearing	1	2.57 CAD	2.57 CAD	Digikey	Metal
	3D printed PLA O2 Sensor Mount (8.19g)	1	0.25 CAD*	0.25 CAD	N/A	Polymer

	3D printed PLA PH Sensor Mount (2.44g)	1	0.07 CAD*	0.07 CAD	N/A	Polymer
	3D printed PLA Thermocouple Mount (9.88g)	1	0.30 CAD*	0.30 CAD	N/A	Polymer
	3D printed PLA Motor Mount (20.26g)	1	0.61 CAD*	0.61 CAD	N/A	Polymer
	3D printed PLA Impeller (11.48g)	1	0.34 CAD*	0.34 CAD	N/A	Polymer
	3D printed Lid Machining Stencil (9.17g)	1	0.28 CAD*	0.28 CAD	N/A	Polymer
	3D printed pump stand top (88.36g)	1	2.65 CAD*	2.65 CAD	N/A	Polymer
	3D printed pump stand side (15.21g)	2	0.46 CAD*	0.91 CAD	N/A	Polymer

	Stone Sparger	1	0.83 CAD	0.83 CAD	Amazon	Other
	Peristaltic Pump	3	12.00 CAD	36.00 CAD	Amazon	Other
	Air Pump Kit	1	38.71 CAD	38.71 CAD	Amazon	Polymer, Other
	2 meter, 2mm ID Silicone Tubing	2	9.99CAD	19.98 CAD	Amazon	Polymer
	1 meter, 6mm ID Silicone Tubing	1	10.99 CAD	10.99 CAD	Amazon	Polymer
	8-Feet, 4mm ID Silicone Tubing	1	2.97 CAD	2.97 CAD	Amazon	Polymer

	PTFE Air Filter	1	15.32 CAD	15.32 CAD	Cole Parmer	Polymer
	Straight Tube Reducers, 6mm-4mm	2	1.08 CAD	2.16 CAD	McMaster-CARR	Polymer
	250ml Glass Bottles	3	3.50 CAD	10.50 CAD	Amazon	Glass
	M3x15 Screws	20	N/A	10.98 CAD	Amazon	Metal
	M3 Nuts	20	N/A	9.28 CAD	Hardware Store	Metal
	Ni-Chrome Wire	2.5m	11.22 CAD	28.05 CAD	Omega	Metal
			Total	**$1,120.58 CAD**		

* Assuming cost of a 1kg spool of PETG as ~$30 (cost per gram = $0.03)

Table 3.4: Loaf Backplane BOM

Designator	**Component**	**Number**	**Cost per unit – CAD**	**Total cost – CAD**	**Source of materials**	**Material type**
C1, C2	CAP_10uF_1206_3216Metric_Pad1.33x1.80mm_HandSolder	2	$0.81	$1.62	Digikey	Semi-conductor
U1	micro_SD_connector	1	$1.74	$1.74	Digikey	Semi-conductor
A1	Arduino_Nano	1	$9.58	$9.58	Amazon	Semi-conductor
J1, J2, J3, J4, J5, J6, J7, J8	PinHeader_1x10_P2.54mm_Vertical	8	$0.25	$2.00	Digikey	Polymer
J11, J12	PinHeader_1x06_P2.54mm_Vertical	2	$0.72	$1.44	Digikey	Polymer
J9	TerminalBlock_Phoenix_MKDS-1,5-2_1x02_P5.00mm_Horizontal	1	$0.68	$0.68	Digikey	Polymer
J10	PinHeader_1x03_P2.54mm_Vertical	1	$0.14	$0.14	Digikey	Polymer
			Total	**$17.20**		

Table 3.5: SLC_RLHT BOM

Designator	**Component**	**Number**	**Cost per unit – CAD**	**Total cost – CAD**	**Source of materials**	**Material type**
A1	Arduino_Nano	1	$11.40	$11.40	Amazon	Semi-conductor
C1, C2	CAP_10uF_1206_HandSoldering	2	$0.56	$1.12	Digikey	Semi-conductor

C3, C6	CAP_0.01uF_1206_HandSoldering	2	$0.19	$0.38	Digikey	Semi-conductor
D1, D2	Diode_Zener_DO-41_SOD81_Horizontal_RM10	2	$0.05	$0.10	Digikey	Semi-conductor
D3	Diode_DO-201AD_Horizontal_RM15	1	$0.10	$0.10	Digikey	Semi-conductor
F1	Fuse_Holder_20A_ATO, ATC, 257 Series	2	$0.31	$0.62	Digikey	Semi-conductor
FB1, FB2, FB3, FB4	Ferrite_Bead_Small	4	$0.26	$1.04	Digikey	Semi-conductor
J1	Conn_01x10_Female_PinSocket_1x10_P2.54mm_Horizontal	1	$0.79	$0.79	Digikey	Semi-conductor
J2, J3, J4, J5, J7, J8	TerminalBlock_Pheonix_MKDS1.5-2pol	6	$0.54	$3.24	Digikey	Semi-conductor
K1	Relay_SPDT_Omron-G5Q-1	1	$1.66	$1.66	Digikey	Semi-conductor
R1, R3, R4, R6	R_10k_1206_HandSoldering	4	$0.10	$0.40	Digikey	Semi-conductor
RV1, RV2	Potentiometer_10k_Bourns_3296W_3-8Zoll_Inline_ScrewUp	2	$1.83	$3.66	Digikey	Semi-conductor
U1, U2	MCP6001U_SOT-23-5	2	$0.32	$0.64	Digikey	Semi-conductor
U3, U4	MAX6675ISA+	2	$19.98	$39.96	Digikey	Semi-conductor
	FUSE AUTOMOTIVE 10A 32VDC BLADE BK/ATC-10	1	$1.79	$1.79	Digikey	Semi-conductor

			Total	**$66.90**		

Table 3.6: SLC_DCMT BOM

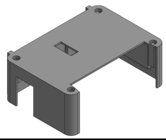
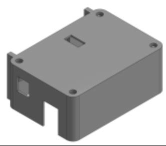
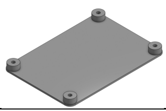
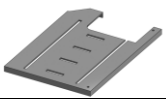
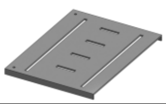
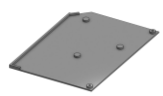
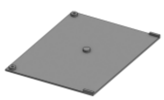
Designator	**Component**	**Number**	**Cost per unit – CAD**	**Total cost – CAD**	**Source of materials**	**Material type**
C10,C9,C8,C7,C6,C5	CAP_10nF_1206_3216Metric_Pad1.33x1.80mm_HandSolder	6	$0.28	$1.68	Digikey	Semi-conductor
U2,U1	LMD18200_TO-220-11_P3.4x5.08mm_StaggerOdd_Lead4.85mm_Vertical	2	$28.28	$56.56	Digikey	Semi-conductor
A1	Arduino_Nano	1	$11.40	$11.40	Amazon	Semi-conductor
R4,R3	R_2.2k_1206_3216Metric_Pad1.30x1.75mm_HandSolder	2	$0.16	$0.32	Digikey	Semi-conductor
J9,J8,J7,J6	TerminalBlock_TE_282834-2_1x02_P2.54mm_Horizontal	4	$1.42	$5.68	Digikey	Semi-conductor
J5,J4	PinSocket_1x03_P2.54mm_Vertical	2	$0.13	$0.26	Digikey	Semi-conductor
J3,J2	TerminalBlock_TE_282834-4_1x04_P2.54mm_Horizontal	2	$2.15	$4.30	Digikey	Semi-conductor
C4,C3,C1,C2	CAP_10uF_2312	4	$0.83	$3.32	Digikey	Semi-conductor
J1	PinSocket_1x10_P2.54mm_Horizontal	1	$1.17	$1.17	Digikey	Semi-conductor
			Total	**$84.69**		

Table 3.7: SLC_PHDO BOM

Designator	Component	Number	Cost per unit – CAD	Total cost – CAD	Source of materials	Material type
A1	Arduino_Nano_v3.x	1	$11.33	$11.33	Amazon	Semi-conductor
C1, C2	CAP_10uF_2312	2	$0.56	$1.12	Digikey	Semi-conductor
J1	Conn_01x10_Female	1	$0.79	$0.79	Digikey	Semi-conductor
J2, J3	Conn_01x05_Female	2	$0.64	$1.28	Digikey	Semi-conductor
J4, J5, J6, J7	Conn_01x03_Male	4	$0.37	$1.48	Digikey	Semi-conductor
			Total	**$16.00**		

Table 3.8: Slice and Loaf Casing BOM

Component Image	Component	Number	Cost per unit – CAD	Total cost – CAD	Source of materials	Material type
	M3x20 Machine Screw	16	$0.15	$2.40	Digikey	Metal
	M3x10 Machine Screw	20	$0.71	$14.20	Digikey	Metal
	Hex Standoff Threaded M3 Aluminum 0.394" (10.00mm)	16	$0.45	$7.20	Digikey	Metal

	Slice Top Case (39.06g)	3	$1.17*	$3.51	N/A	Polymer
	PHDO Top Case (48.96g)	1	$1.47*	$1.47	N/A	
	Slice Bottom Case (14.35g)	4	$0.43*	$1.72	N/A	Polymer
	Loaf Top Half 1 (66.78g)	1	$2.00*	$2.00	N/A	Polymer
	Loaf Top Half 2 (67.88g)	1	$2.03*	$2.03	N/A	Polymer
	Loaf Bottom Half 1 (75.48g)	1	$2.26*	$2.26	N/A	Polymer
	Loaf Bottom Half 2 (62.81g)	1	$1.88*	$1.88	N/A	Polymer
			Total	**$38.67**		

* Assuming cost of a 1kg spool of PETG as ~$30 (cost per gram = $0.03)

After totaling the BOMs, the total cost of the bioreactor is CAD$1,234.93 (or about USD$925 depending on exchange rate). The total costs for each Slice and Loaf backplane do not include the cost of the printed circuit board (PCB) since this cost may vary widely. Reducing this cost can be accomplished by machining the PCBs with an open-source PCB mill [13,41,42]. A separated top piece is needed for SLC_PHDO because of the Atlas Scientific sensor orientations.

3.5 Build Instructions

The open-source bioreactor is controlled by the BREAD system made up of slices (individual boards to control a specific function).

3.5.1 Slice and Loaf Build Instructions

1. Order the printed circuit board (PCB) to be manufactured using the supplied Gerber files for each slice and Loaf or fabricate it with an open-source PCB mill [41-43].

2. Order the respective parts included in the bill of materials file (BOM) (Section 3.4).

3. Solder the components to the PCB using a hand soldering iron or other tools if accessible (i.e., reflow oven, heat gun, etc.). Refer to the designators in the BOM and the KiCAD files for part placement.

4. 3D print the top and bottom casing out of any rigid filament/material for the Slices and Loaf backplane (Figure 3.14 and Figure 3.15).

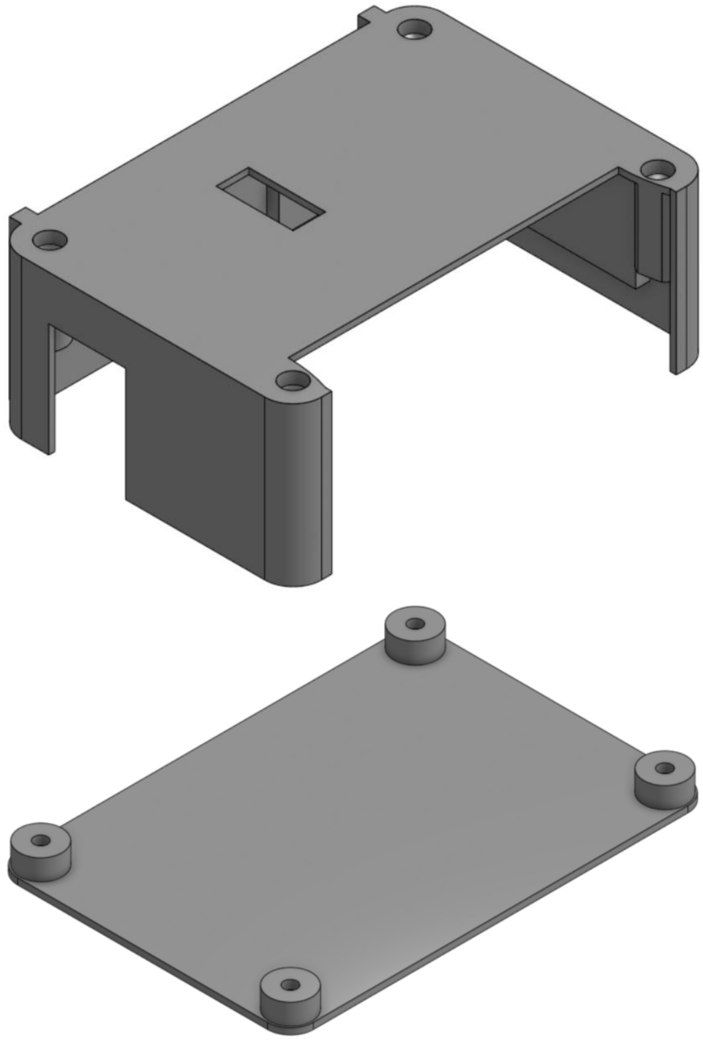


Figure 3.14: Slice Casing Top (top) Bottom (bottom)

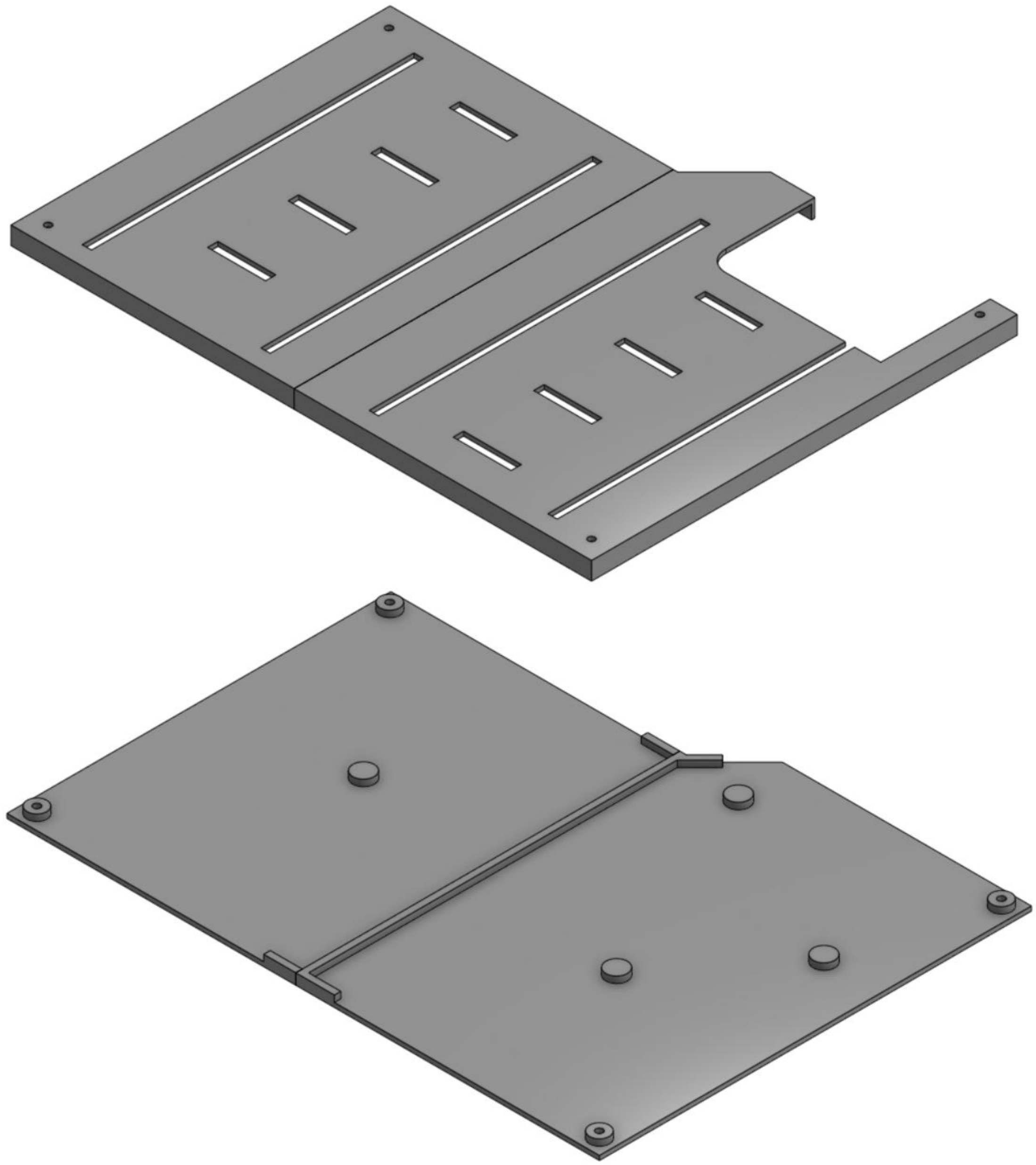


Figure 3.15: Loaf Casing Top (top) Bottom (bottom)

5. For the bioreactor outlined in this paper, the following print settings were used on an open-source Prusa i3 MK3S+:

Table 3.9: Casing Print Settings

Material	PETG
Layer height	0.3mm

Infill	20%
Nozzle temp	235°
Bed temp	85°
Supports	None

6. Assemble the slice with mechanical hardware (Figure 3.16). First fasten the bottom part with M3x10mm bolts and hex standoff. Then fasten the top part using M3x20mm bolts to the other end of the hex standoff (Figure 3.16).

Figure 3.16: Assembled SLC_DCMT with Hex Standoff (circled in yellow)

7. Upload the required firmware to each Slice making sure to provide a unique I^2C address for each. Use the following addresses:

Table 3.10: I2C Addresses for Slices

Slice	Function	Address
SLC_DCMT	Stirring motor, feed pump	1
SLC_DCMT	Acid and base pump	2

SLC_RLHT	Heating, temperature reading	7
SLC_PHDO	pH reading	99
	O_2 reading	97

8. Using hot glue, glue the top and bottom halves of the Loaf casing together, respectively.

9. Install the Loaf Backplane casing with M3x8 bolts and nuts.

10. Plug in each Slice into its respective slot ensuring that the Arduino USB plug faces the middle of the Loaf Backplane (Figure 3.17).

