12-5-2023

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

Finn K. Hafting
*Western University*

Daniel G. Kulas
*Michigan Technological University*, dgkulas@mtu.edu

Etienne Michels
*College of Engineering*, ebmichel@mtu.edu

Sarvada Chipkar
*Michigan Technological University*, shchipka@mtu.edu

Stefan Wisniewski
*College of Engineering*, sgwisnie@mtu.edu

*See next page for additional authors*

## Recommended Citation

Hafting, F., Kulas, D. G., Michels, E., Chipkar, S., Wisniewski, S., Shonnard, D., & Pearce, J. (2023). Modular Open-Source Design of Pyrolysis Reactor Monitoring and Control Electronics. *Electronics (Switzerland), 12*(24). http://doi.org/10.3390/electronics12244893
Retrieved from: https://digitalcommons.mtu.edu/michigantech-p2/462

Follow this and additional works at: https://digitalcommons.mtu.edu/michigantech-p2

Part of the Biomedical Engineering and Bioengineering Commons, and the Electrical and Computer Engineering Commons

## Authors

Finn K. Hafting, Daniel G. Kulas, Etienne Michels, Sarvada Chipkar, Stefan Wisniewski, David Shonnard, and Joshua M. Pearce

*electronics*

*Article*

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

**Finn K. Hafting [1], Daniel Kulas [2], Etienne Michels [3], Sarvada Chipkar [2], Stefan Wisniewski [2], David Shonnard [2] and Joshua M. Pearce [1,4,*]**

[1] Department of Electrical & Computer Engineering, Western University, 1151 Richmond St. N., London, ON N6A 3K7, Canada; fhafting@uwo.ca
[2] Department of Chemical Engineering, Michigan Technological University, 1400 Townsend Dr., Houghton, MI 49931, USA; dgkulas@mtu.edu (D.K.); sgwisnie@mtu.edu (S.W.); drshonna@mtu.edu (D.S.)
[3] Department of Biomedical Engineering, Michigan Technological University, 1400 Townsend Dr., Houghton, MI 49931, USA; ebmichel@mtu.edu
[4] Ivey School of Business, Western University, 1151 Richmond St. N., London, ON N6A 3K7, Canada
* Correspondence: joshua.pearce@uwo.ca

**Abstract:** 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. 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.

**Keywords:** open hardware; open-source hardware; open-source electronics; automation; Arduino; automation; pyrolysis; data acquisition; controls; monitoring

check for updates

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

Academic Editors: Isaías González Pérez and Antonio José Calderón Godoy

Received: 21 October 2023
Revised: 30 November 2023
Accepted: 1 December 2023
Published: 5 December 2023

## 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-graded 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 recycling technology 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. Materials and Methods

### *2.1. Commercial Methods of Pyrolysis Reactor Control*

The pilot pyrolysis reactor system (see Figure 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].

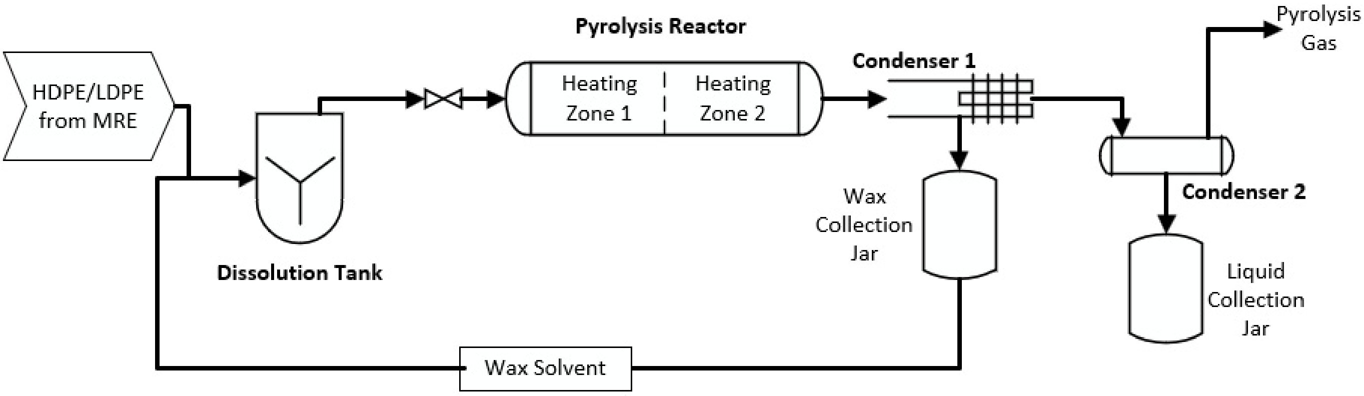


**Figure 1.** Process flow diagram of liquid-fed plastic pyrolysis. Waste polyethylene plastic enters the dissolution tank and is broken down in the primary and secondary pyrolysis reactors. A dual condenser system connects the wax, liquid, and gas product. MRE means meals-ready-to-eat bags.

### 2.2. 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 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].

### 2.3. 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). 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 (SLC_DCMT) and the relay heater Slice (SLC_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 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.

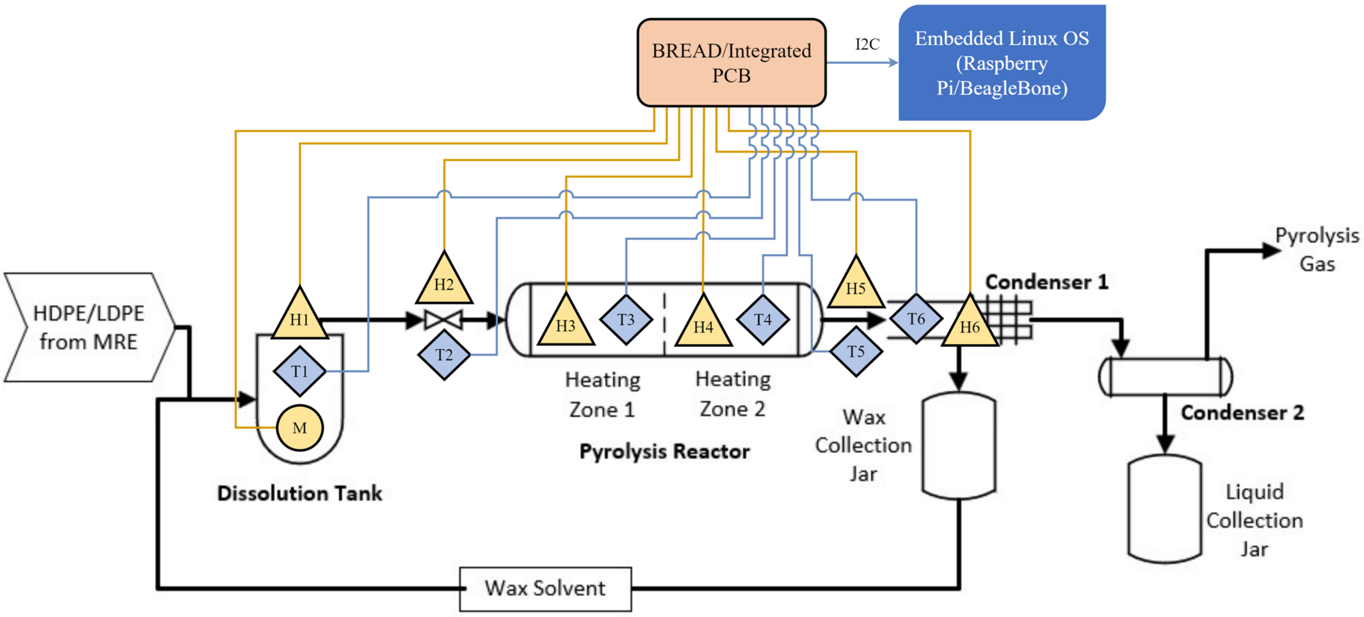


**Figure 2.** Pyrolysis system diagram (M: motor, T: thermocouple, H: heater). The arrows indicate the material flow paths.

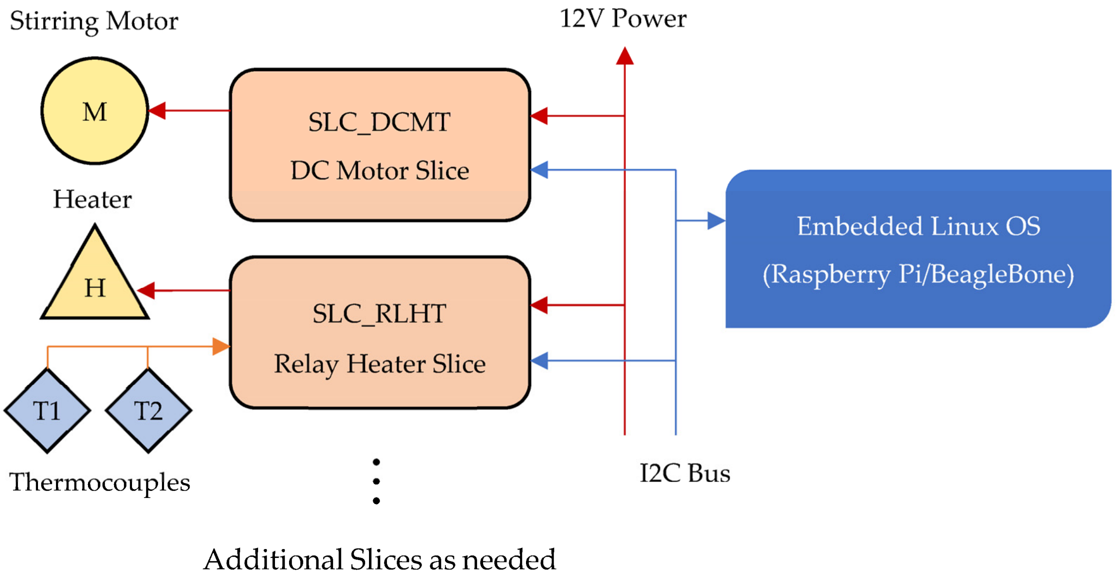


**Figure 3.** System connection diagram. Arrows indicate the direction of information (blue) and power (red).

Each relay heater Slice can control a single heater (maximum of 10 A @ 250VAC or 5 A @ 30VDC) and monitor 2 k-type thermocouples and 2 thermistors (Figure 4). The DC motor Slice can control the speed and direction of two DC motors (maximum of 3 A @ 12 V) (Figure 5). The 10 pin connector linked each Slice to the Loaf backplane, was provided 12 V of power, and enabled I2C communication to the Raspberry Pi. Connection of the Loaf to the Raspberry Pi was undertaken via the I2C port (Figures 6 and 7). For more technical details on the BREAD framework, one should review the original BREAD publication [36].

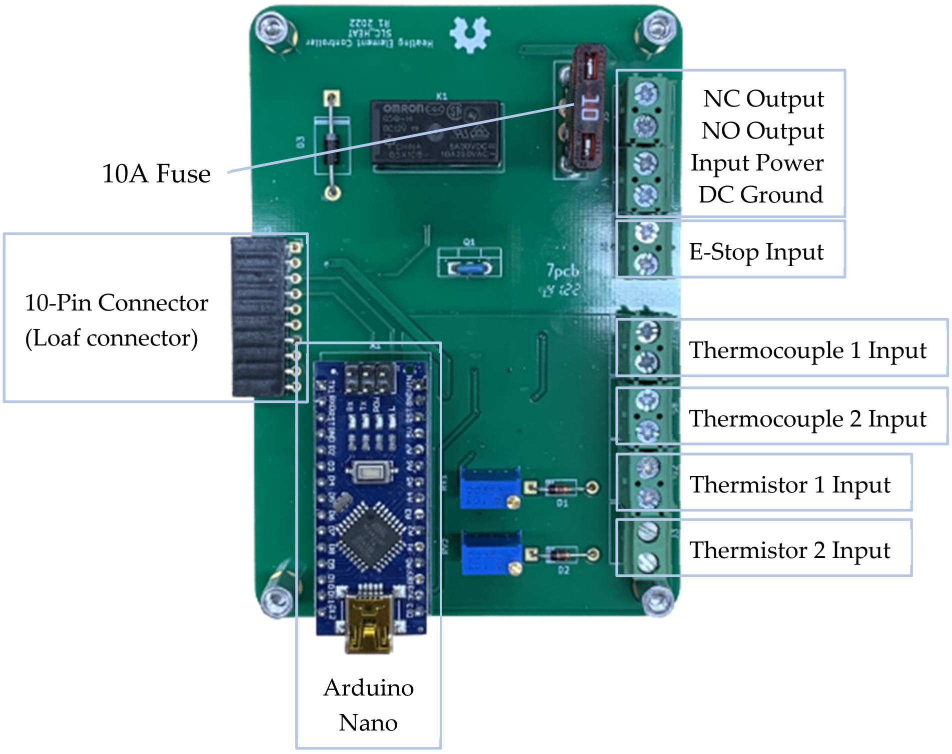


**Figure 4.** SLC_RLHT connection diagram (when using the relay as a switch, only use input power and the normally closed (NC) output or the normally open (NO) output).