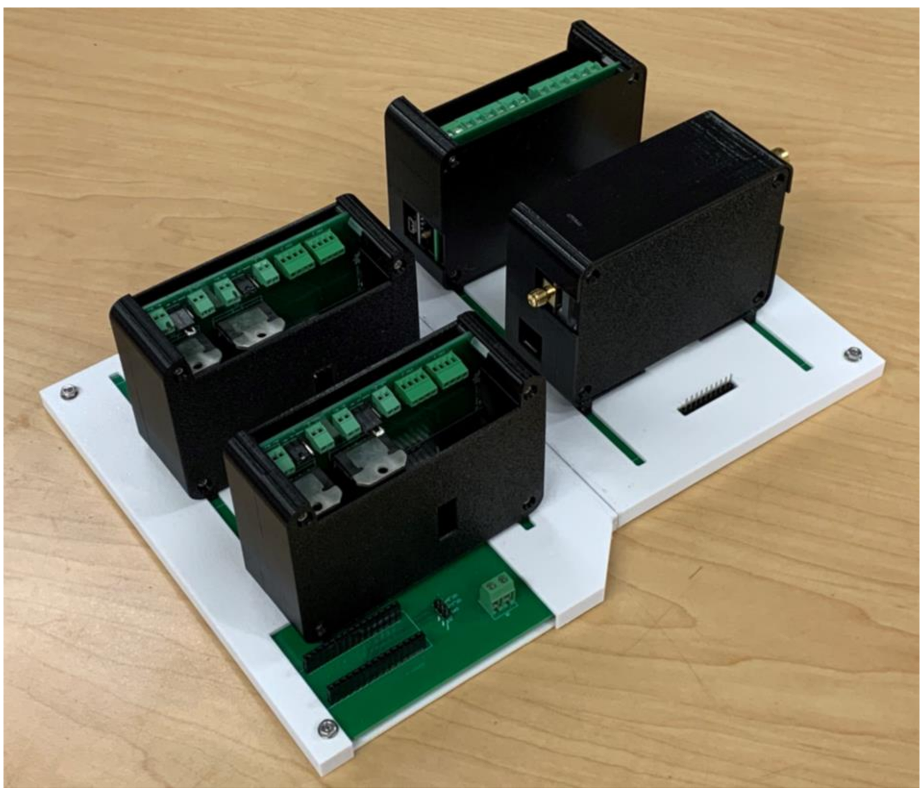


Figure 3.17: Completed BREAD Hardware

3.5.2 Bioreactor Materials

1. Select Appropriate Vessel

a. From any general store or supermarket, select a glass jar with a minimum inner diameter of 98mm and a minimum height of 220mm. Jar lids are recommended to be metal or other autoclavable materials, and it is not recommended to exceed the minimum dimensions by more than 10%.

2. Acquire other BOM items or alternatives.

3. Modify FreeCad files to fit the alternative vessel or parts if needed and print the files in any food safe material on any RepRap-class 3D printer.

3.5.3 Bioreactor Build Instructions

1. Acquire appropriate machining hardware:

   a. Hardware such as a hand drill, drill bits, or stepper bits or alternatives like a drill press or CNC tool.

   b. Hardware must be able to drill the following minimum diameters of holes: 3mm, 4.8mm, 6.4mm, 15mm, 21mm.

2. Machine lid of vessel (Figure 3.18)

   a. 3D print the LidMachiningStencil.stl file in any material and ensure the resulting 3D stencil has clear edges and the diameter of the holes are within 1 mm of the virtual model.

   b. Center the stencil on the inside or outside of the lid

   c. Using a marker, draw all hole outlines on the lid, with the stencil

   d. Use machining hardware to drill the stenciled holes.

   e. File and sand machined lid to ensure there are no sharp edges, all 3D printed mounts, tubing and drive shaft fit in the holes, and no semi-detached metal shavings are present.

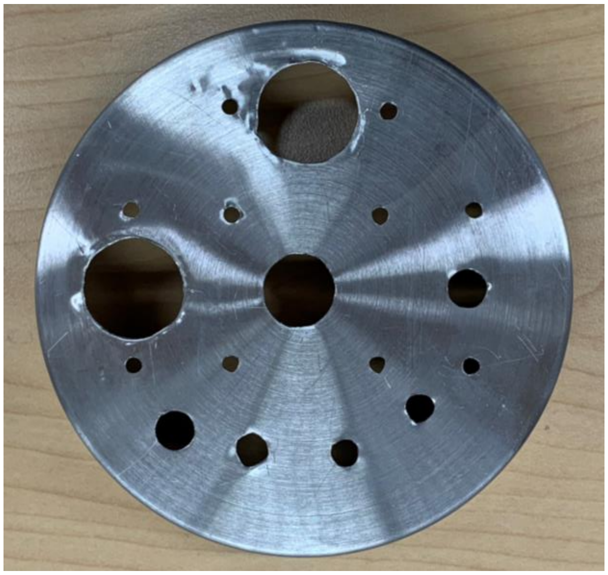


Figure 3.18: Machined Lid

3. Assemble Drive Shaft (Figure 3.19).

   a. Attach aluminum coupler to motor.

   b. Use a sanding belt, angle grinder, or any filing tool to sand 2 cm of one end of the 8 mm linear steel rod flat

   c. Insert 8 mm linear steel rod into aluminum coupler and fasten the screws on the coupler, making sure that the flat side of the motor shaft and flat side of the linear steel rod face the screw holes.

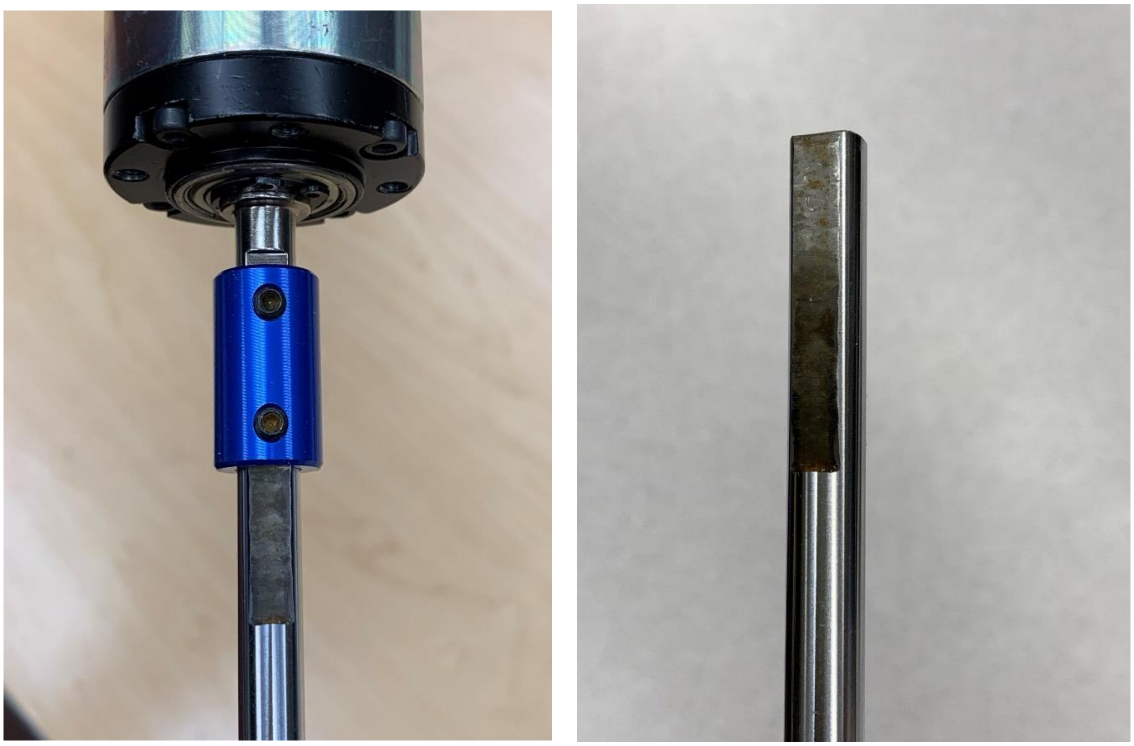


Figure 3.19: Motor Shaft Assembly (a, c: step a and c above, b: step b above)

d. Fasten motor mount to motor using the side of mount with holes closer to inner circle

e. Slide 8mm ID bearing along steel rod until flush with bottom of motor mount

f. Attach motor mount assembly to the machined lid with M3 screws and bolts (Figure 3.20).

g. Attach impeller to end of drive shaft, through friction fit. If too loose, use hole in impeller shaft to screw in M3 screw to provide fastening.

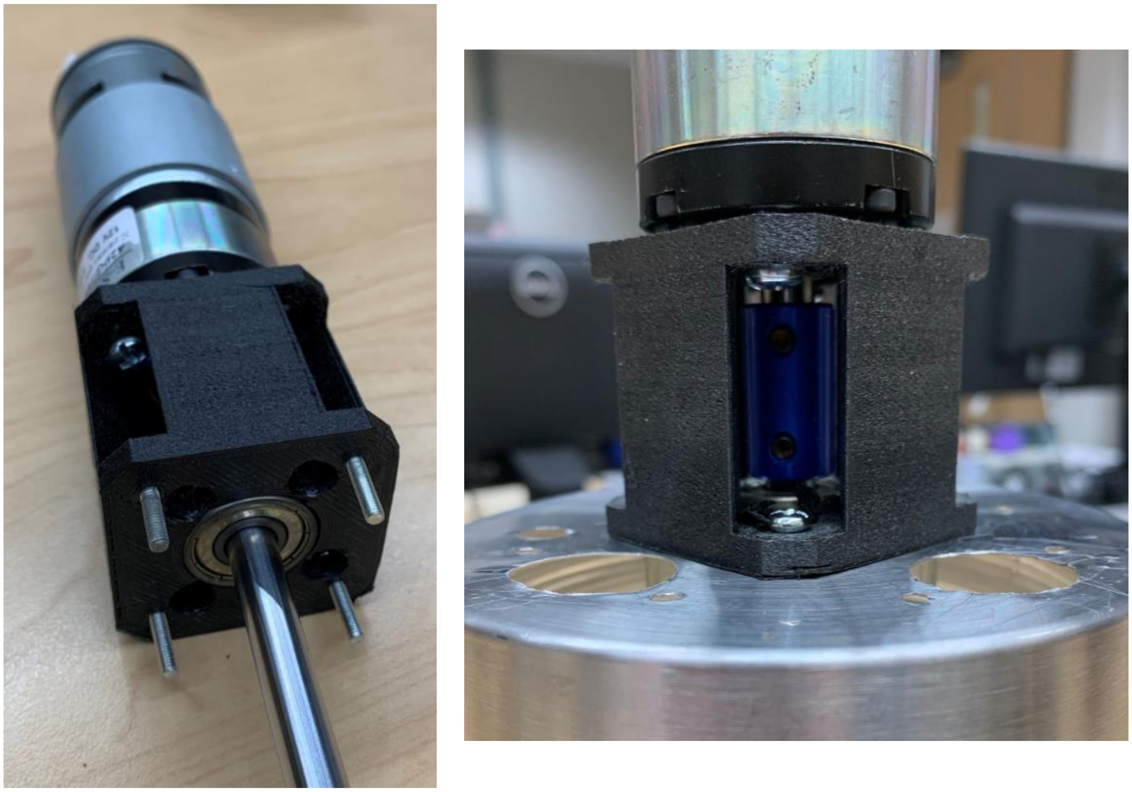


Figure 3.20: Motor Assembly

4. Insert sensors into mounts.

5. Attach mounts to lid in the following holes.

   a. O_2: furthest 21mm hole from center

   b. pH: other 21mm hole

   c. Thermocouple: 4.9mm hole between 2 3mm holes

6. Insert 2mm ID base, acid, and media tubes into respective 4.8 mm holes

7. Assemble O_2 filter

   a. Cut two sections of 6mm ID tubing

   b. Attach air filter ends to two straight pipe reducers

c. Cut one section of 4mm ID tubing

d. Connect pump and air filter with 4mm ID tubing

e. Connect air filter to bioreactor with another section of 4mm ID tubing

8. Insert O_2 tube into 6.4mm hole

9. Attach the stone sparger to the end of the O_2 tube inside the bioreactor.

10. Wrap air supply tubing to O_2 mount with zip ties to prevent tangling with impeller (Figure 3.21).

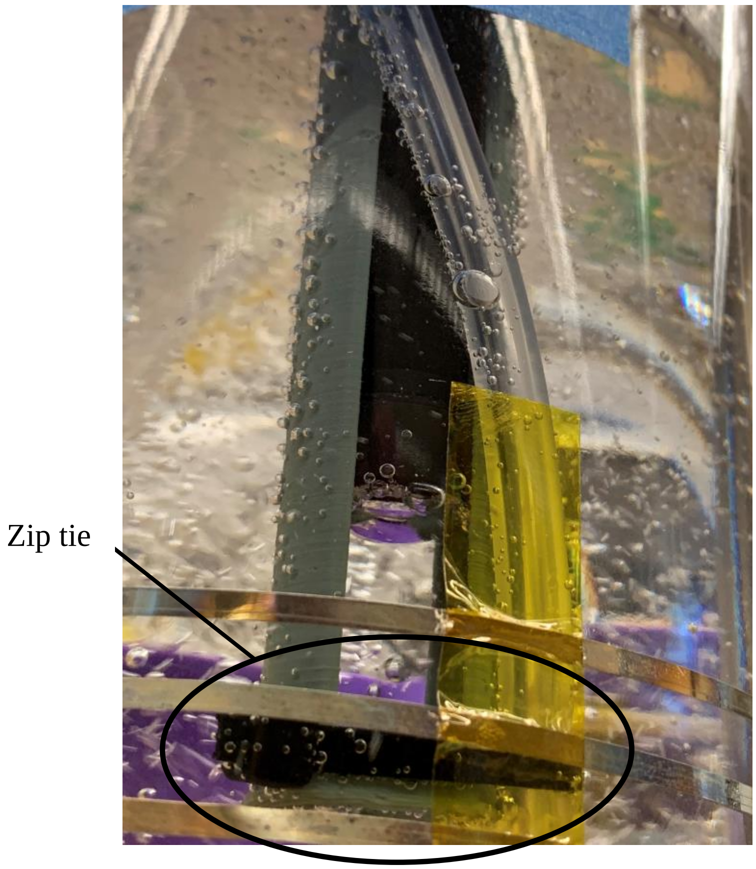


Figure 3.21: Air Tube Attachment Point

11. Attach impeller to drive shaft and stone sparger to tube ending.

12. Assemble Pump Stand, and attach pumps (Figure 3.22)

13. Connect respective tubing to pumps.

a. 2mm ID: acid, base, or media pumps

b. 4mm ID: air pump

14. Connect sensors and pumps to electronic components (Figure 3.22 to Figure 3.25)

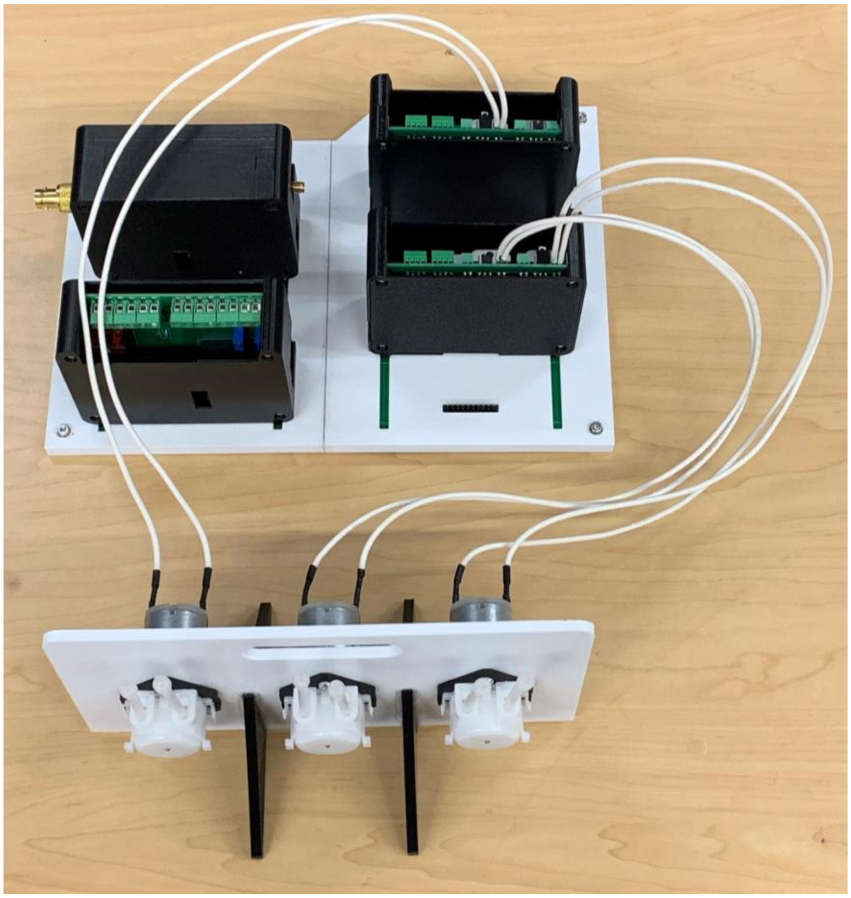


Figure 3.22: Pump Electrical Connections

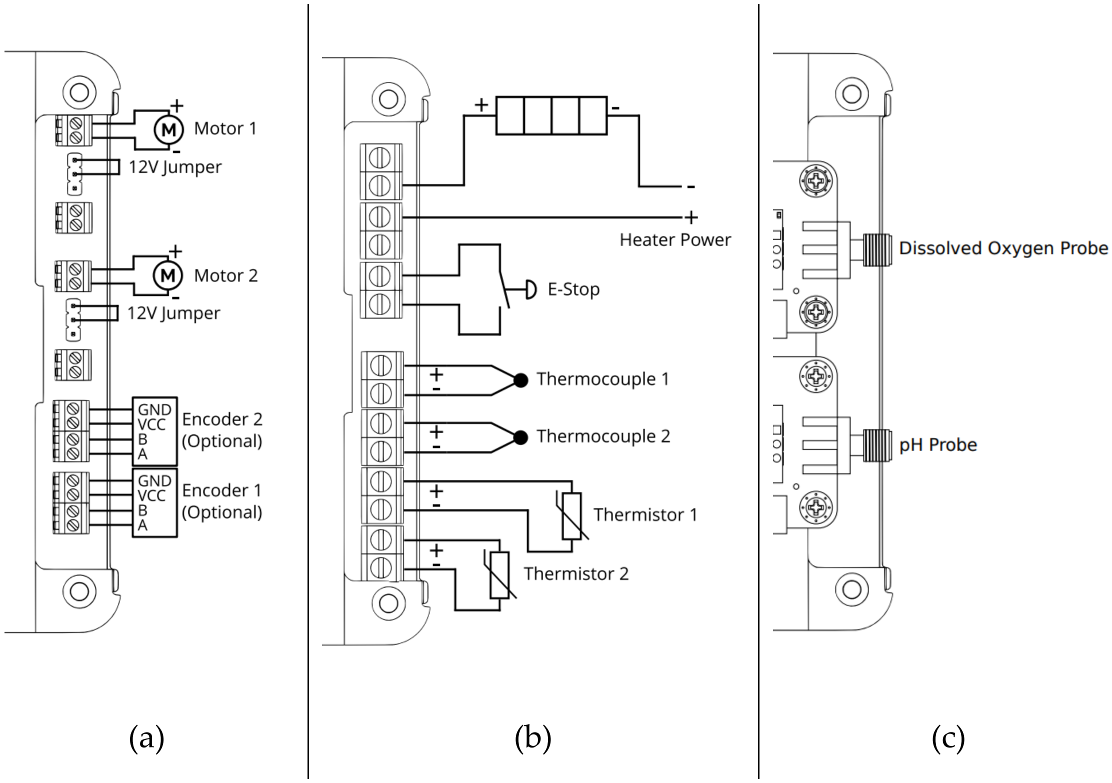


Figure 3.23: BREAD Slice Connections: DCMT (a) RLHT (b) PHDO (c)