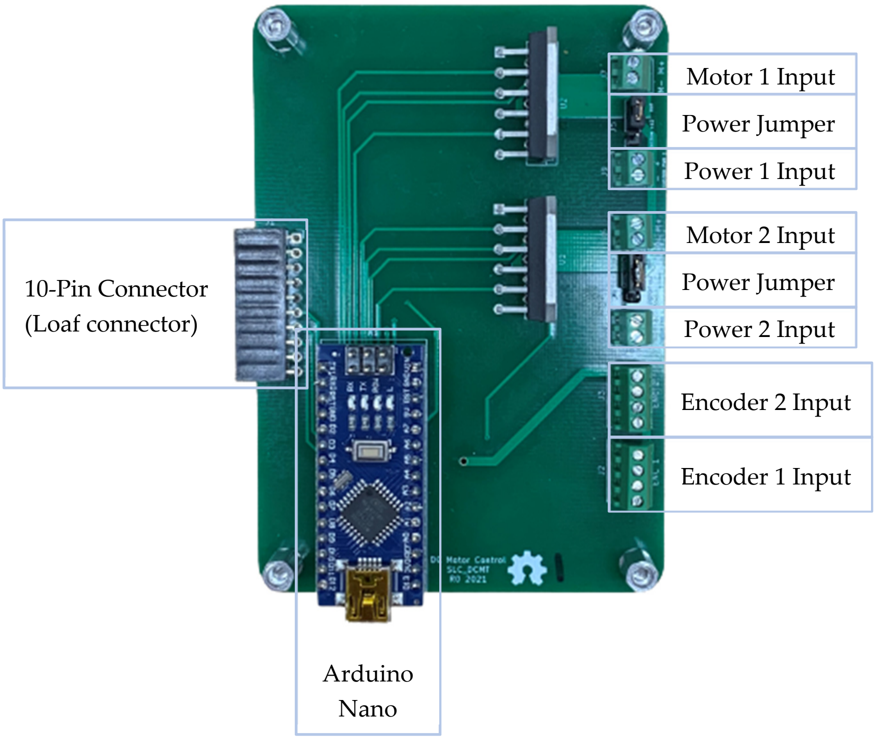


**Figure 5.** SLC_DCMT connection diagram.

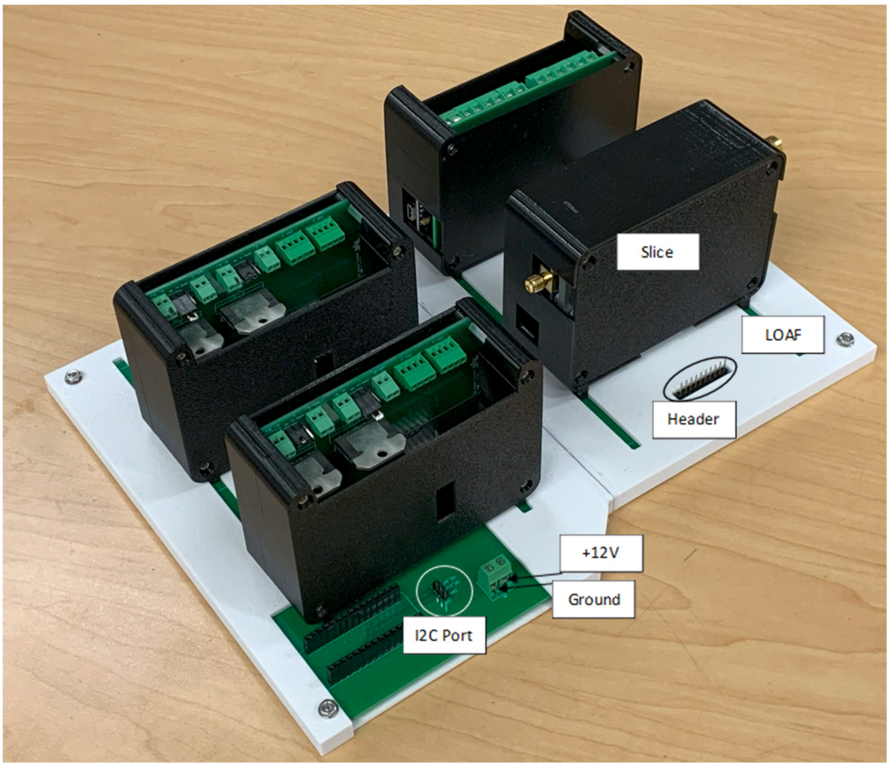
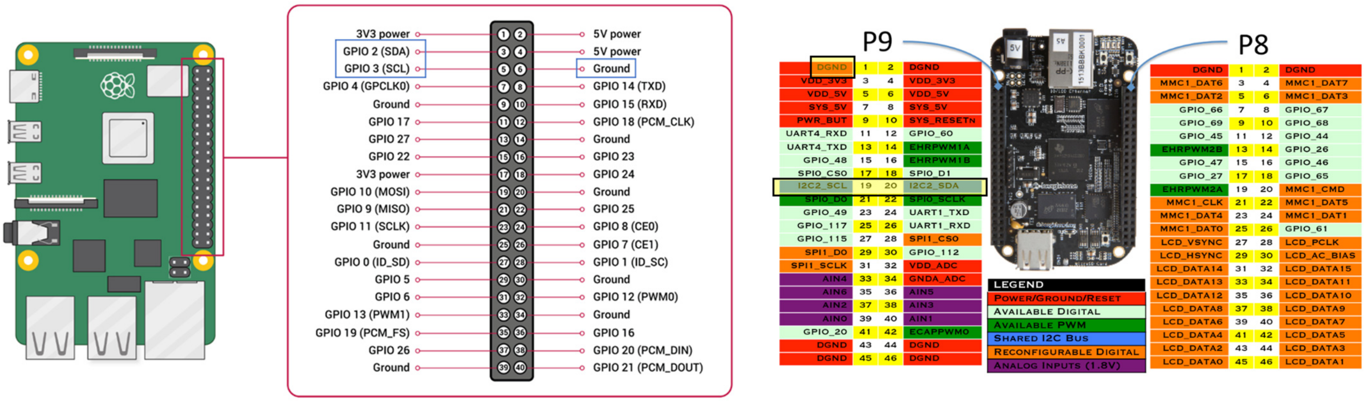


**Figure 6.** Loaf connection diagram.



|  |  |  |  |
|---|---|---|---|
| 3V3 power | 1 | 2 | 5V power |
| GPIO 2 (SDA) | 3 | 4 | 5V power |
| GPIO 3 (SCL) | 5 | 6 | Ground |
| GPIO 4 (GPCLK0) | 7 | 8 | GPIO 14 (TXD) |
| Ground | 9 | 10 | GPIO 15 (RXD) |
| GPIO 17 | 11 | 12 | GPIO 18 (PCM_CLK) |
| GPIO 27 | 13 | 14 | Ground |
| GPIO 22 | 15 | 16 | GPIO 23 |
| 3V3 power | 17 | 18 | GPIO 24 |
| GPIO 10 (MOSI) | 19 | 20 | Ground |
| GPIO 9 (MISO) | 21 | 22 | GPIO 25 |
| GPIO 11 (SCLK) | 23 | 24 | GPIO 8 (CE0) |
| Ground | 25 | 26 | GPIO 7 (CE1) |
| GPIO 0 (ID_SD) | 27 | 28 | GPIO 1 (ID_SC) |
| GPIO 5 | 29 | 30 | Ground |
| GPIO 6 | 31 | 32 | GPIO 12 (PWM0) |
| GPIO 13 (PWM1) | 33 | 34 | Ground |
| GPIO 19 (PCM_FS) | 35 | 36 | GPIO 16 |
| GPIO 26 | 37 | 38 | GPIO 20 (PCM_DIN) |
| Ground | 39 | 40 | GPIO 21 (PCM_DOUT) |

P9

|  |  |  |  |
|---|---|---|---|
| DGND | 1 | 2 | DGND |
| VDD_3V3 | 3 | 4 | VDD_3V3 |
| VDD_5V | 5 | 6 | VDD_5V |
| SYS_5V | 7 | 8 | SYS_5V |
| PWR_BUT | 9 | 10 | SYS_RESETN |
| UART4_RXD | 11 | 12 | GPIO_60 |
| UART4_TXD | 13 | 14 | EHRPWM1A |
| GPIO_48 | 15 | 16 | EHRPWM1B |
| SPI0_CS0 | 17 | 18 | SPI0_D1 |
| I2C2_SCL | 19 | 20 | I2C2_SDA |
| SPI0_D0 | 21 | 22 | SPI0_SCLK |
| GPIO_49 | 23 | 24 | UART1_TXD |
| GPIO_117 | 25 | 26 | UART1_RXD |
| GPIO_115 | 27 | 28 | SPI1_CS0 |
| SPI1_D0 | 29 | 30 | GPIO_112 |
| SPI1_SCLK | 31 | 32 | VDD_ADC |
| AIN4 | 33 | 34 | GNDA_ADC |
| AIN6 | 35 | 36 | AIN5 |
| AIN2 | 37 | 38 | AIN3 |
| AIN0 | 39 | 40 | AIN1 |
| GPIO_20 | 41 | 42 | ECAPPWM0 |
| DGND | 43 | 44 | DGND |
| DGND | 45 | 46 | DGND |

P8

|  |  |  |  |
|---|---|---|---|
| DGND | 1 | 2 | DGND |
| MMC1_DAT6 | 3 | 4 | MMC1_DAT7 |
| MMC1_DAT2 | 5 | 6 | MMC1_DAT3 |
| GPIO_66 | 7 | 8 | GPIO_67 |
| GPIO_69 | 9 | 10 | GPIO_68 |
| GPIO_45 | 11 | 12 | GPIO_44 |
| EHRPWM2B | 13 | 14 | GPIO_26 |
| GPIO_47 | 15 | 16 | GPIO_46 |
| GPIO_27 | 17 | 18 | GPIO_65 |
| EHRPWM2A | 19 | 20 | MMC1_CMD |
| MMC1_CLK | 21 | 22 | MMC1_DAT5 |
| MMC1_DAT4 | 23 | 24 | MMC1_DAT1 |
| MMC1_DAT0 | 25 | 26 | GPIO_61 |
| LCD_VSYNC | 27 | 28 | LCD_PCLK |
| LCD_HSYNC | 29 | 30 | LCD_AC_BIAS |
| LCD_DATA14 | 31 | 32 | LCD_DATA15 |
| LCD_DATA13 | 33 | 34 | LCD_DATA11 |
| LCD_DATA12 | 35 | 36 | LCD_DATA10 |
| LCD_DATA8 | 37 | 38 | LCD_DATA9 |
| LCD_DATA6 | 39 | 40 | LCD_DATA7 |
| LCD_DATA4 | 41 | 42 | LCD_DATA5 |
| LCD_DATA2 | 43 | 44 | LCD_DATA3 |
| LCD_DATA0 | 45 | 46 | LCD_DATA1 |

LEGEND
🟥 Power/Ground/Reset
🟩 Available Digital
🟩 Available PWM
🟦 Shared I2C Bus
🟧 Reconfigurable Digital
🟪 Analog Inputs (1.8V)

(**a**) (**b**)

**Figure 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 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 A4. The procedure for setting up a new Slice is detailed in Appendix B.