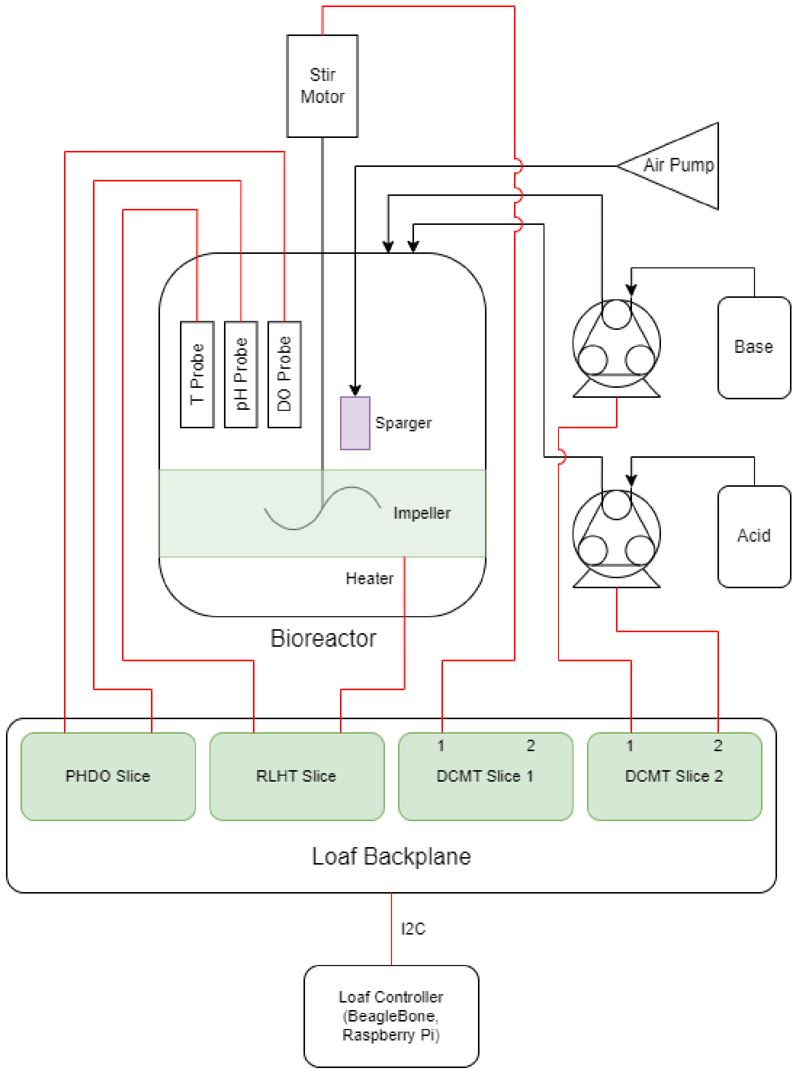


Figure 3.24: Bioreactor Connection Diagram

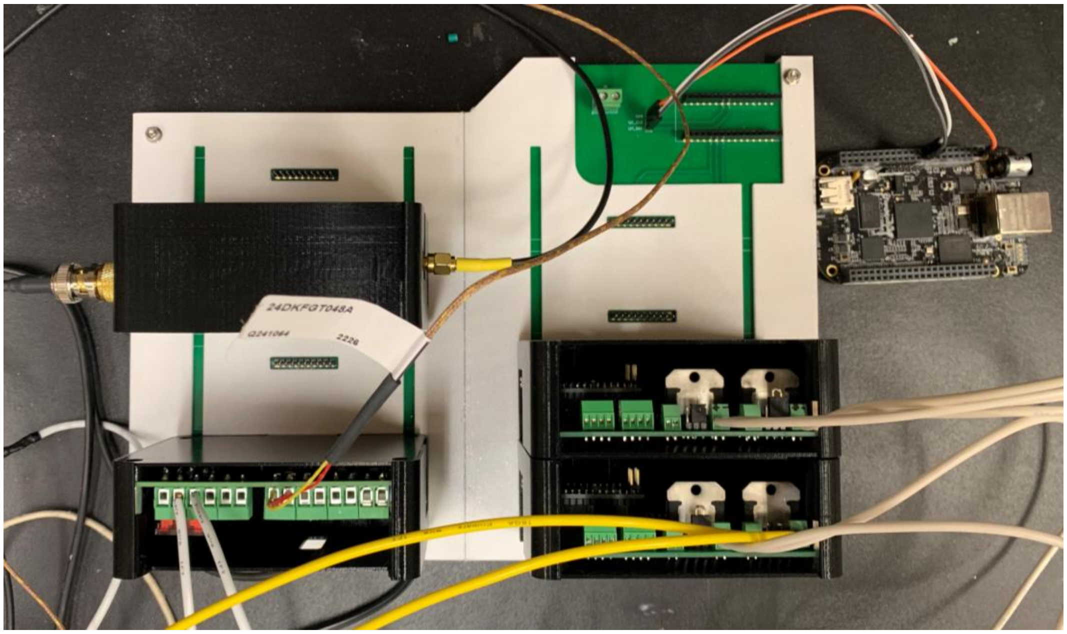


Figure 3.25: Electrical Connections

15. Wrap Ni-Chrome wire around jar and fix with Kapton tape (Figure 3.26).

16. Screw or Attach Assembled Bioreactor Lid onto Jar (Figure 3.27).

17. Place Jar in 3D printed holder and attach wires to Ni-Chrome wire.

18. See Figure 3.28 for the fully assembled bioreactor system.

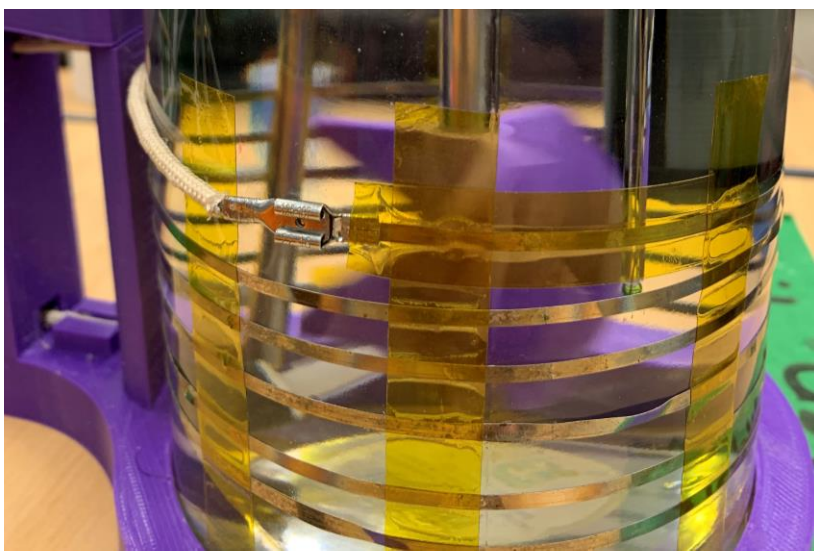


Figure 3.26: Ni-Chrome Wire Assembly

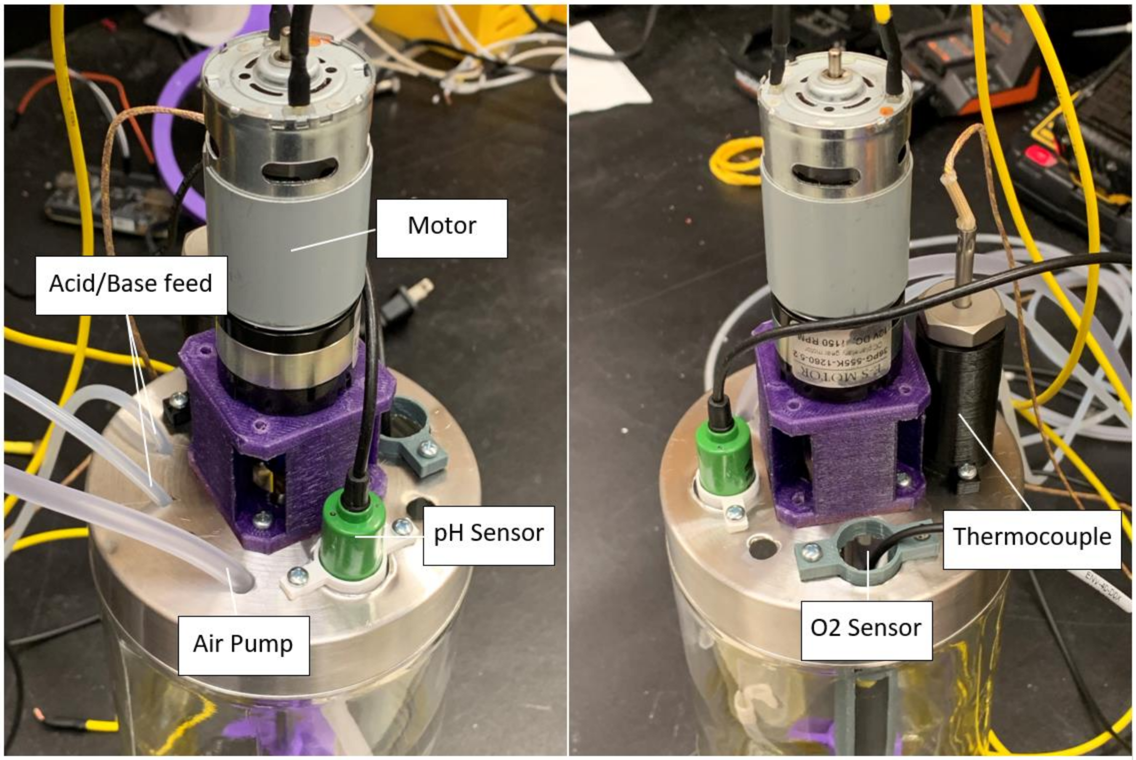


Figure 3.27: Bioreactor Lid Assembly

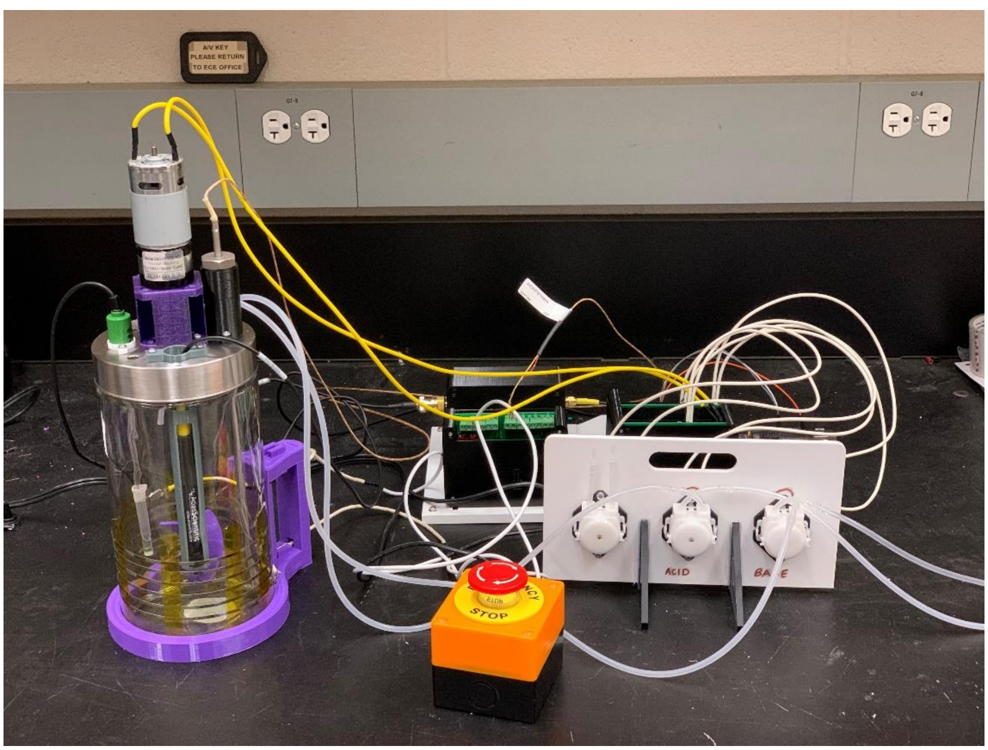


Figure 3.28: Fully Assembled Bioreactor

3.6 Operation Instructions

When setting up a system for the first time, follow the instructions below in order. After the system is operational, the pH and DO sensors no longer need to be calibrated and only the following commands need to be inputted into the PuTTY terminal:

```
git checkout bioreactor2
git pull
sudo ./start.sh
```

3.6.1 Setting up OpenReactor Software on BeagleBone Black

1. Plug in the BeagleBone

2. Open an SSH terminal with PuTTY on the IP address 192.168.7.2

3. Login to the BeagleBone (username: debian, password: temppwd)

4. Input the following commands:

```
# Make sure git is installed
sudo apt install -y git

#Make sure node is installed
sudo apt install nodejs

# Clone and enter repository
git clone https://gitlab.com/mtu-most/most_openreactor
cd most_openreactor

# Run setup script
sudo ./first_time_setup.sh

git checkout bioreactor2
git pull
```

5. Wait until the BeagleBone is finished setup

6. To run the software, input the following command:

```
sudo ./start.sh
```

7. Access the web interface by typing the address displayed into any browser. For example:

```
* Running on http://141.219.193.214:5000/ (Press CTRL+C to
quit)
```

3.6.2 Setting up Bioreactor

1. Remove assembled lid from jar and disconnect electrical components.
2. Remove tubing from pumps.
3. Disinfect and sterilize assembled lid, jar, 250 ml glass storage jars and peristaltic pumps and tube lines with alcohol, acid, or autoclave.
4. Calibrate pH sensor, O_2 sensor, and thermocouple.
5. Attach new PTFE filter to air tubing line.
6. Connect piping to air pump and connect piping to peristaltic pumps.
7. Fill jar with media.
8. Insert and attach assembled lid onto jar.
9. Fill storage containers with respective feed substance, acid, and base.
10. Ensure the tubes are primed and have no air contained in them.
11. Connect sensors and pumps to electrical components.
12. Connect the BeagleBone to any computer and run the OpenReactor software.

3.6.3 Sanitizing Parts

When a sterile environment is required, all 3D printed plastic parts if printed with a low-temperature polymer should be considered single use and replaced after contacting growth media. It should be noted that they can be 3D printed using an open-source high temperature 3D printer that would allow for thermal sterilization [44]. This includes the pH sensor mount, the oxygen probe mount, the thermocouple mount, motor mount, impeller, and jar stand. Additional parts can be printed in PLA and PETG, or other food safe plastics. Additionally, the stone sparger is also single use. For sterilization of fresh parts before an experiment is performed, surface application of alcohol, bleach, or other sanitizing solution should be adequate, and parts should be allowed to dry before use.

The main jar with heating apparatus, acid and base jars, metal lid, impeller shaft, bearing, sensors, nuts and bolts, shaft coupler, and silicone tubing can all be submerged in water to be washed, and then subsequently autoclaved. Just ensure that the Ni-Chrome wire surrounding the jar is completely dry before use. The motor should not be submerged in water and should be sanitized with surface application of alcohol, bleach, or other sanitization solution. Salt buildup may be seen in silicone tubing before or after use, so ensure these are rinsed accordingly. Additionally, screwdrivers and other building tools should be sanitized before use as well.

3.6.4 Calibrating pH Sensor

1. Obtain pH calibration solutions of pH 4.00, 7.00, and 10.00.

2. Place pH probe in pH 7.00 solution (Figure 3.29).

3. Wait for pH readings to stabilize (~1-2 mins).

4. Go to the Control page on the OpenReactor software and click “measure” under the “Cal Mid” block. The pH sensor should read pH 7.