**Table 1.** Design file links.

| Files | URL (Accessed 30 November 2023) |
|---|---|
| BREAD SLC_RLHT | https://osf.io/pf6gy/ |
| BREAD SLC_DCMT | https://osf.io/6aw9m/ |
| Integrated pyrolysis board | https://osf.io/3ugbn/ |

### 2.4. Integrated Single-Board Design

To reduce costs and design more permanent control hardware, all components used with the BREAD system were integrated onto a single PCB (Figure 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 SLC_RLHT and 1 SLC_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.

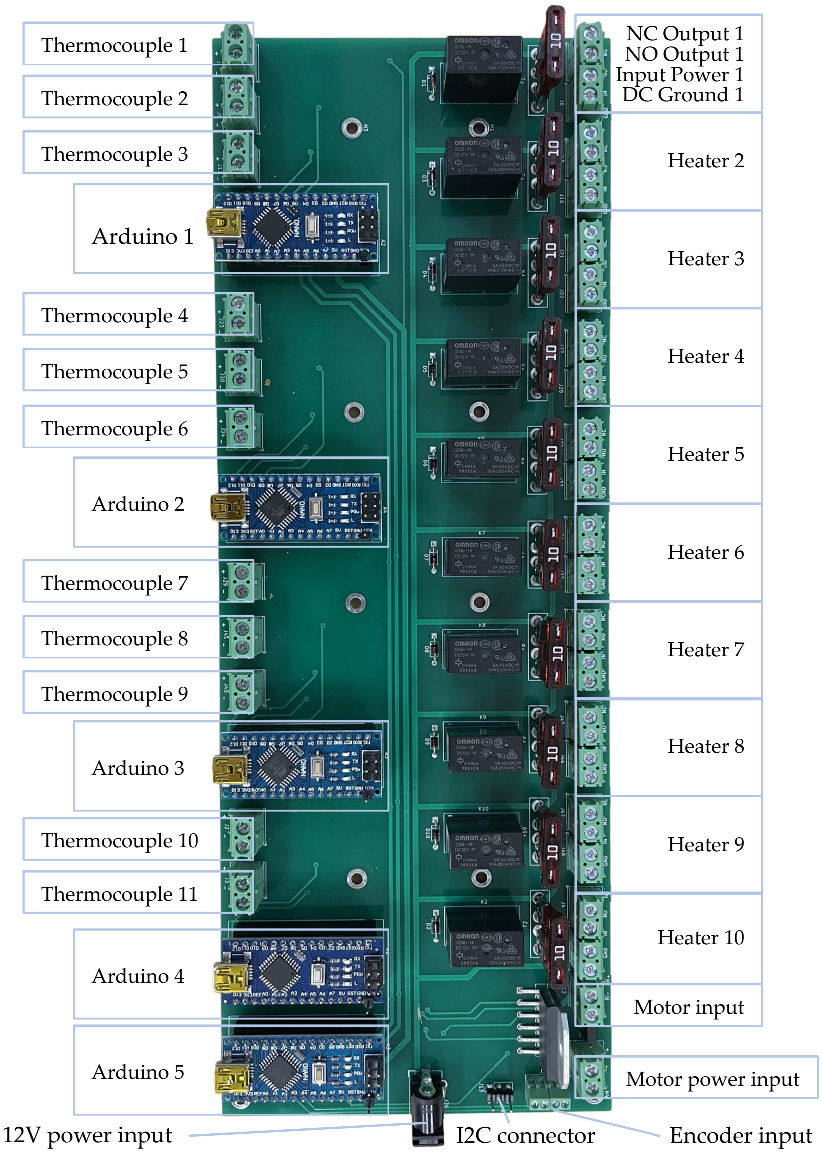


**Figure 8.** Integrated pyrolysis board connection diagram.

As shown in Table 2, the cost of the single board design was greatly reduced when compared with the BREAD system.

**Table 2.** 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 (12 V 5 A) | 1 | $1.18 | $1.18 |
| Capacitor 10 nF | 14 | $0.27 | $3.78 |
| Capacitor 10 µF | 3 | $0.81 | $2.43 |
| Diode | 10 | $0.14 | $1.40 |
| Fuse Holder 10 A | 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 12 V 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 01 × 02 | 33 | $0.81 | $26.73 |
| Screw Terminal 01 × 04 | 1 | $2.00 | $2.00 |
| Automotive Fuse 10 A | 1 | $1.64 | $1.64 |
| | | Total | $348.35 |

### 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 between the two controllers and experiments.

## 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.

### 3.1. Pyrolysis Temperature Control

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 3.

After tuning the system, the temperature of heating zone 1 (see Figure 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 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.

**Table 3.** PID tuning parameters.

| Parameter | Value |
|---|---|
| $K_u$ | 64 |
| $T_u$ | 150 s |
| $K_P$ | 28.8 |
| $K_I$ | 0.23 |

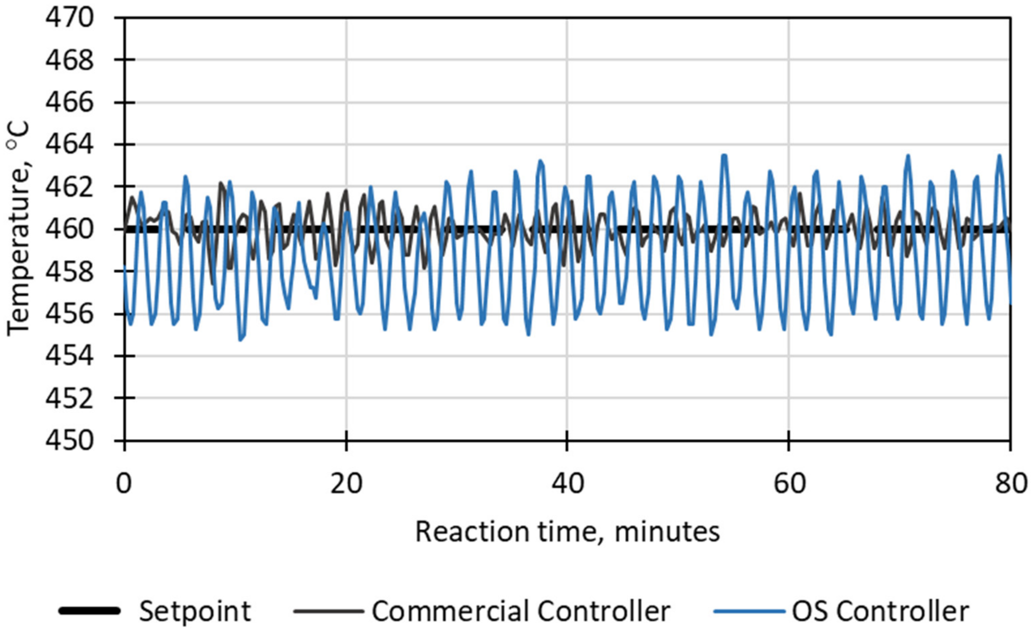


**Figure 9.** Temperature traces for the inside of heating zone 1 of the pyrolysis reactor over an 80 min pyrolysis reaction for both a commercial controller and the OS controller at identical operating conditions and at a temperature setpoint of 460 °C and feed flow rate of 0.7 kg/h. The average absolute error is 0.49% for the OS controller and 0.14% for the commercial controller.