5. Repeat steps 2-4 for pH 4 and 10 clicking on “measure” under “Cal Low” and “Cal High” for pH 4 and 10, respectively.

Figure 3.29: pH Sensor Calibration with pH 7.00 Solution

3.6.5 Calibrating O_2 Sensor

1. Place O_2 sensor in still air and wait for readings to stabilize (~5-30 sec).

2. Go to the Control page and click “measure” under the “Air Cal” block. The sensor should read between 9.09-9.1 mg/L.

3.6.6 Priming the Pumps

1. In the PuTTY terminal, edit the “devices.json” file by inputting:

```
nano devices.json
```

2. Add the following lines to the Control section of the file (This changes the functionality of the acid and base pumps so they can be continuously operated)

```
{
      "name":["Base Control","Acid Control"],
      "address":2,
      "unit":["",""],
```

```
        "form":["byte","byte"],
        "req_msg":[[],[]],
        "delay":[0.0,0.0],
        "read_length":[4,4],
        "enabled":[false,false],
        "params":[[{
           "speed":0,
           "control":"control.BREADmotor_I"
         }],
         [{
           "speed":0,
           "control":"control.BREADmotor_II"
         }]],
         "def_state":[false,false]
},
```

3. Press “CTRL+S” and “CTRL+X” to save and exit.

4. Run the software.

5. In the web interface navigate to the Control page, enable the acid and base pumps, and set the speed to 100.

6. Once the pumps are turning, place both ends of the tubing into their respective solution tanks.

7. When no more bubbles are coming out of the tubes, stop the pumps by pressing “Reset” on the Control page.

8. Place the output ends of each pump into their respective holes in the lid of the bioreactor.

3.7 Validation and Characterization

Heating control was tested by filling the bioreactor to 50% capacity with room temperature water. With the thermocouple inserted and the stirring speed at 700RPM, the water was heated to a temperature of 35°C. The temperature change was logged over time (Figure 3.30). Once stable, the heater kept the water temperature within +/- 1°C. It took about 12 mins to reach the desired temperature from 25°C.

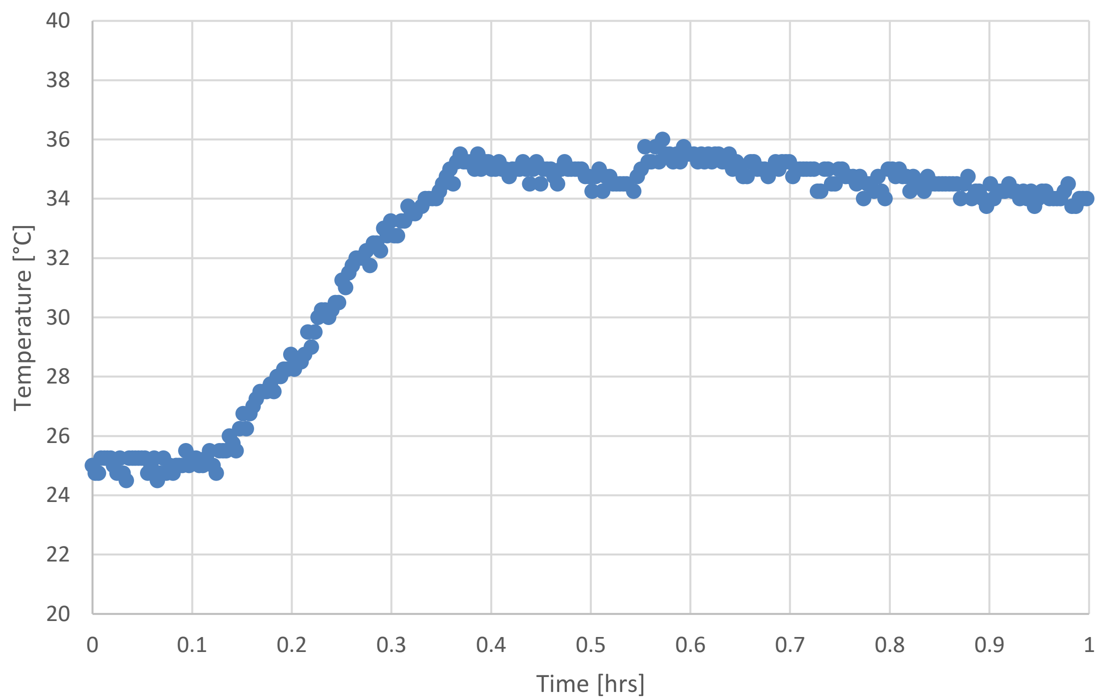


Figure 3.30: Temperature Precision Test

Control of the pH of the bioreactor was tested by filling the bioreactor to 50% capacity with a water and lactic acid solution of pH 3.8. The base peristaltic pump reservoir was filled with a sodium bicarbonate solution and inserted into the bioreactor along with the pH probe. The results are shown in Figure 3.31. Once stable, the pH was maintained within pH +/- 0.035. The large jumps in the results were caused by injecting more lactic acid solution into the bioreactor to test the system's recovery. On average, the system recovered within 10 minutes, which is sufficient for most bioreactor applications.

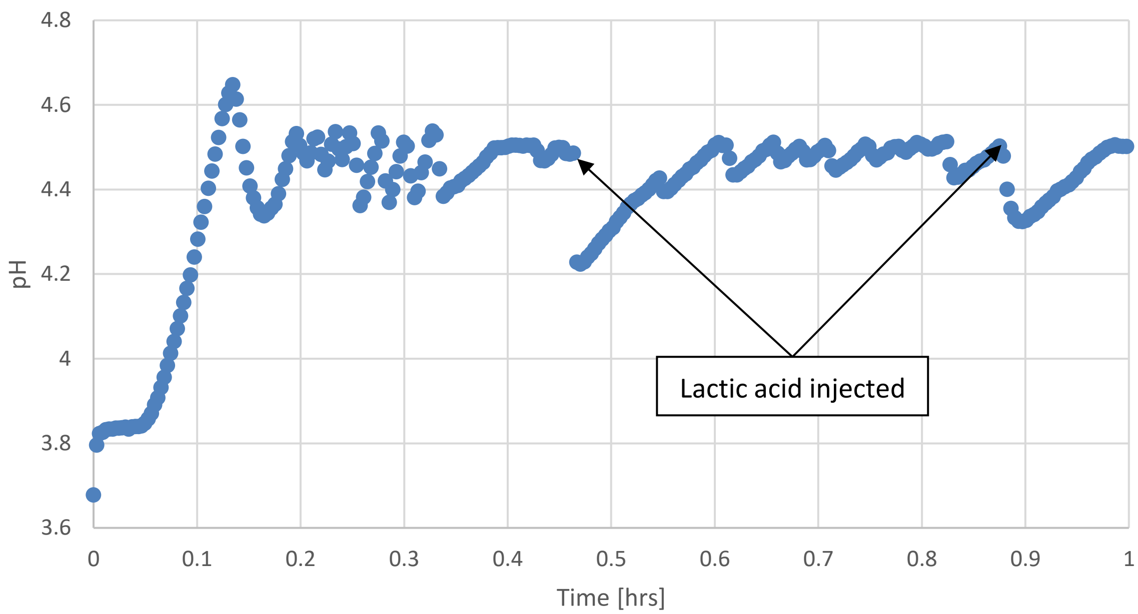


Figure 3.31: pH Precision Test (lactic acid injected at 0.45hrs and 0.87hrs)

The precision of all aspects of the system are listed below based on the previous tests and their respective datasheets (Table 3.11).

Table 3.11: System Characteristics

Characteristic	Maximum	Minimum	Precision
Heating control	100°C	Room temperature	+/- 1°C
pH control	pH 14	pH 0.001	pH +/- 0.035
O_2 sensor	100 mg/L	0.01 mg/L	+/- 0.05 mg/L
pH sensor	pH 14	pH 0.001	pH +/- 0.002
Thermocouple	482°C	0°C	+/- 0.75%
Stirring motor	1000 rpm	0 rpm	N/A
Air pump	30GPH	0GPH	N/A

The growth of Saccharomyces cerevisiae (baker's yeast) in the bioreactor was performed in order to validate the subsystems of the bioreactor (Figure 3.32). Using active dry baker's yeast bought from the supermarket, YPD broth, lactic acid, and baking soda, 8.5 grams of dry S. cerevisiae were fermented at 30 degrees Celsius and 5.5 PH. The experiment lasted 3 hours because this was indicated as the approximate amount of time needed to double the mass of the yeast. The dissolved oxygen spike at around 3.2hrs in Figure 3.33 indicated that the yeast was finished growing so the reaction was stopped. The yeast was drained and dried on a 3D printer build plate set to 50°C before weighing (Figure 3.34 - Figure 3.37). The experimental results lead to a 1.57x increase in yeast mass from 8g to 12.578g which indicates that the bioreactor can sustain a healthy environment to promote growth.

An anti-foaming agent was not used in these experiments and led to the bioreactor overflowing. To prevent a short circuit of the heating wire, the heater was disabled for the experiment; however, the fermentation of S. cerevisiae produced the heat required to maintain a desirable temperature for most of the experiment (Figure 3.38). The heating capabilities of this bioreactor have also already been proven with previous tests.

Figure 3.32: Bioreactor Experiment

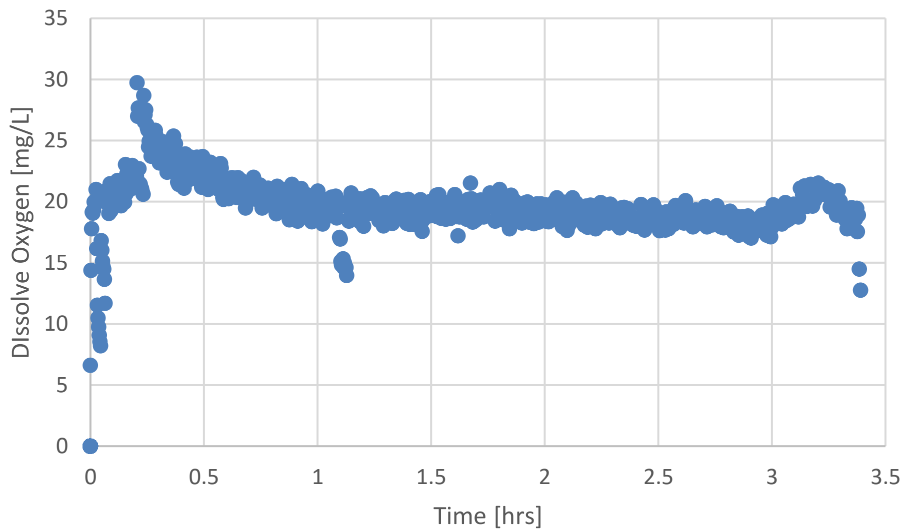


Figure 3.33: Yeast Growth Experiment: Dissolved Oxygen over Time

Figure 3.34: Settled Yeast After Two Days

Figure 3.35: Draining Yeast Samples

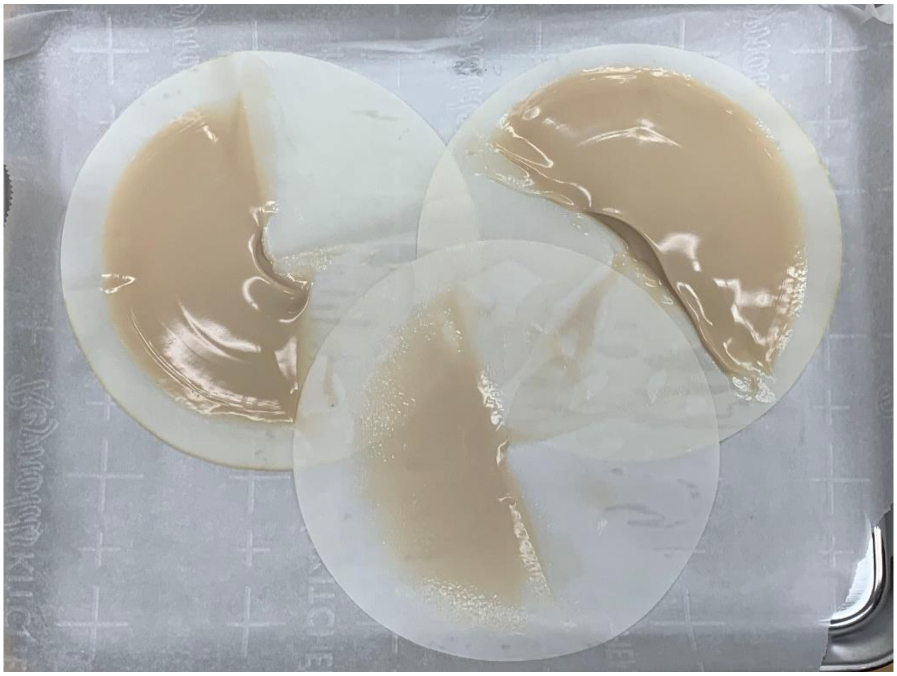


Figure 3.36: Yeast Samples Drying

Figure 3.37: Dried yeast samples (Samples were dried on a desktop 3D printer build plate set to 50°C)

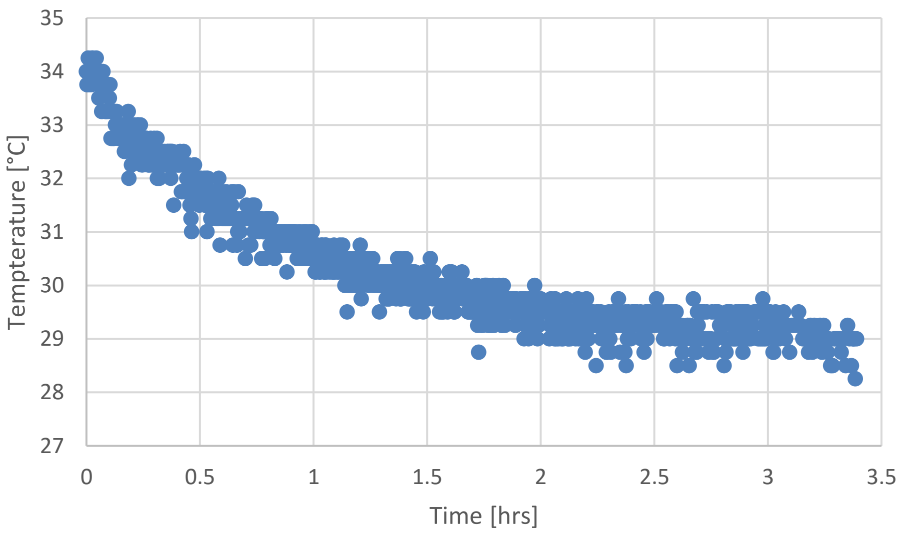


Figure 3.38: Yeast Growth Experiment: Temperature over Time

3.8 Safety

When using the bioreactor, proper PPE and standard operating procedure should be observed at all times. This includes, but is not limited to, safety glasses, lab coats or protective wear, safety gloves, and closed toe shoes. When running, the spinning impeller shaft is exposed through holes in the motor mount. Ensure that no loose clothing, wires, or tubing are close to the inside of the motor mount, and that it is not disturbed while in use.

The heating apparatus on the bioreactor is exposed while in use. Accidental contact with heating elements could result in burns. Ensure that heating coil wires are not touched or shorted while bioreactor is in operation. Alternatively, the user can cover the bioreactor with insulation to prevent this hazard but loses the ability of visual observation of the reactor.

Many wires and connections remain exposed while in operation. Ensure electrical components are dry before use, and that wire connections and electrical components are not touched or made wet during use.

Strong acids or bases may be used while working with the bioreactor, and may be present in tubes before, during, or after use. Ensure caution is used when clearing tubes and cleaning components.

Finally, ensure that bioreactor is on a stable surface and the lid remains sealed and well-ventilated for experiments where foaming occurs. If bioreactor contents are spilled, ensure correct cleaning procedures are observed.

3.9 Conclusions and Future Work

The bioreactor was operated continuously for a week without any issues but to test this system further, longer experiments should be conducted with different bacterial cultures and continuous harvesting. Additionally, the latency of oxygen dissolving in the media and heat transferring to the media from a physical and sensing perspective should be characterized. This would improve dynamic accuracy of the sensors and provide a better understanding of how each biological system is behaving. As this system was designed to be parametric there are numerous areas of future adaptations and testing. First, testing different sizes of bioreactors can be accomplished, which are primarily governed by the volume of the selected locally sourced container. Next, stirring with an air pump rather than the impeller would simplify the bioreactor system. In such a system convection mixing by using the air pump would be used both with and without and a vertical tube inserted into the bioreactor center. These types of bioreactors are referred to as airlift bioreactors [45] and have the advantage of lower costs (removing the mechanical mixer) but reduce the control as stirring and aeration are coupled. Future work could also explore insulating the heating elements and testing different types of heaters that are accessible in different locations. In addition, safety secondary heat monitoring could be added to prevent thermal run away. This addition is recommended for larger systems and those that are unmonitored for long runs. Also, a third pump can be added for supplementing feed mixture to prevent loss from evaporation for longer experiments. For bioreactors using an impeller as demonstrated here different 3D printable designs can be tested (e.g. rushton-style impeller) along with different motor types to determine lowest-cost alternative that can still achieve 1000 rpm in most mixing conditions. Because the bioreactor system outlined in this article is installed in the lid of a mixing vessel, larger volume systems could be easily adapted. The system used to control and monitor the bioreactor, BREAD, has many of the electrical and mechanical features of the open source Chi.Bio. system [46]. Algorithms used for in situ characterization could be adapted from the Chi.Bio. system to the BREAD framework to expand its functionality. To move away from single use parts future work could explore instituting chemical sealing using chemical compatibility information [47] or thermal sealing [48] or mechanical sealing measures on 3D printed parts and impeller shaft to prevent substances from exiting the bioreactor, such as rubber o-rings, sealed bearings, and waterproofing. Future work can improve the design by adding shielding for electrical connectors on motor as well as sourcing quick connect/disconnects to replace screws and make building easier. The mechanical design can be improved by the conversion of the motor mount to two piece connecting mount, which would allow for the removal of the motor without detaching lid. Similarly, there are potential improvements to add a 3D printed stand/case for adding storage space for pumps and external storage jars. The functionality could be pushed further by adding a cell density sensor for tracking of cell growth. Finally, there are many applications this bioreactor can be adapted for to adapt the focused work of prior open-source bioreactors including tissue cultures [49], gas fermentation [50], anaerobic membrane separation [51], coupling to light microscopy [52] and algal photosynthesis [53] by making it a photo-bioreactor by adding lights.