## *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 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 10A). Overall, the product distribution for the two pyrolysis experiments is remarkedly consistent (Figure 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 10, seems to validate the performance of the OS controller, however, the product quality must also be tested.

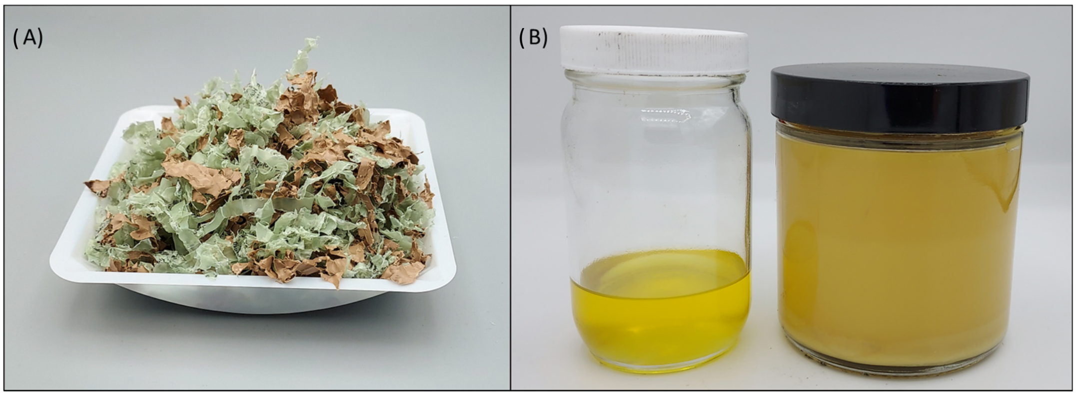


**Figure 10.** Shredded waste MRE plastic (**A**) is broken down into oil (left (**B**)), wax (right (**B**)), and gas (not pictured).

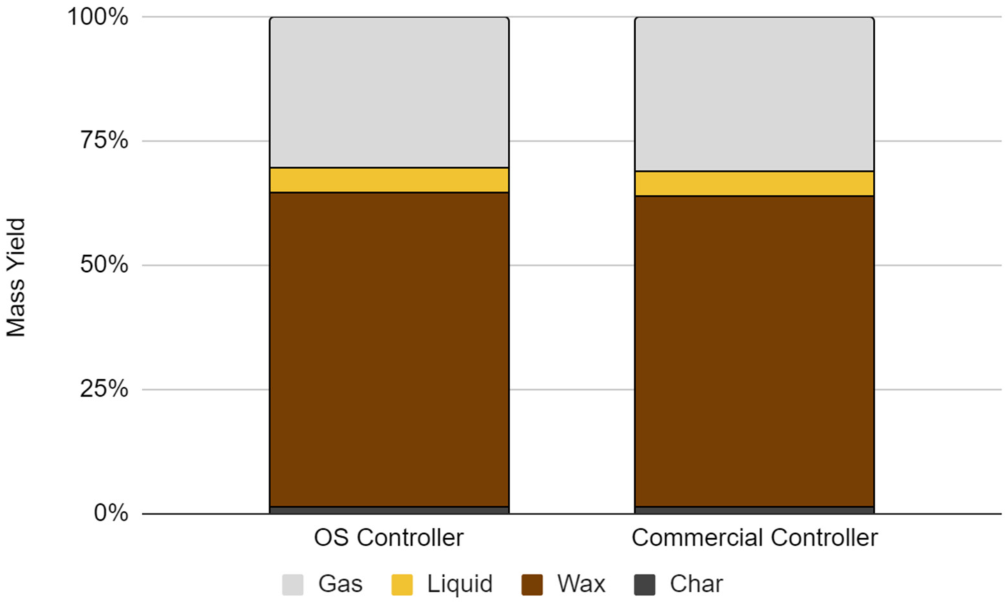


**Figure 11.** Pyrolysis mass yields for experiments using OS and commercial controller.

### 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 12) and wax (Figure 13) are qualitatively and quantitatively compared for each controller. The pyrolysis oil produced by both controllers contains primarily alkenes from C6 to C15 (Figure 11) while the wax contains a mix of alkenes and alkanes from C15 to C30 (Figure 12). 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 A7 in the Appendix B). The avereage aboslute 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.