3.10 References

[1] G. Lidén, 2002. Understanding the bioreactor. *Bioprocess and biosystems engineering*, *24*, pp.273-279.

[2] Y. Yusuf Chisti, and M. Moo-Young, 2003. Bioreactors in Myers, R.A. *E. of Physical Science and Technology*. Elsevier.

[3] A.M. Chagas, Haves and have nots must find a better way: The case for open scientific hardware, PLOS Biology 16 (9) (2018), https://doi.org/10.1371/journal.pbio.3000014 e3000014

[4] Biostat B - Benchtop Bioreactor Controller | Sartorius. (n.d.). Retrieved April 18, 2023, from https://www.sartorius.com/en/products/fermentation-bioreactors/benchtop-bioreactors/biostat-b

[5] BioFlo 120—Eppendorf Canada. (n.d.). Retrieved April 18, 2023, from https://www.eppendorf.com/ca-en/eShop-Products/Bioprocess/Bioprocess-Controllers/BioFlo-120-p-PF-176595

[6] Rocker Bioreactor—CA. (n.d.). Retrieved April 18, 2023, from https://www.thermofisher.com/ca/en/home/life-science/bioproduction/single-use-bioprocessing/single-use-equipment/single-use-bioreactors/rocker-bioreactor.html

[7] DynaDrive Single-Use Bioreactor—CA. (n.d.). Retrieved April 18, 2023, from https://www.thermofisher.com/ca/en/home/life-science/bioproduction/single-use-bioprocessing/single-use-equipment/single-use-bioreactors/dynadrive-sub.html

[8] Bioprocessing Single-Use Bioreactors. (n.d.). Retrieved April 18, 2023, from https://www.sigmaaldrich.com/CA/en/products/pharma-and-biopharma-manufacturing/bioprocessing-cell-culture/single-use-bioreactors

[9] Stirred Tank Bioreactors—Cell Culture | Pall Corporation. (n.d.). Retrieved April 18, 2023, from https://www.pall.com/en/biotech/technologies/cell-culture/stirred-tank-bioreactors.html

[10] BPC Bioreactors: CSTR & High-Rate AD Reactors Series | BiogasWorld. (n.d.). Retrieved April 18, 2023, from https://www.biogasworld.com/product/anaerobic-digestion/bpc-bioreactors-cstr-series/

[11] Labfors 5—The all-rounder among bioreactors | INFORS HT. (n.d.). Retrieved April 18, 2023, from https://www.infors-ht.com/en/bioreactors/bench-top-bioreactors/labfors5/

[12] A. Gibb, Building Open Source Hardware: DIY Manufacturing for Hackers and Makers, Pearson Education, 2014, google-Books-ID: dKnfBQAAQBAJ..

[13] S. Oberloier, J.M. Pearce, General design procedure for free and open-source hardware for scientific equipment, Designs 2 (11) (2018) 2, https://doi.org/10.3390/designs2010002.
[14] J.M. Pearce, Building research equipment with free, open-source hardware, Science 337 (6100) (2012) 1303—1304, https://doi.org/ 10.1126/science.1228183.
[15] J.M. Pearce, *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*, Elsevier, 2013.
[16] E. Canessa, C. Fonda, M. Zennaro, E. Canessa, Low—cost 3D printing for science, education and sustainable development, ICTP Science Dissemination Unit, 2013. url:http://sdu.ictp.it/3D/book.html.
[17] J.M. Pearce, Cut costs with open-source hardware, Nature 505 (74857485) (2014), https://doi.org/10.1038/505618d, 618—618.
[18] E. Gibney, (2016). 'Open-hardware' pioneers push for low-cost lab kit: conference aims to raise awareness of shared resources for building lab equipment. *Nature*, *531*(7593), 147-149.
[19] T. Wenzel, (2023). Open hardware: From DIY trend to global transformation in access to laboratory equipment. *PLoS biology*, *21*(1), e3001931.
[20] S. Mishra, V. Singh, L. Cheng, A. Hussain, and B. Ormeci, “Nitrogen removal from wastewater: A comprehensive review of biological nitrogen removal processes, critical operation parameters and bioreactor design,” J. Environ. Chem. Eng., vol. 10, no. 3, p. 107387, Jun. 2022, doi: 10.1016/j.jece.2022.107387.
[21] C. Lepine, L. Christianson, J. Davidson, and S. Summerfelt, “Woodchip bioreactors as treatment for recirculating aquaculture systems’ wastewater: A cost assessment of nitrogen removal,” Aquac. Eng., vol. 83, pp. 85–92, Nov. 2018, doi: 10.1016/j.aquaeng.2018.09.001.
[22] S. T. Magar, “Bioreactor- Definition, Design, Principle, Parts, Types, Applications, Limitations,” Sep. 07, 2022. https://microbenotes.com/bioreactor/ (accessed Apr. 14, 2023).
[23] S. Dahiya and S. Venkata Mohan, “Synergy of selective buffering, intermittent pH control and bioreactor configuration on acidogenic volatile fatty acid production from food waste,” Chemosphere, vol. 302, p. 134755, Sep. 2022, doi: 10.1016/j.chemosphere.2022.134755.
[24] H. Falahti-Marvast and A. Karimi-Jashni, “Performance of simultaneous organic and nutrient removal in a pilot scale anaerobic–anoxic–oxic membrane bioreactor system treating municipal wastewater with a high nutrient mass ratio,” Int. Biodeterior. Biodegrad., vol. 104, pp. 363–370, Oct. 2015, doi: 10.1016/j.ibiod.2015.07.001.
[25] C. M. Narayanan and V. Narayan, “Biological wastewater treatment and bioreactor design: a review,” Sustain. Environ. Res., vol. 29, no. 1, p. 33, Dec. 2019, doi: 10.1186/s42834-019-0036-1.

[26] S. Al-Asheh, M. Bagheri, and A. Aidan, “Membrane bioreactor for wastewater treatment: A review,” Case Stud. Chem. Environ. Eng., vol. 4, p. 100109, Dec. 2021, doi: 10.1016/j.cscee.2021.100109.

[27] A. Ahmad et al., “Chapter 21 - Bioreactor in petroleum refinery wastewater treatment: challenges and future prospects,” in Bioremediation for Environmental Sustainability, V. Kumar, G. Saxena, and M. P. Shah, Eds., Elsevier, 2021, pp. 467–479. doi: 10.1016/B978-0-12-820318-7.00021-6.

[28] “Pioreactor 20ml,” Pioreactor. https://pioreactor.com/en-ca/products/pioreactor-20ml (accessed Apr. 14, 2023).

[29] V. Gopalakrishnan et al., “A Low-Cost, Open Source, Self-Contained Bacterial EVolutionary biorEactor (EVE).” bioRxiv, p. 729434, Oct. 10, 2019. doi: 10.1101/729434.

[30] “Team:Aachen/Lab/Bioreactor/Hardware - 2015.igem.org.” https://2015.igem.org/Team:Aachen/Lab/Bioreactor/Hardware (accessed Apr. 14, 2023).

[31] “Open Source Bioreactor - Hackuarium.” https://wiki.hackuarium.ch/w/Open_Source_Bioreactor#History (accessed Apr. 14, 2023).

[32] Pearce, J.M. Economic savings for scientific free and open source technology: A review, HardwareX 8 (2020), https://doi.org/10.1016/j.ohx.2020.e00139 e00139.

[33] M. Banzi, & M. Shiloh, (2022). *Getting started with Arduino*. Maker Media, Inc..

[34] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, A. Bowyer, Reprap — the replicating rapid prototyper, Robotica 29 (1) (2011) 177—191, https://doi.org/10.1017/S026357471000069X.

[35] E. Sells, S. Bailard, Z. Smith, A. Bowyer, V. Olliver, RepRap: The Replicating Rapid Prototyper: Maximizing Customizability by Breeding the Means of Production, World Scientific Publishing Company, Ch. 28 (2009) 568—580, https://doi.org/10.1142/9789814280280_0028,

[36] Bowyer, A. (2014). 3D printing and humanity's first imperfect replicator. *3D printing and additive manufacturing*, *1*(1), 4-5.

[37] Oberloier, S.; Whisman, N.G.; Hafting, F.; Pearce, J.M. Open Source Framework for a Broadly Expandable and Reconfigurable Data Acquisition and Automation Device (BREAD). *HardwareX* **2023**, *15*, e00467. https://doi.org/10.1016/j.ohx.2023.e00467.

[38] T. Baden, A.M. Chagas, G. Gage, T. Marzullo, L.L. Prieto-Godino, T. Euler, Open labware: 3-d printing your own lab equipment, PLOS Biology 13 (3) (2015), https://doi.org/10.1371/journal.pbio.1002086 e1002086.

[39] M. Coakley, D.E. Hurt, 3d printing in the laboratory: Maximize time and funds with customized and open-source labware, Journal of Laboratory Automation 21 (4) (2016) 489—495, https://doi.org/10.1177/2211068216649578

[40] FreeCAD: Your own 3D parametric modeler. (n.d.). Retrieved April 18, 2023, from https://www.freecad.org/

[41] G. C. Anzalone, B. Wijnen, & J. M. Pearce, (2015). Multi-material additive and subtractive prosumer digital fabrication with a free and open-source convertible

delta RepRap 3-D printer. *Rapid Prototyping Journal*. Vol. 21 No. 5, pp. 506-519. https://doi.org/10.1108/RPJ-09-2014-0113

[42] N. Sathyakumar, Balaji, K. P., Ganapathi, R., & Pandian, S. R. (2018). A build-your-own three axis CNC PCB milling machine. *Materials Today: Proceedings*, *5*(11), 24404-24413.

[43] S. Oberloier, & J.M. Pearce (2018). Belt-driven open source circuit mill using low-cost 3-D printer components. *Inventions*, *3*(3), 64.

[44] N.G. Skrzypczak, N. G., Tanikella, J.M. & Pearce, (2020). Open source high-temperature RepRap for 3-D printing heat-sterilizable PPE and other applications. *HardwareX*, *8*, e00130.

[45] J. C. Merchuk, (2003). Airlift bioreactors: review of recent advances. *The Canadian Journal of Chemical Engineering*, *81*(3-4), 324-337.

[46] H. Steel, R. Habgood, C. L. Kelly, & A. Papachristodoulou, (2020). In situ characterisation and manipulation of biological systems with Chi. Bio. *PLoS biology*, *18*(7), e3000794.

[47] I. T. S. Heikkinen *et al.*, "Chemical compatibility of fused filament fabrication-based 3-D printed components with solutions commonly used in semiconductor wet processing," *Additive Manufacturing*, vol. 23, pp. 99–107, Oct. 2018, doi: 10.1016/j.addma.2018.07.015.

[48] P. J. Mayville, A. L. Petsiuk, and J. M. Pearce, "Thermal Post-Processing of 3D Printed Polypropylene Parts for Vacuum Systems," *Journal of Manufacturing and Materials Processing*, vol. 6, no. 5, Art. no. 5, Oct. 2022, doi: 10.3390/jmmp6050098.

[49] A. Daneshgar, et al. (2019). Teburu—Open source 3D printable bioreactor for tissue slices as dynamic three-dimensional cell culture models. Artificial Organs, 43(10), 1035-1041

[50] C. M. Klask, N. Kliem-Kuster, B. Molitor, and L.T. Angenent, L. T. An open-source multiple-bioreactor system for replicable gas-fermentation experiments: Nitrate feed results in stochastic inhibition events, but improves ethanol production of Clostridium ljungdahlii with CO2 and H2. BioRxiv.

[51] A. D., Kappell, et al. (2018). Removal of antibiotic resistance genes in an anaerobic membrane bioreactor treating primary clarifier effluent at 20 C. Environmental Science: Water Research & Technology, 4(11), 1783-1793.

[52] I.M. Sauer, et al. (2005). The SlideReactor—a simple hollow fiber based bioreactor suitable for light microscopy. Artificial organs, 29(3), 264-267.

[53] A.L. Karam, et al. (2017). Construction and setup of a bench-scale algal photosynthetic bioreactor with temperature, light, and ph monitoring for kinetic growth tests. JoVE (Journal of Visualized Experiments), (124), e55545.

Chapter 4

4 Converting the Open Source Broadly Reconfigurable and Expandable Automation Device (BREAD) into a Full Supervisory Control and Data Acquisition System

While the Broadly Reconfigurable and Expandable Automation Device (BREAD) has demonstrated its functionality as an inexpensive replacement for many commercial controllers, some aspects of its design require updating to make it more aligned with commercial supervisory control and data acquisition (SCADA) systems. These updates made to BREAD were labeled as BREAD v2 and the new design was integrated into an electrical enclosure for airline-based pH control. After comparing the BREAD controller to a commercial equivalent, BREAD was found to be more precise, less expensive, and offered additional functionality like data logging.

4.1 Hardware in Context

The Broadly Reconfigurable and Expandable Automation Device (BREAD) framework was created to address the needs of scientists and researchers [1]. Within the realm of controllers, simple tasks like pH control and heating become expensive when data logging capabilities are also desired. BREAD is less expensive than many of these controllers, provides more functionality in terms of control customizability, and has data logging capabilities [1].

Supervisory Control and Data Acquisition (SCADA) systems are commonly used in research and industry to automate experiments and industrial pilot projects [2], [3]. For example, automating biological growth experiments within a bioreactor [4] or controlling the heating zones within a pyrolysis reactor to break down plastic waste into fuel, oil, and wax [5]. SCADA systems consist of physical hardware responsible for controlling actuators like heaters and motors and gathering sensor data, and software to display these data and allow users to interact with the system. Unfortunately, these hardware/software packages, like National Instruments' (NI) CompactRIO [3] and Opto 22's groov EPIC systems [2], are prohibitively expensive for research projects that may require multiple reconfigurations of the system before an optimal process is determined. BREAD has proven to be a significantly less expensive alternative to proprietary SCADA systems without sacrificing functionality (Table 4.1). This in part is due to the well-established economic savings for open hardware [6], as it is built upon the Arduino platform [7]. Despite BREAD's advantages, in its current configuration, it cannot be installed in an electrical enclosure which is typically done for protection and cable organization for industrial systems. Other aspects of BREAD, summarized in Table 4.2, could also be improved to make the system more robust and capable of being integrated into a product.

Table 4.1: Cost Comparison between NI CompactRIO and BREAD for a Bioreactor Control System (approximate costs assume each circuit board is ~$5 from JLCPCB [8])

Function	**NI CompactRIO**	**Cost (CAD)**	**BREAD**	**Cost (CAD)**
Chassis for plugging in modules	NI cRIO-9030	$7,695 [8]	Loaf Backplane	~$10
			Loaf Controller	~$70
Temperature measurement	NI 9210	$855 [9]	RLHT Slice	~$75
Heating/relay control	NI 9482	$440 [10]		
DC motor control	NI 9470	$1,580 [11]	DCMT Slice (x2)	~$97
pH sensing	NI 9203	$1,415 [12]	PHDO Slice	~$715
Dissolve oxygen sensing	NI 9219	$2,815 [13]		
	Total	**$14,800**	**Total**	**$1,064**

This article will summarize the improvements in BREAD that have converted it to a full SCADA system. The resultant BREAD v2 is compared and contrasted with a commercial pH controller used to balance the pH of small-scale seaweed growth tanks as a case study. The changes made to BREAD are summarized in Table 4.2.

Table 4.2: Changes to BREAD v1 and v2.

BREAD Limitation	BREAD v2 Improvement
No Loaf mounting method. BREAD system could be easily knocked over.	Loaf mounted to DIN rail clips.
Loafs could not be physically connected. Limit of 8 Slices per system.	Loaf form factor changed so multiple can be connected. Slice limit based on integrity of I^2C signal (~30 Slices over 1m).
No Slice mounting method. Slices could become unplugged easily.	Slices mounted with two M3 screws to Loaf.
Difficult to align 10-pin Slice connector with Loaf to plug in Slices.	Each 10-pin connector has a 3D printed alignment cover.
Individual Loafs need a full Raspberry Pi (RPi) desktop for normal operation (RPi, monitor, keyboard, mouse). Impractical and expensive for multiple system.	Loaf Controller changed to an ESP32. All communication with Loafs happens wirelessly over a locally hosted Wi-Fi network on the ESP32.
User interface requires user to have some comfort with Linux terminal coding to startup system.	New software, hosted by the ESP32, runs in a Chrome web browser on the user's laptop or computer of choice.

4.2 Hardware Description

While the new BREAD architecture remains the same, some changes to the Slice cases, Loaf Backplane and new Loaf Controller enable the system to be expandable. All changes to the BREAD system are available for this article on the Open Science Framework [15] and the live version is available on GitHub: https://github.com/uwo-fast/BREAD.

Figure 4.1 shows how the Slices plug into the Loaf and all mount to a DIN rail for installation in an electrical enclosure. The new Loaf configuration has 4 Slices per circuit board and multiple Loafs can be connected to create larger systems (Figure 4.2). The new Loaf Controller connects to the left-most Loaf in the system and is responsible for running the software "Butter" [16] (Figure 4.3).

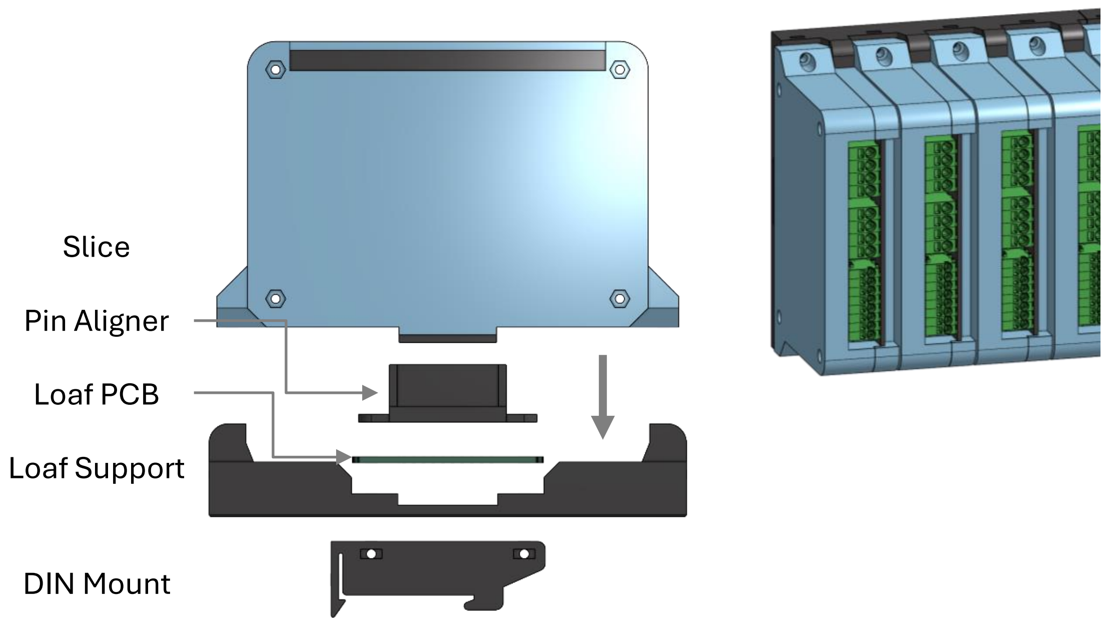


Figure 4.1: Slice Connection Configuration

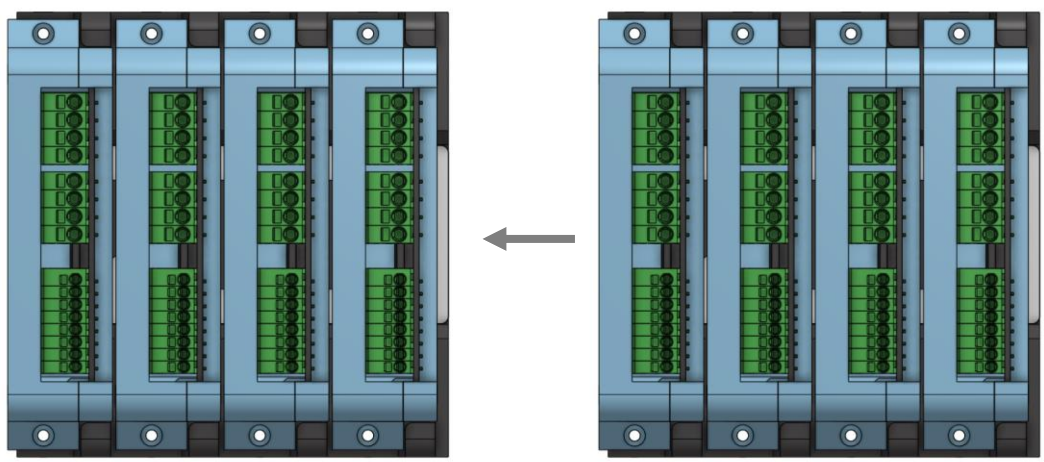


Figure 4.2: Loaf Connection Configuration

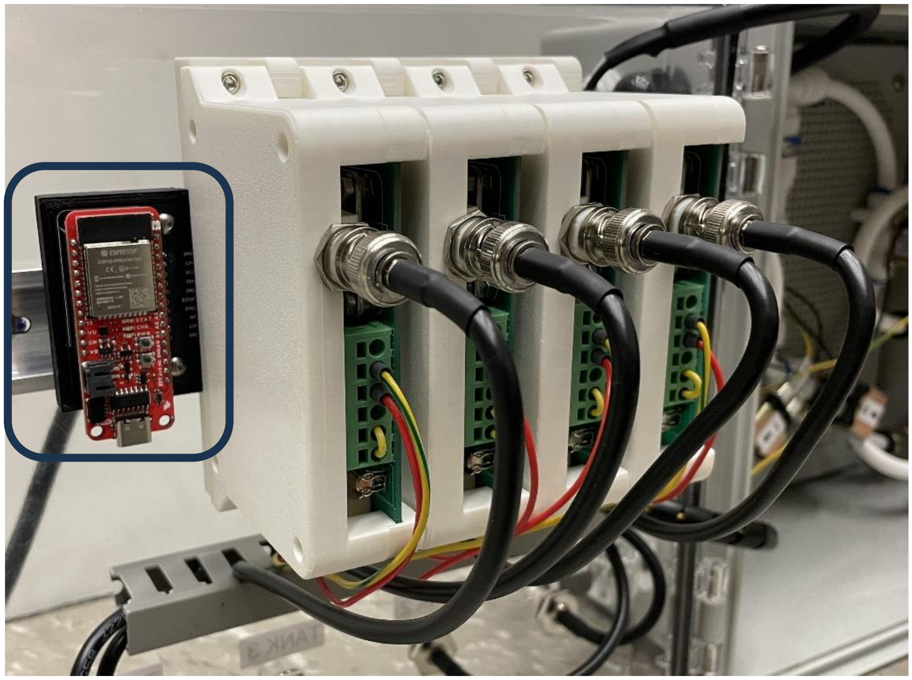


Figure 4.3: Loaf Controller Connection Configuration

New software, called "Butter", was created to facilitate real-time control of each Slice through a convenient web interface hosted by an ESP32 microcontroller (Loaf Controller). It is responsible for logging data from Slices to an SD card, sending commands to Slices when new parameters are assigned by users, and displaying the current state of each Slice [16]. The ESP32 hosts its own, local Wi-Fi network that users first connect to. Then they can type in the IP address "192.168.4.1" into a Chrome web browser where, after 20-30 seconds, the web interface will display allowing the user full control of the BREAD system. Butter was created for a specific BREAD system that combines the pyrolysis reactor [5], bioreactor [4], and a chemical deconstruction reactor. However, as shown later in this chapter, the software can be modified to suit a variety of BREAD configurations.

A high-level bill of materials (BOM) is shown in Table 4.3. This is consistent with every BREAD v2 system; however, the specific Slice PCB will need to be chosen by the user from the OSF repository.

Table 4.3: BREAD v2 General Bill of Materials

BREAD Element	Part	#	URL
Slice	Slice PCB	1	Order through JLCPCB. Gerbers found on OSF: https://osf.io/u2h4g/
	Slice Top Case	1	3D Print: https://osf.io/wyfbr
	Slice Bottom Case	1	3D Print: https://osf.io/ce6zv
	M2x25mm screws	4	https://www.mcmaster.com/92010A111/
	M2 nuts	4	https://www.mcmaster.com/90592A075/
	M3x12mm screws	2	https://www.mcmaster.com/91292A114/
Loaf Backplane	Loaf PCB	1	Order through JLCPCB. Gerbers found on OSF: https://osf.io/u2h4g/
	Loaf Support	1	3D Print: https://osf.io/u453q
	Pin Aligner	1	3D Print: https://osf.io/utq5c
	DIN Mount	2	3D Print: https://osf.io/cna48
	M3 nuts	4	https://www.mcmaster.com/90592A085/
	M3x15mm screws	4	https://www.mcmaster.com/92095A119/
Loaf Controller	Loaf Controller PCB	1	Order through JLCPCB. Gerbers found on OSF: https://osf.io/u2h4g/
	Loaf Controller Case	1	3D Print: https://osf.io/2xwr3
	DIN Mount	1	3D Print: https://osf.io/cna48
	M3 nuts	2	https://www.mcmaster.com/90592A085/
	M3x15mm screws	2	https://www.mcmaster.com/92095A119/

4.3 Build Instructions

1. In-depth instructions on populating Slices with components from their BOM can be found in [1].

2. 3D print all parts in needed quantities.

3. Place M3 nuts within the Loaf Support and DIN Mount parts as shown in Figure 4.4.

4. Assemble Loaf with all parts in Table 4.3 as shown in Figure 4.5.

5. Assemble Slice with all parts in Table 4.3 and shown in Figure 4.6.

6. Connect Slice to Loaf according to Figure 4.7.

7. Assemble Loaf Controller with all parts in Table 4.3 and shown in Figure 4.8.

8. Connect Loaf Controller to Loaf according to Figure 4.8.

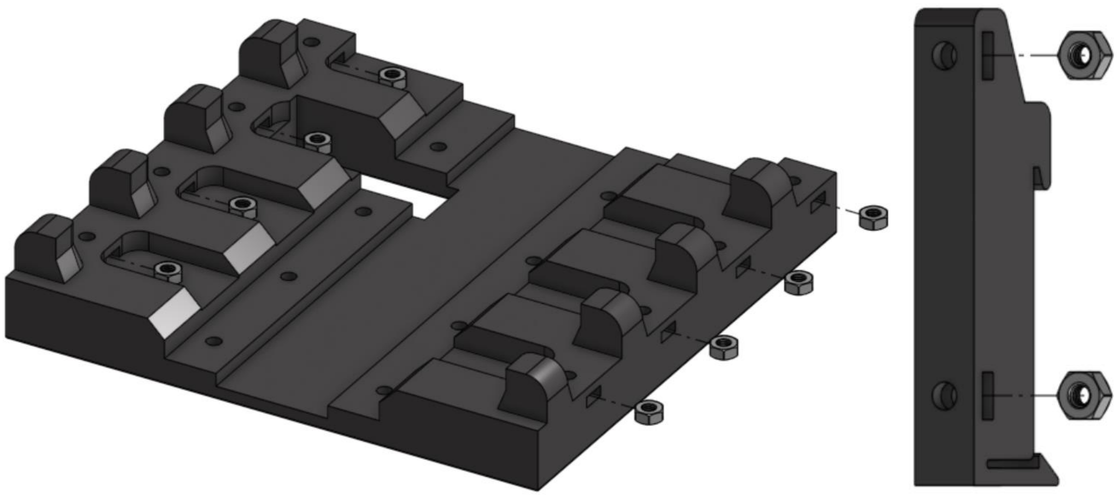


Figure 4.4: M3 Nut Placement in Loaf Support (left) and DIN Mount (right) Parts

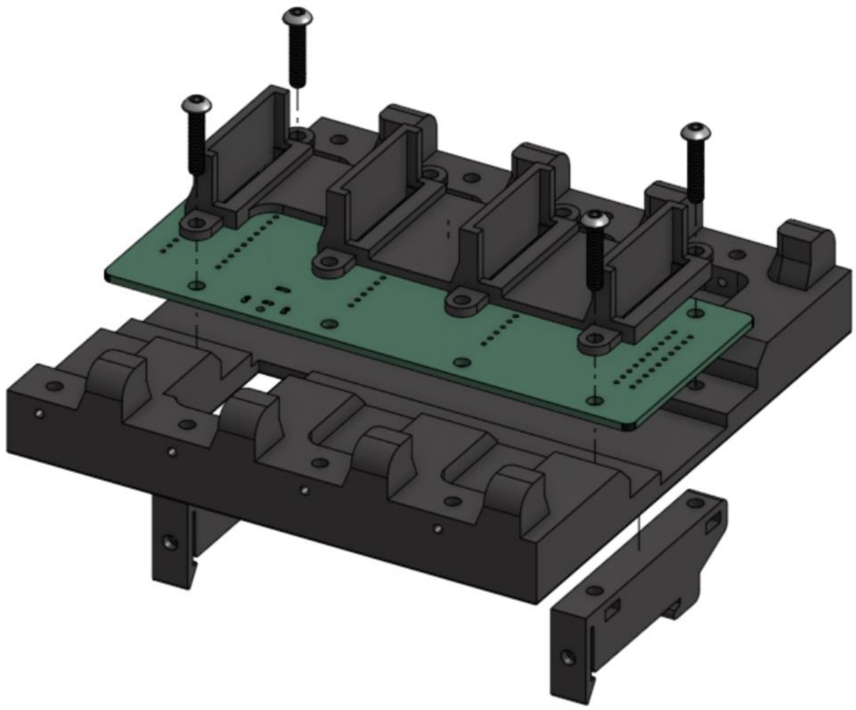


Figure 4.5: Loaf Assembly

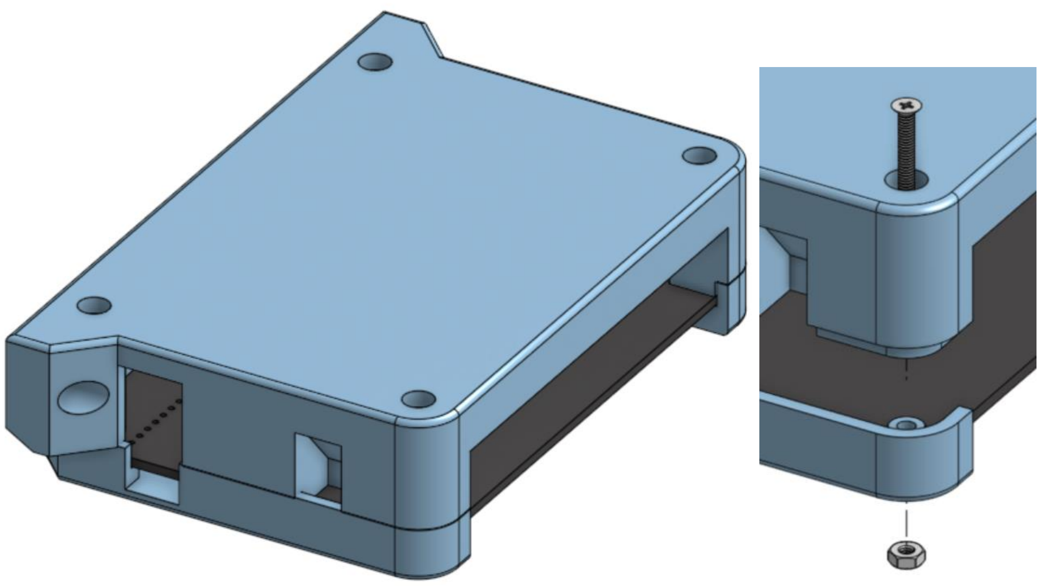


Figure 4.6: Slice Assembly (Left: Full Slice, Right: Screw Location)

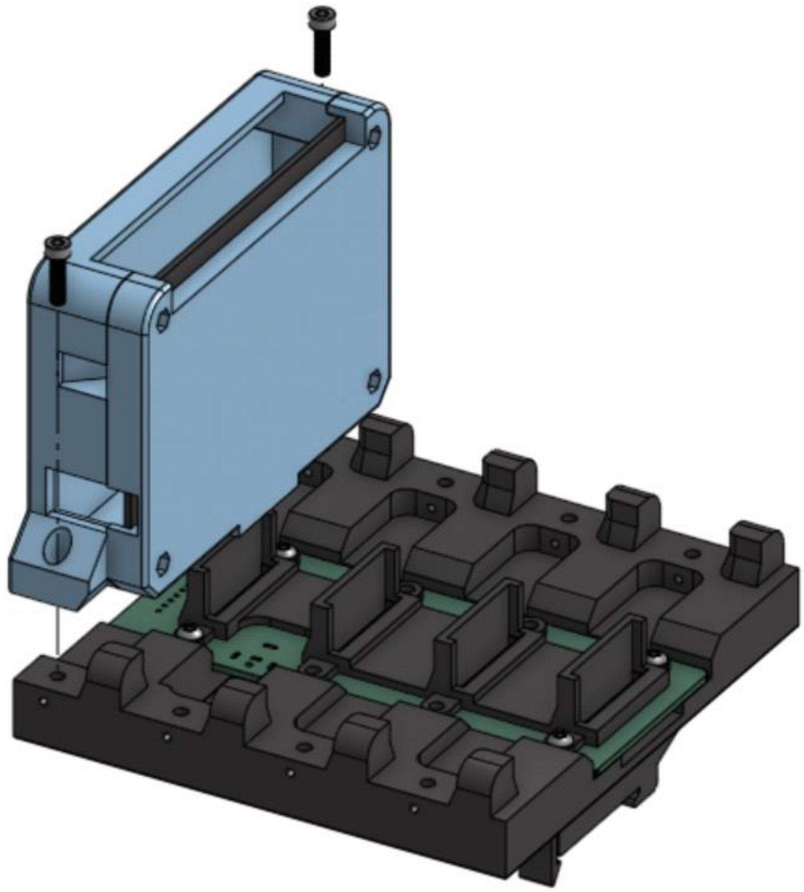


Figure 4.7: Slice Connection to Loaf

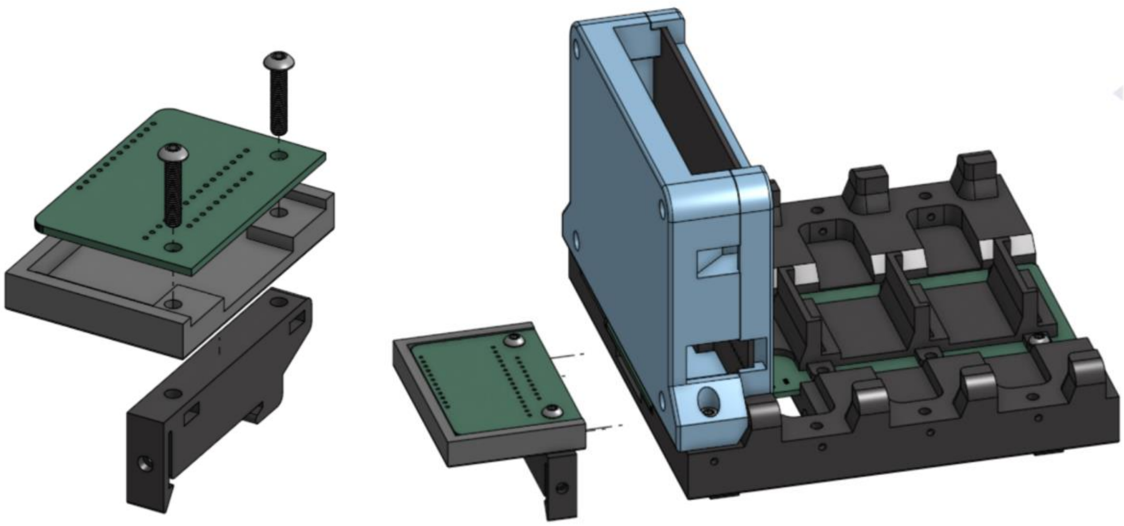


Figure 4.8: Loaf Controller (Left: Assembly, Right: Connection to Loaf)

4.4 Operation Instructions

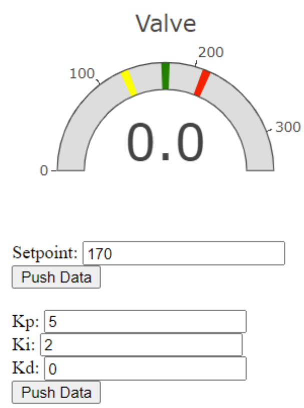
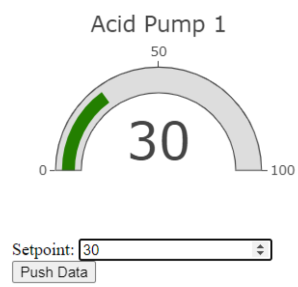
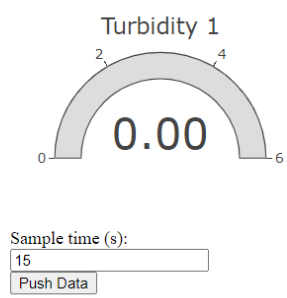
Setting up the firmware on each Slice is described in Appendix B of [5].