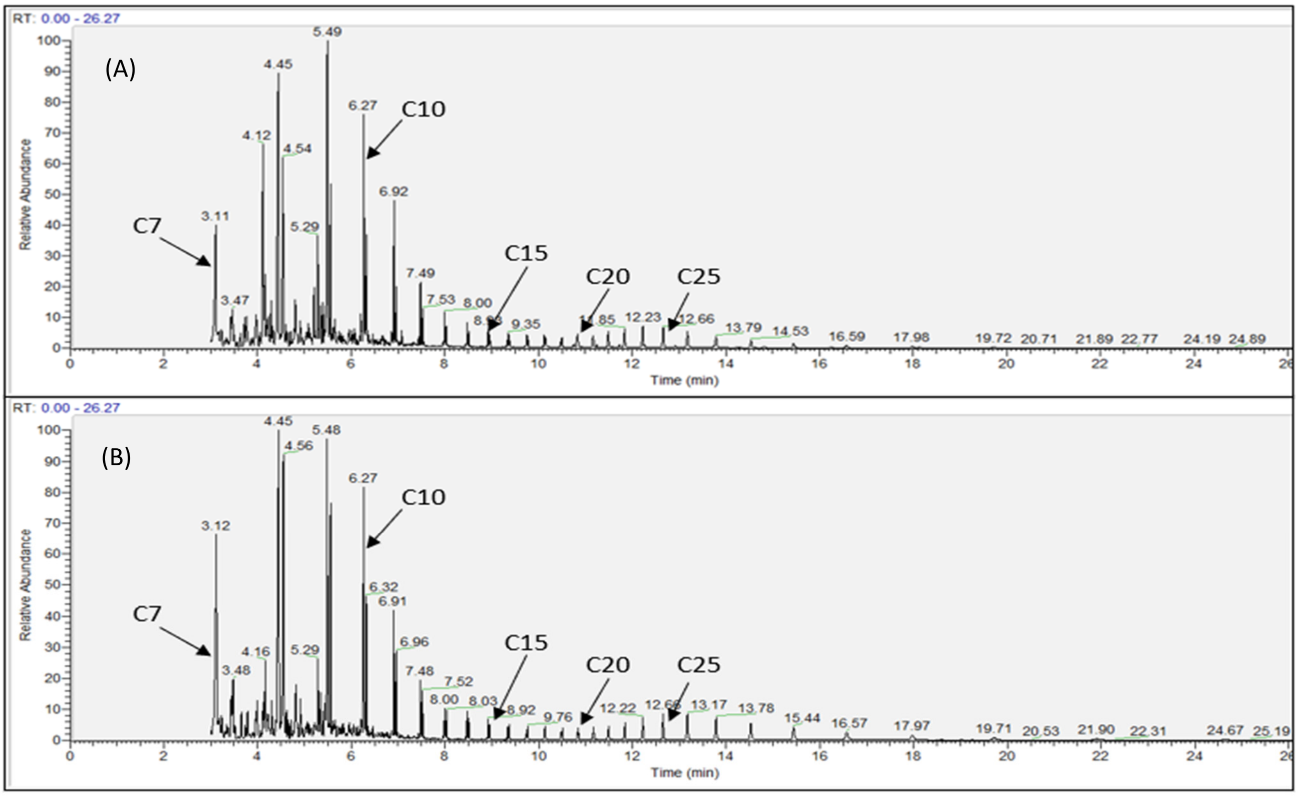


**Figure 12.** GC–MS chromatogram of pyrolysis oil produced using the OS control system (**A**) and the commercial control system (**B**). Key peaks of interest are labeled, while unlabeled peaks are one carbon number apart from each other.

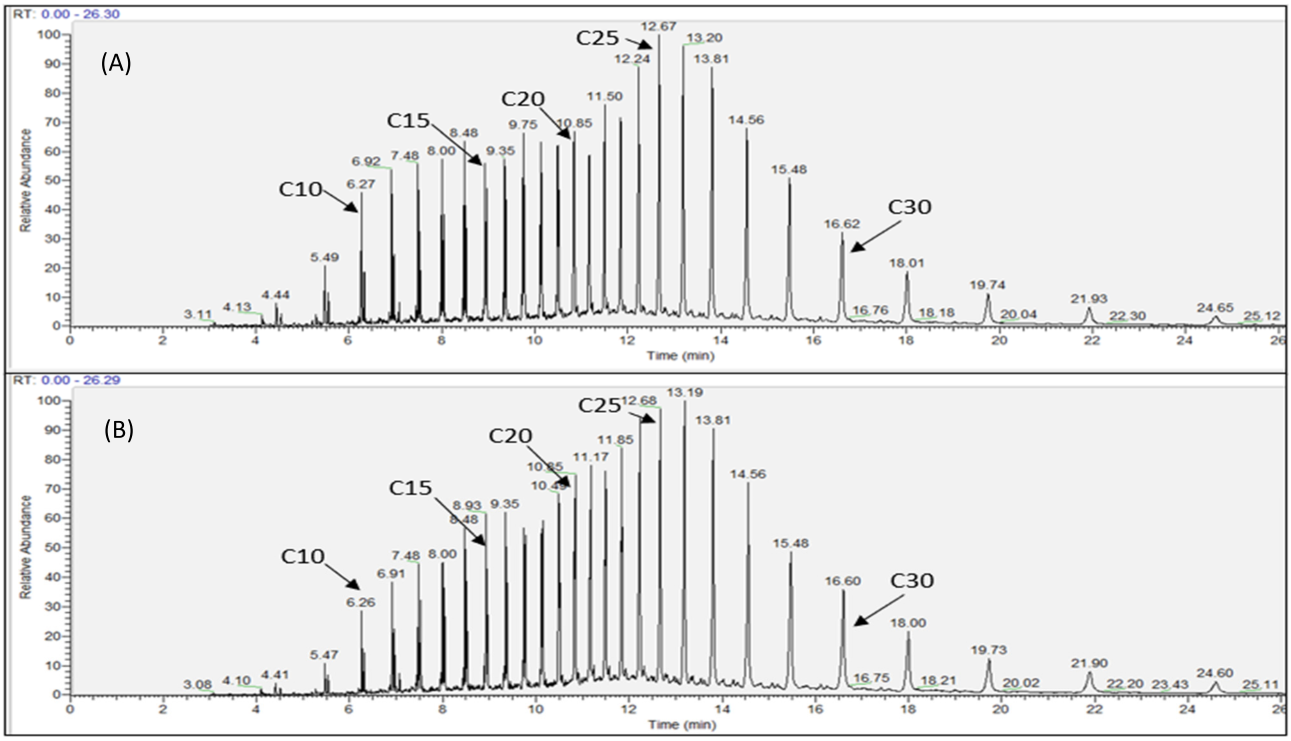


**Figure 13.** GC–MS chromatogram of pyrolysis wax produced using the OS control system (**A**) and the commercial control system (**B**). Key peaks of interest are labeled while unlabeled peaks are one carbon number apart from each other.

## 4. Discussion

### *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 recording 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].

### *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. 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.

## 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.

**Author Contributions:** Conceptualization, J.M.P.; methodology, F.K.H., D.K., E.M., S.C., S.W. and J.M.P.; validation, F.K.H., D.K. and J.M.P.; formal analysis, F.K.H. and D.K.; investigation, F.K.H. and D.K.; resources, J.M.P. and S.W.; data curation, F.K.H. and J.M.P.; writing—original draft preparation, F.K.H., D.K. and J.M.P.; writing—review and editing, F.K.H., D.K., E.M., S.C., S.W., D.S. and J.M.P.; visualization, F.K.H. and D.K.; supervision, J.M.P. and D.S.; project administration, J.M.P. and D.S.; funding acquisition, J.M.P. and D.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** Funding for this work was provided by the Defense Advanced Research Projects Agency ReSource program cooperative agreement HR00112020033, the NSF SBIR Phase II grant number: 1746480, and the Thompson Endowment. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

**Data Availability Statement:** Data are available upon request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. BREAD System Price Breakdown

**Table A1.** 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 10 µF | 2 | \$0.81 | \$1.62 |
| Capacitor 10 nF | 2 | \$0.27 | \$0.54 |
| Zener Diode 5.1 V | 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 A2.** SLC_DCMT bill of materials (cost in CAD, sourced from Digikey).

| Component | Number | Cost Per Unit | Total Cost |
|---|---|---|---|
| Capacitor 10 nF | 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 10 µF | 4 | \$0.81 | \$3.24 |
| 01 × 10 Female Header | 1 | \$1.14 | \$1.14 |
| | | Total | \$84.55 |

**Table A3.** Loaf bill of materials (cost in CAD, sourced from Digikey).

| Component | Number | Cost Per Unit | Total Cost |
|---|---|---|---|
| 10 µF | 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 |

**Table A4.** 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 |

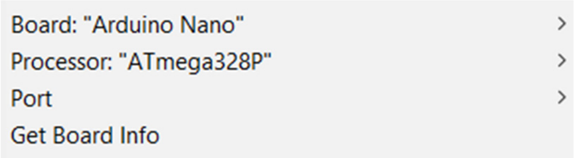
## Appendix B. Setting Up a New Slice

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".



Board: "Arduino Nano"
Processor: "ATmega328P"
Port
Get Board Info

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:

1. On the Linux board, open "devices.json".
2. Each thermocouple and thermistor can be added by defining their specific parameters in the DEVICES section (Table A5).
3. The heater actuator is added by defining its parameters in the CONTROL section (Table A6).
4. Save "devices.json" after adding all sensors and actuators.

**Table A5.** Sensor setup parameters.

| Parameter | Value | Description |
|---|---|---|
| name | "Thermo 1" | Device name. Referred to in CONTROLS section |
| address | 15 | I2C address of Slice |
| unit | "C" | Data units sent by Slice |
| form | "byte" | Data form sent by Slice |
| req_msg | 1,84 | Specific message to request data. For thermocouple 1 this is "T" (in ASCII 0 × 84) followed by 1 |
| delay | 0.3 | Delay between readings in seconds |
| read_length | 4 | Number of bytes to read. Thermocouple data in float format (4 bytes) |

**Table A6.** 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 |

**Table A7.** 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% | - |

**Table A7.** *Cont.*

| Carbon Number | Wax |  |  | Oil |  |  | Gas |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Commercial | Open Source | % Absolute Error | Commercial | Open Source | % Absolute Error | Commercial | Open Source | % Absolute Error |
| 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% | - |

## References

1. Kogut, B.; Metiu, A. Open-Source Software Development and Distributed Innovation. *Oxf. Rev. Econ. Policy* **2001**, *17*, 248–264. [CrossRef]
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. [CrossRef]
5. Oberloier, S.; Pearce, J.M. General Design Procedure for Free and Open-Source Hardware for Scientific Equipment. *Designs* **2018**, *2*, 2. [CrossRef]
6. Pearce, J.M. Sponsored Libre Research Agreements to Create Free and Open Source Software and Hardware. *Inventions* **2018**, *3*, 44. [CrossRef]
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. [CrossRef]
8. Pearce, J.M. Economic Savings for Scientific Free and Open Source Technology: A Review. *HardwareX* **2020**, *8*, e00139. [CrossRef]
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. [CrossRef]
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. [CrossRef]
11. Oberloier, S.; Pearce, J.M. Open Source Low-Cost Power Monitoring System. *HardwareX* **2018**, *4*, e00044. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
16. Pandey, G.; Vora, A. Open Electronics for Medical Devices: State-of-Art and Unique Advantages. *Electronics* **2019**, *8*, 1256. [CrossRef]
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. [CrossRef]
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*, ENEURO.0212-20.2020. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]

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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
27. Woo, W.; Richards, W.; Selker, J.; Udell, C. WeatherChimes: An Open IoT Weather Station and Data Sonification System. *HardwareX* **2023**, *13*, e00402. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
33. Niehaus, K.; Westhoff, A. An Open-Source Data Acquisition System for Laboratory and Industrial Scale Applications. *Meas. Sci. Technol.* **2022**, *34*, 027001. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
38. Nanda, S.; Berruti, F. Thermochemical Conversion of Plastic Waste to Fuels: A Review. *Environ. Chem. Lett.* **2021**, *19*, 123–148. [CrossRef]
39. Papari, S.; Bamdad, H.; Berruti, F. Pyrolytic Conversion of Plastic Waste to Value-Added Products and Fuels: A Review. *Materials* **2021**, *14*, 2586. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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).
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. [CrossRef]
47. FakhrHoseini, S.M.; Dastanian, M. Predicting pyrolysis products of PE, PP, and PET using NRTL activity coefficient model. *J. Chem.* **2013**, *2013*, 487676. [CrossRef]
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, 2023.
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
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. [CrossRef]
61. Urban, P.L. Universal Electronics for Miniature and Automated Chemical Assays. *Analyst* **2015**, *140*, 963–975. [CrossRef]
62. Urban, P.L. Prototyping Instruments for the Chemical Laboratory Using Inexpensive Electronic Modules. *Angew. Chem. Int. Ed.* **2018**, *57*, 11074–11077. [CrossRef]
63. Urban, P.L. Open-Source Electronics As a Technological Aid in Chemical Education. *J. Chem. Educ.* **2014**, *91*, 751–752. [CrossRef]
64. Ngo, T.D. Open-Source Electronics Platforms: Development and Applications. *Electronics* **2019**, *8*, 428. [CrossRef]
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).