Setting up the Loaf Controller with Butter is outlined below and explained in detail in [16]:

1. Download the contents from the GitHub repository: https://github.com/FHafting/BREAD-Local-Software
2. Upload the contents under “Website Code on SD Card” to the SD card on the ESP32.
3. Upload the code under “Arduino Code” to the ESP32.
4. Turn on the system and connect to the “BREAD” Wi-Fi network. The password is 12345678.
5. Navigate to a Chrome browser and type “192.168.4.1” into the address bar. Wait 20-30 seconds for the web interface to display.

After opening the web interface, there are a variety of inputs, gauges, and graphs which are used to control Slices and display data. Their purposes are summarized in Table 4.4.

Table 4.4: Gauge Descriptions in Butter Software

Element	Image	Purpose
PID Controller		Assign the desired setpoint and PID tuning parameters for heating, cooling, or pH control.
Motor/Pump Controller		Assign the desired motor speed from -100% to 100% (full speed reverse and full speed forward, respectively)
Turbidity Controller		Assign the sampling interval of the turbidity pump

The top header of the webpage is used to start logging data, set the emergency stop, download and clear data, and show the connection status to the Loaf Controller (red: no communication, green: good communication). Figure 4.9 shows the webpage header.

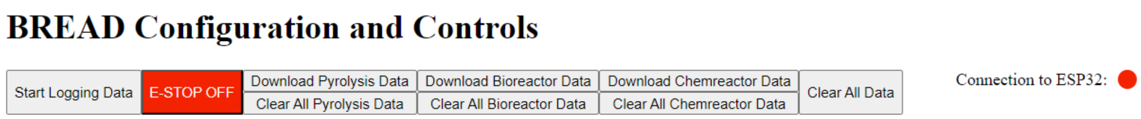


Figure 4.9: Butter Webpage Header

4.5 Validation and Characterization

Controlling the pH of large tanks in a continuous manner is common in swimming pool facilities. Products exist which balance the pH of the pool by injecting carbon dioxide (CO_2) such as Clearwater's pH Pure [17]. Similar control is used to balance the pH of seaweed growth tanks. During photosynthesis, CO_2 is consumed, and the pH rises which slows the reaction. Keeping the pH at 8.0 is optimal for growth and a controller is needed to facilitate CO_2 injection when needed to maintain this setpoint. Commercial controllers work by monitoring the pH and actuating a solenoid valve to inject CO_2 until the pH reaches a value below the setpoint. This feedback loop allows the controller to balance the pH. Since the growth tanks are also aerated, the CO_2 is injected into the air line and diffused into the water in the tank. A typical growth tank is shown in Figure 4.10.

Figure 4.10: Industrial Seaweed Growth Tank

To validate BREAD v2, it was compared to a JENCO pH/ORP Controller 3672, referred to as the "commercial controller", installed at Acadian Seaplants. Because this controller is proprietary and has no data logging capabilities, when it fails, the researchers are unable to diagnose the issue or determine when the failure occurred. After comparing the two controllers, BREAD will be considered an improvement if it can achieve a similar control precision and reliability because it already has data logging capabilities and can be easily repaired or diagnosed.

4.5.1 BREAD Controller

Within the BREAD framework, no Slice exists which monitors pH and can control a solenoid valve. A new Slice called “PHCL” was created for this purpose. To first validate the feasibility of using BREAD as a commercial pH controller alternative, an open-source amplifier board from DF Robot was integrated into each Slice. The Gravity pH Meter V2.0 is a simple amplifier circuit which scales the ±0.414V potential measured at the probe (equivalent to pH 14-0 respectively) to 5-0V which can be measured with an Arduino Nano analog-to-digital converter (ADC). This board can only be purchased within a $57 kit, so the cost of this Slice can be further reduced by integrating the amplifier onto the Slice. Solenoid actuation was achieved by reusing the relay circuit found in the RLHT Slice used for pyrolysis heating control [5]. The finished Slice is shown in Figure 4.11.

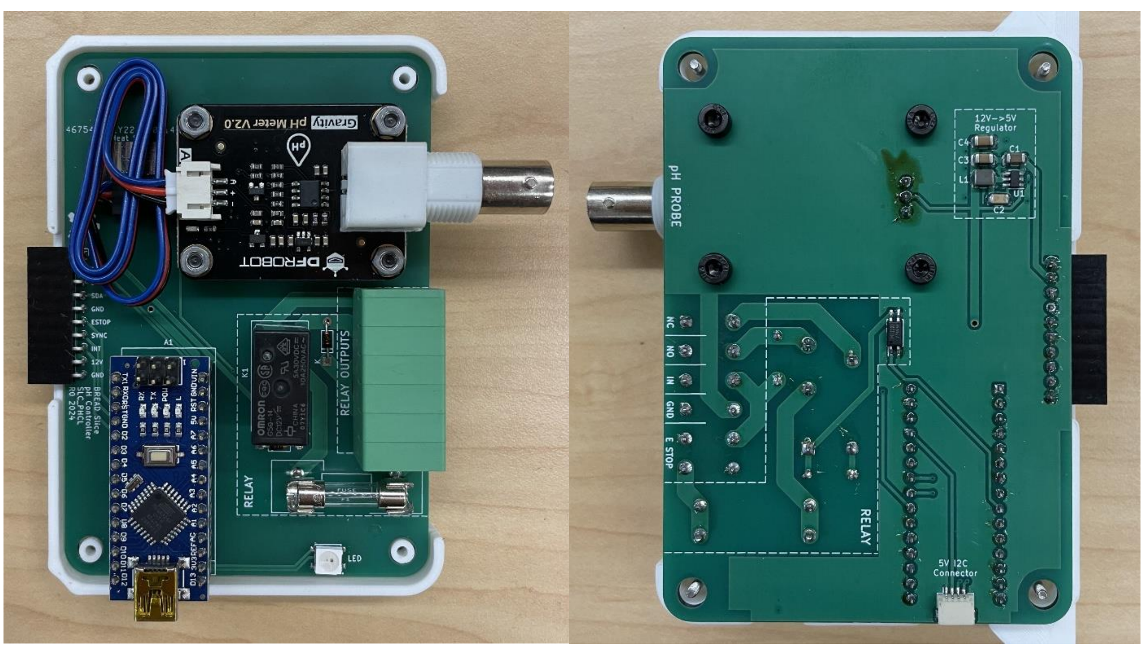


Figure 4.11: PHCL Slice (Left: Top, Right: Bottom)

To compare BREAD with a commercial pH controller, complete with a system to deliver CO_2, an electrical enclosure was made to contain the BREAD system, air lines, solenoid valves, and power supply (Figure 4.12 & Figure 4.13). Since each BREAD Loaf can hold 4 Slices, the enclosure was sized for 4 tanks. Sizing for 4 tanks was also more cost effective since most of the cost of the system was the enclosure and airline hardware.

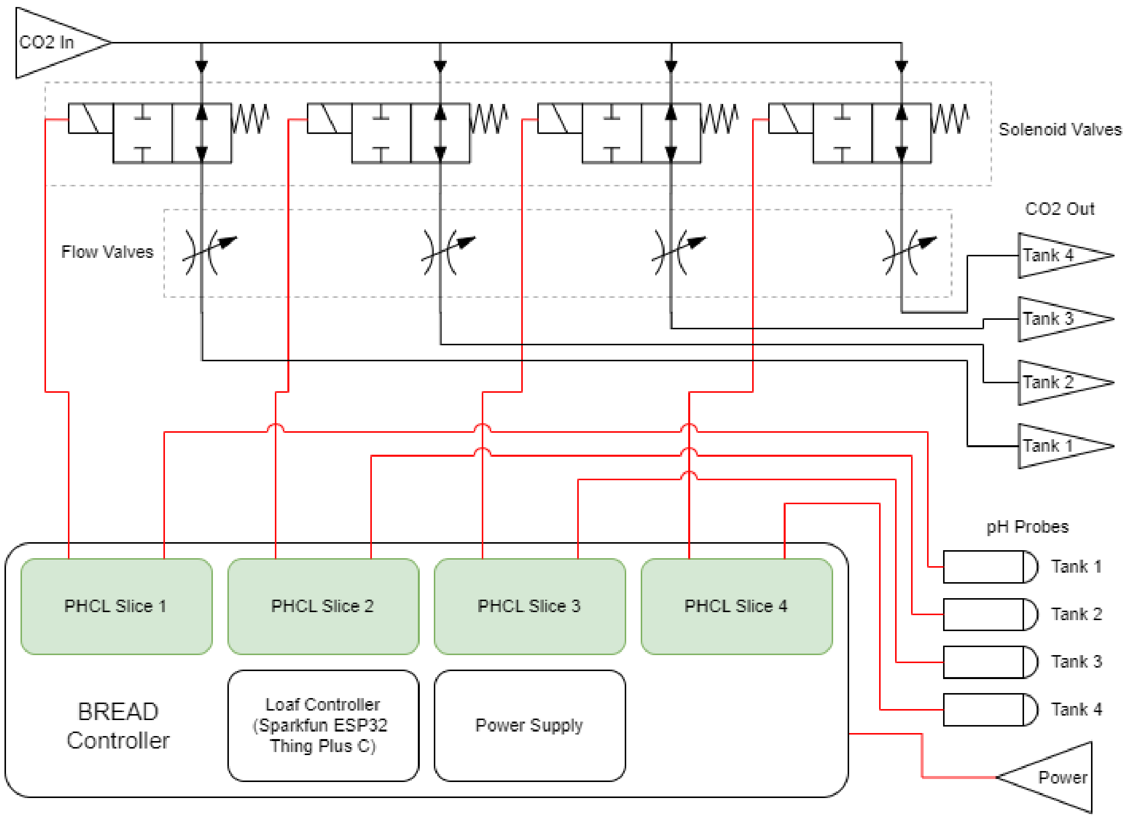


Figure 4.12: BREAD Controller Schematic (black: pneumatic connections, red: electrical connections)

Figure 4.13: BREAD Enclosure

Additionally, the software for BREAD, Butter [16], was modified to let users change the setpoint and PID tuning of each Slice, use single or two-point calibration, and show the current pH of each tank. Figure 4.14 shows the user interface.

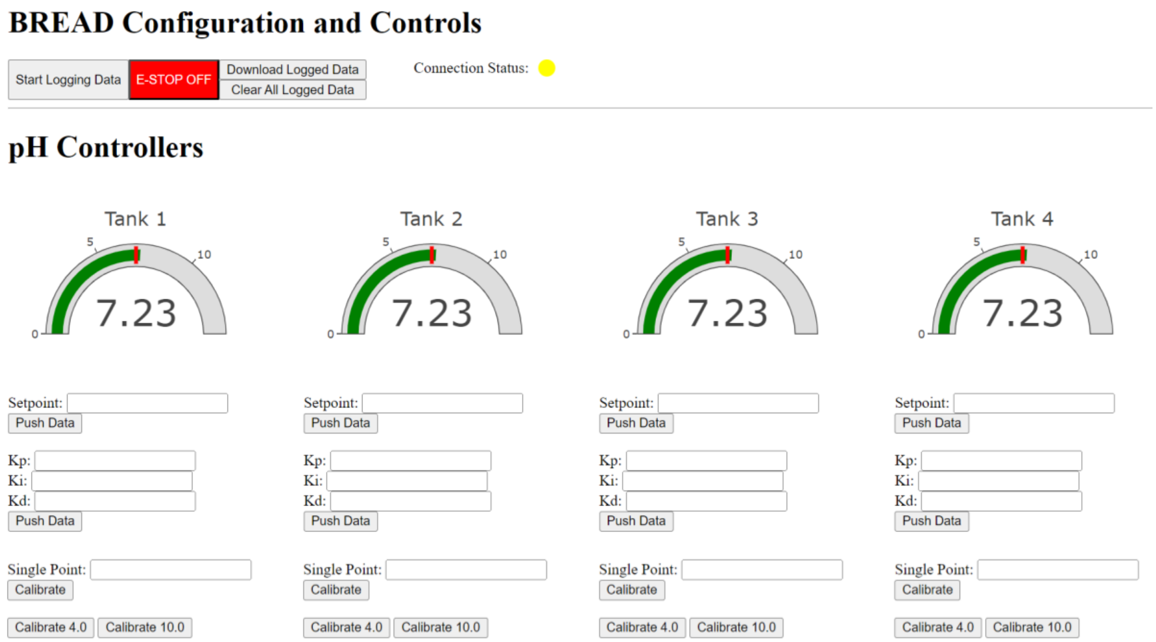


Figure 4.14: Modified Software for pH Control

4.5.2 Experimental Setup

To compare the BREAD system with a commercial pH controller, two growth tanks were prepared, each with an airline and pH probe. The JENCO pH/ORP Controller 3672 performs bang-bang control; injecting CO_2 into the tank's airline when the pH rises above the setpoint. However, since the controller has no data logging feature, the probe from Tank 1 of the BREAD controller was placed inside the tank to gather data. The BREAD controller was responsible for controlling the pH of tank 2 using a PID framework to adjust the pulse duty cycle of a solenoid valve to inject CO_2 into the airline. The specifications of each controller and setup are summarized in Table 4.5. Figure 4.15 shows the experimental setup with the BREAD controller and Figure 4.16 shows the commercial controller. Both tanks were left for 12 hours so each could reach a steady state.

Table 4.5: Controller Setup Parameters

Parameter	Tank 1	Tank 2
Controller	JENCO pH/ORP Controller 3672	BREAD Controller

Control scheme	Bang-bang	PID
Tunings	N/A	Kp = 2 Ki = 1 Kd = 0
Setpoint (pH)	8.0	8.0

Figure 4.15: Experimental Setup (Left: Tank 1, Middle: Tank 2, Right: BREAD Controller)

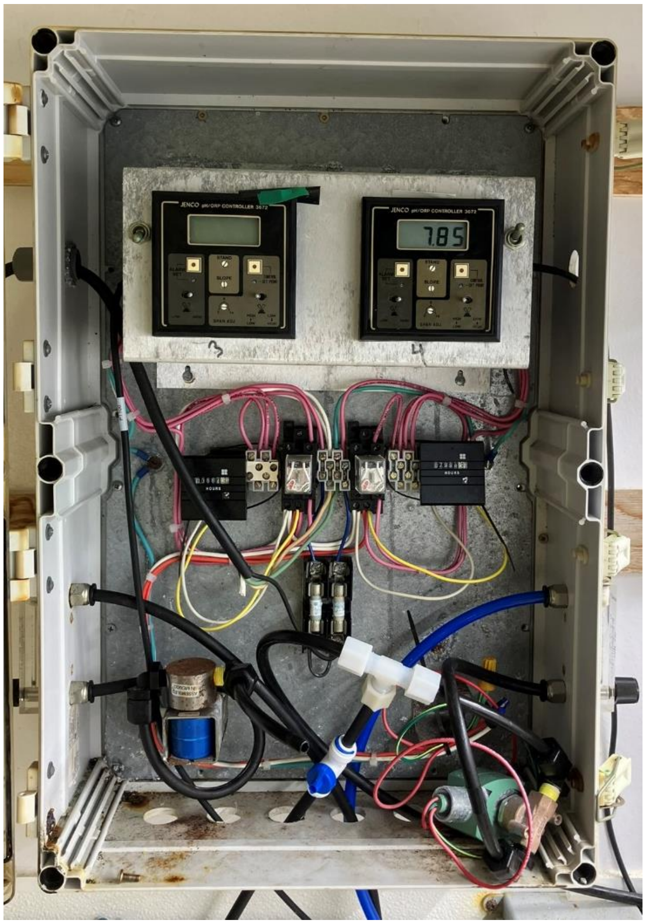


Figure 4.16: JENCO pH/ORP Controller 3672 "Commercial Controller"

Before starting the experiment, the BREAD controller was calibrated using two-point calibration of pH 7.0 and 10.0. Since both the BREAD and commercial controller do not have temperature compensation of the pH readings, each require calibration when moving to a new environment.

4.6 Results

The pH from Tank 1, with the commercial controller, and Tank 2, with the BREAD controller, are plotted over time and shown in Figure 4.17. The BREAD controller exceeded the precision of the commercial controller and reduced the amount of undershoot around the setpoint. Table 4.6 summarizes the performance characteristics of both controllers. Bang-bang control requires a deadband around the setpoint, so the solenoid is not continuously toggling. This also preserves the longevity of the valve.

However, this deadband also causes significant undershoot from the setpoint. Since the BREAD controller uses PID control to change the duty cycle of a set interval (5 seconds), there are only ever two toggles within that time frame. Thus, both the longevity and performance of the control system can be more easily predicted. Additionally, the BREAD controller tuning parameters were not optimized, so the precision of the controller could be further improved using tuning methods like the Ziegler–Nichols method [18].

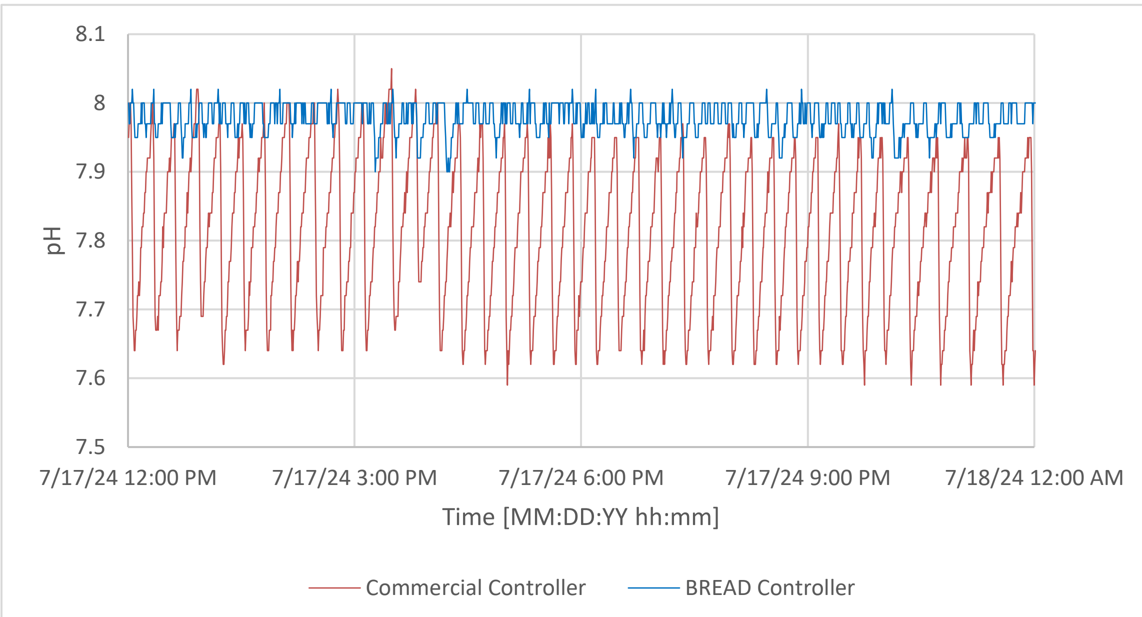


Figure 4.17: Controller Results

Table 4.6: Performance of Both Controllers

	Average	Deviation
Commercial Controller	7.81	0.095
BREAD Controller	7.98	0.019

The hardware to deliver CO_2 to the tanks is the same, so the cost of both controllers can be compared directly. Table 4.7 summarizes the costs of the BREAD controller when compared to the commercial controller. For a more in-depth breakdown, see Appendix C.

Not only is the BREAD controller more precise, but it is close to three times cheaper than the commercial controller.

Table 4.7: Cost Breakdown of Both Controllers

System	Part	Individual Cost (CAD)	Total Cost (CAD)
BREAD Controller	Slice	$86.78 (x4)	$454.80
	Loaf	$3.61	
	Loaf Controller	$66.54	
Commercial Controller	Commercial Controller	$330 (x4)	$1,320
Full System	BREAD Controller	$454.80	$861.66 (65% savings)
	Enclosure/Air tubing	$406.86	

4.7 Discussion and Limitations

The results are clear that the BREAD v2 controller is a significant improvement over the commercial controller both in cost and functionality. While these controllers were compared under steady state conditions, the startup conditions should also be addressed. The PID control worked well at steady state but was significantly slower than bang-bang control when trying to initially balance the pH to the setpoint. Fortunately, the BREAD v2 controller can be easily reprogrammed to use bang-bang control initially until the pH reaches the setpoint and switch to PID control to maintain a balance. This would be especially useful for larger commercial tanks. There are several other areas in which BREADv2 could be improved for this specific application for useability of the software:

- Data logging works reliably, however, the graphs on the web interface sometimes display data improperly or not at all. The reliability of the graphs could be improved.
- Settings are not saved when the system is rebooted. Having the settings saved on the SD card would mean that, when a user reboots the system, it remembers the previous configuration. This would also make the controller more reliable in the event of a power outage.

- CO_2 flow rate could not be adjusted during normal operation because the solenoid was being pulsed continuously. There needs to be a way to manually turn on the solenoid valve so the CO_2 flow rate can be adjusted.

Since BREAD is easily customizable, these changes can be added in the future, and the software continuously modified to ensure it best meets the needs of the user. Also, due to the open-source nature of BREAD, users with some coding knowledge can choose to add their own changes to suit their needs. The power of BREAD as a controller lies in its documentation [15]. Since users have free access to the continuously updated repository of Slice designs, they can diagnose hardware issues and purchase replacement parts without needing to buy a new system. Open-source hardware has made it easier than ever to replace proprietary products with powerful, easily customizable, and reparable alternatives that are both cheaper and provide more functionality.

4.8 Conclusion

BREAD continues to be a formidable alternative to commercial SCADA systems with both a reduction in costs and improvement in functionality. The new form factor of BREAD v2 enables it to be installed in an electrical enclosure which widens its application possibilities. This study demonstrated how BREAD can be integrated into a full pH control system that can be easily set up and used.

Appendix C: BREAD Controller Cost Breakdown

Table 4.8: Enclosure/Air Tubing Cost

Part	#	Cost	Total	URL
Electrical box	1	$95.99	$95.99	https://www.amazon.ca/dp/B0BN3MWN6K/?th=1
Power supply	1	$25.84	$25.84	https://www.digikey.ca/en/products/detail/ideal-power-ltd/56YSD30S-1202500/14306708
Solenoid valve	4	$12.59	$50.36	https://www.amazon.ca/GENEDEY-Solenoid-Normally-Closed-Switch/dp/B08K8QRRBJ/
Flow meter	4	$32.89	$131.56	https://www.amazon.ca/Control-Flowmeter-Acrylic-Adjustable-1-10LPM/dp/B0C79414W5/
DIN Rail	1	$16.99	$16.99	https://www.amazon.ca/DKARDU-Slotted-Aluminum-Mounting-Terminal/dp/B09Z6BS51D/

Power cable	1	$11.99	$11.99	https://www.amazon.ca/AmazonBasics-Extension-Cord-Feet-Black/dp/B075BMVZ2N/
Tubing + Tube Fittings	1	$16.19	$16.19	https://www.amazon.ca/GREDIA-Connect-Purifiers-Fittings-Reverse/dp/B08NVHRW6B/
Screw tube fittings	1	$15.97	$15.97	https://www.amazon.ca/dp/B08RS4VCSV/
Bulkhead fittings	1	$9.99	$9.99	https://www.amazon.ca/ZAOJIAO-Bulkhead-Connector-Connect-Reverse/dp/B06Y29M74D/
BCN jumper cable	2	$15.99	$31.98	https://www.amazon.ca/dp/B09D3JFL4S?psc=1&ref=ppx_yo2ov_dt_b_product_details
		Total	$406.86	

Table 4.9: PHCL Slice Cost

Part	#	Cost	Total	URL
pH reader	1	$57.28	$57.28	https://www.mouser.ca/ProductDetail/DFRobot/SEN0161-V2?qs=qSfuJ%252Bfl%2Fd6eAiYFumrJvQ%3D%3D
22uF	3	$0.51	$1.53	https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL31A226KAHNNNE/3888705
100nF	1	$0.16	$0.16	https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL31B104KBCNNNC/3886675
4.7uH	1	$2.32	$2.32	https://www.digikey.ca/en/products/detail/tdk-corporation/BCL322520RT-4R7M-D/15849704
AP63205	1	$1.31	$1.31	https://www.digikey.ca/en/products/detail/diodes-incorporated/AP63205WU-7/9858424
TCMT1100	2	$0.92	$1.84	https://www.digikey.ca/en/products/detail/vishay-semiconductor-opto-division/TCMT1100/1738630
WS2812B_5050	1	$0.91	$0.91	https://www.digikey.ca/en/products/detail/sparkfun-electronics/COM-16347/11630204
General_Purpose_Diode	2	$0.14	$0.28	https://www.digikey.ca/en/products/detail/onsemi/1N914B/978757
Fuse clip	2	$0.10	$0.20	https://www.digikey.ca/en/products/detail/bel-fuse-inc/fc-203-bright-tin/2650843
Fuse 5x15mm 10A	1	$0.71	$0.71	https://www.digikey.ca/en/products/detail/littelfuse-inc/0225010-MXUP/777789
Conn_01x10_Female	1	$1.09	$1.09	https://www.digikey.ca/en/products/detail/sullins-connector-solutions/PPTC101LGBN-RC/775904
I2C_01x04	1	$0.85	$0.85	https://www.digikey.ca/en/products/detail/jst-sales-america-inc/SM04B-SRSS-TB/926710

Term_PressFit_ Phoenix_Contac t 5mm	1	$4.70	$4.70	https://www.digikey.ca/en/products/detail/phoenix-contact/1991011/2527265
Relay G5Q-1	2	$2.27	$4.54	https://www.digikey.ca/en/products/detail/omron-electronics-inc-emc-div/G5Q-14-DC12/355243
Arduino nano	1	$9.06	$9.06	https://www.amazon.ca/ELEGOO-Arduino-ATmega328P-Without-Compatible/dp/B0713XK923/
		Total	$86.78	

Table 4.10: Loaf Controller Cost

Part	**#**	**Cost**	**Total**	**URL**
Sparkfun Thing Plus C	1	$24.95	$24.95	https://www.sparkfun.com/products/20168
32GB SD Card	1	$34.98	$34.98	https://www.amazon.ca/Gigastone-32GB-U1-C10-Nintendo/dp/B07N73LB4T/
Conn_01x10_Fe male	1	$1.09	$1.09	https://www.digikey.ca/en/products/detail/sullins-connector-solutions/PPTC101LGBN-RC/775904
22uF	3	$0.51	$1.53	https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL31A226KAHNNNE/3888705
100nF	1	$0.16	$0.16	https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL31B104KBCNNNC/3886675
4.7uH	1	$2.32	$2.32	https://www.digikey.ca/en/products/detail/tdk-corporation/BCL322520RT-4R7M-D/15849704
AP63205	1	$1.31	$1.31	https://www.digikey.ca/en/products/detail/diodes-incorporated/AP63205WU-7/9858424
Resistor 2.2k	2	$0.10	$0.20	https://www.digikey.ca/en/products/detail/yageo/RC0603JR-132K2L/14286397
		Total	$66.54	

Table 4.11: Loaf Backplane Cost

Part	**#**	**Cost**	**Total**	**URL**
Conn_01x10_M ale	4	$0.25	$1.00	https://www.digikey.ca/en/products/detail/amphenol-cs-fci/10129378-910002BLF/7915958
1x10 RA Male Header 2.54mm Pitch	1	$0.23	$0.23	https://www.digikey.ca/en/products/detail/amphenol-cs-fci/10129379-910002BLF/7916010
PinSocket_1x10 _P2.54mm_Hori zontal	1	$0.77	$0.77	https://www.digikey.ca/en/products/detail/sullins-connector-solutions/PPTC101LGBN-RC/775904

BarrelJack_CUI_PJ-063AH_Horizontal	1	$1.61	$1.61	https://www.digikey.ca/en/products/detail/cui-devices/pj-063ah/2161208
		Total	$3.61	

Table 4.12: Slice Case Hardware Cost

Part	#	Cost	Total	URL
M2x25mm screws (x50)	2	$9.31	$18.62	https://www.mcmaster.com/92010A111/
M2 nuts (x100)	1	$4.00	$4.00	https://www.mcmaster.com/90592A075/
M3x12mm screws (x100)	1	$6.00	$6.00	https://www.mcmaster.com/91292A114/
M3 nuts (x100)	1	$2.62	$2.62	https://www.mcmaster.com/90592A085/
m3x15 bolts	1	$6.28	$6.28	https://www.mcmaster.com/92095A119/
		Total	$37.52	

4.9 References

[1] S. Oberloier, N. G. Whisman, F. Hafting, and J. M. Pearce, "Open source framework for a Broadly Expandable and Reconfigurable data acquisition and automation device (BREAD)," *HardwareX*, vol. 15, p. e00467, Sep. 2023, doi: 10.1016/j.ohx.2023.e00467.

[2] "Opto22 - groov EPIC System." Accessed: May 01, 2024. [Online]. Available: https://www.opto22.com/products/groov-epic-system

[3] "CompactRIO Systems (cRIO)." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en/shop/compactrio.html

[4] F. K. Hafting, S. Yeung, H. Al-Aribe, E. Michels, and J. M. Pearce, "Parametric Low-Cost Open Source Bioreactor Designed for Distributed Manufacturing," In Review.

[5] F. K. Hafting *et al.*, "Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics," *Electronics*, vol. 12, no. 24, Art. no. 24, Jan. 2023, doi: 10.3390/electronics12244893.

[6] J. M. Pearce, "Economic savings for scientific free and open source technology: A review," *HardwareX*, vol. 8, p. e00139, Oct. 2020, doi: 10.1016/j.ohx.2020.e00139.

[7] "Arduino - Home." Accessed: Jul. 23, 2024. [Online]. Available: https://www.arduino.cc/

[8] "PCB Prototype & PCB Fabrication Manufacturer - JLCPCB." Accessed: Jul. 25, 2024. [Online]. Available: https://jlcpcb.com/?from=VGPCBA&gad_source=1&gclid=CjwKCAjw74e1BhBnEiwAbqOAjPinT95y3kicmEcGoxhGIT5N5ecGG8C3Ay0yHyRZPq2gG9kaOIsczhoC5jIQAvD_BwE

[9] "NI cRIO-9030." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/crio-9030.html

[10] "NI 9210." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/ni-9210.html
[11] "NI 9482." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/ni-9482.html
[12] "NI 9470." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/ni-9470.html
[13] "NI 9203." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/ni-9203.html
[14] "NI 9219." Accessed: Jul. 24, 2024. [Online]. Available: https://www.ni.com/en-ca/shop/model/ni-9219.html
[15] S. Oberloier, N. Whisman, F. Hafting, and J. M. Pearce, "BREAD," Apr. 2022, doi: 10.17605/OSF.IO/U2H4G.
[16] X. Chan, F. Hafting, and J. M. Pearce, "Control software for the open-source Broadly Reconfigurable and Expandable Automation Device (BREAD) System," *J. Open Source Softw.*, Under review.
[17] "pH PURE™ CO2 Injection for pH Control - Swimming Pools," Clearwater Pool Systems. Accessed: Jul. 21, 2024. [Online]. Available: https://www.clearwaterpoolsystems.com/ph-pure/
[18] K. J. Åström and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers, Second Edition*. Princeton University Press, 2021.

Chapter 5

5 Conclusions and Future Work

After comparing the performance of BREAD with a variety of commercial control systems, it is clear that BREAD can provide more functionality at a reduced cost. In many cases, the high accuracy promised by these commercial control systems was less desirable than additional functionality like data logging. For the pyrolysis reactor, the BREAD system cost approximately $580 CAD and could still produce the same yield and product quality as a $6000 commercial equivalent. BREAD also contributed significant cost savings to the open-source bioreactor while still providing precise control of heating (±1°C) and pH (±0.035). BREADv2 can be integrated into electrical enclosures and, potentially, sold as a product. As explored in Chapter 4, BREADv2 can be fully integrated into an airline-based pH controller for seaweed growth tanks. With this controller, scientists can save 60% in cost when compared to a JENCO pH/ORP Controller 3672, have 80% more control precision, and data logging capabilities.

The specifications and limitations of each Slice covered in this work and within the greater BREAD framework can be found on the GitHub repository linked below: https://github.com/uwo-fast/BREAD

Continuing to integrate existing open-source designs with the BREAD framework will further expand its functionality. Additionally, pushing the limits of the BREAD hardware into new research topics will lead to more Slice development and a more robust design. A few future improvements could be made to BREADv2 as outlined below:

- Individual Slices can technically operate as independent units for simple applications like temperature logging. However, single Slice mode was investigated very little in this work.
- Multi-Loaf mode could also be explored further in an industrial setting where multiple controllers are monitored by a central computer. This scenario would also require improvements to the software.

- Add the ability for users to program tasks in the software like heating cycles or drainage times for pumps.
- Explore existing open-source software like ThingsBoard or the Robot Operating System (ROS).
- Explore how sharing information between Slices could be implemented. CAN could be a viable alternative to I^2C communication since it has strong signal integrity and provides full duplex communication.
- Investigate potential for CSA approval and electromagnetic compatibility if commercial opportunities are explored with BREAD.

With the current state of BREAD, improvements to the software will have the greatest impact on its usability and functionality. The potential of BREAD as a competitive and open-source alternative to commercial, proprietary controllers and as a fully integrated product was explored in this work. BREAD will only continue to grow as more researchers add to the repository of Slice designs, improve the software, and explore new applications. As with many open-source projects, the community of BREAD users will be the force that drives its innovation.

Curriculum Vitae

Name:	Finn Hafting
Post-secondary Education and Degrees:	University of Western Ontario London, Ontario, Canada 2019-2023 B.E.Sc.
	The University of Western Ontario London, Ontario, Canada 2023-2024 M.E.Sc.
Honours and Awards:	Loran Scholars Award 2019-2023
	The University of Western Ontario Gold Medal 2023
	Queen Elizabeth II Graduate Scholarship in Science and Technology 2023-2024

Publications:

Oberloier, S., Whisman, N. G., Hafting, F., & Pearce, J. M. (2023). Open source framework for a Broadly Expandable and Reconfigurable data acquisition and automation device (BREAD). HardwareX, 15, e00467.

Hafting, F. K., Kulas, D., Michels, E., Chipkar, S., Wisniewski, S., Shonnard, D., & Pearce, J. M. (2023). Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics. Electronics, 12(24), 4893.

McDonald, M. T., Hayibo, K. S., Hafting, F., & Pearce, J. M. (2022). Economics of Open-Source Solar Photovoltaic Powered Cryptocurrency Mining. Available at SSRN 4205879.