



# A low-cost, accessible, and high-performing Arduino-based seawater pH control system for biological applications



Keegan M. McLean, Alexis L. Pasulka, Emily E. Bockmon \*

California Polytechnic State University, San Luis Obispo 1 Grand Ave, San Luis Obispo, CA 93407, USA

## ARTICLE INFO

### Article history:

Received 17 June 2021  
Received in revised form 29 October 2021  
Accepted 4 November 2021

### Keywords:

Ocean acidification  
pH-stat  
CO<sub>2</sub> gas bubbling  
Solenoid  
Control aquaria  
Climate change  
Biological responses

## ABSTRACT

In the last two decades, the need for seawater pH control methodologies paralleled the rise in attention to the biological impacts of ocean acidification. Many effective and high-performing systems have been created, but they are often expensive, complex, and difficult to establish. We developed a system that is similarly high performing, but at a low cost and with a simple and accessible design. This system is controlled by an Arduino Nano, an open-source electronics platform, which regulates the flow of CO<sub>2</sub> gas through electric solenoid valves. The Arduino and other inexpensive materials total ~\$150 (plus CO<sub>2</sub> gas and regulator), and a new treatment can be added for less than \$35. Easy-to-learn code and simple wire-to-connect hardware make the design extremely accessible, requiring little time and expertise to establish. The system functions with a variety of pH probes and can be adapted to fit a variety of experimental designs and organisms. Using this set up, we were able to constrain seawater pH within a range of 0.07 pH units. Our system thus maintains the performance and adaptability of existing systems but expands their accessibility by reducing cost and complexity.

© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## Specifications table

Hardware name	<i>Arduino Aquarium pH Control System (AAPCS)</i>
Subject area	<ul style="list-style-type: none"> <li>• Biological Sciences</li> <li>• Environmental Sciences</li> <li>• Marine Sciences</li> </ul>
Hardware type	<ul style="list-style-type: none"> <li>• Biological laboratory experiments</li> <li>• Educational tools and open-source alternatives to existing infrastructure</li> </ul>
<b>Open-Source License</b>	GNU GPL v3
Cost of Hardware	~\$150 USD (+CO <sub>2</sub> gas and regulator)
<b>Source File Repository</b>	<a href="https://doi.org/10.17605/OSF.IO/3CDVZ">https://doi.org/10.17605/OSF.IO/3CDVZ</a>

\* Corresponding author.

E-mail address: [ebockmon@calpoly.edu](mailto:ebockmon@calpoly.edu) (E.E. Bockmon).

## 1. Hardware in context

Attention to the biological implications of ocean acidification has increased dramatically in the last two decades [1–4], underscoring the need for methods to manipulate and monitor seawater pH in controlled environments. The development of such methods has rapidly accelerated in recent years and thus expanded access to this important research. From this work, a variety of high-performing systems have emerged that support a wide array of species and experimental designs.

Seawater pH control of experimental aquaria is commonly achieved either by direct addition of CO<sub>2</sub> or mixed gases [5–13] or mixing with CO<sub>2</sub>-enriched seawater [11,14–16]. These techniques serve a wide variety of experimental designs including flow-through capabilities [7,12,15], which are necessary for many organisms, and multi-stressor designs [5,12,13], enabling investigations of the potentially synergistic impacts of ocean acidification, ocean warming, and hypoxia [1,17]. The existing combinations of pH manipulation techniques and experimental designs enable scientists to study a diverse range of organisms. Seawater pH in these experimental systems can be monitored reliably and without frequent calibration through the use of a spectrophotometric pH measurement system [14] or stable ISFET electrodes, often Honeywell Durafet probes [5,13,18]. In addition to their diversity, these systems are also high-performing and often able to control pH within hundredths or thousandths of a pH unit [5,9,13,15].

Despite the advances made by these systems, they can be burdensome to establish. Many of the systems described above carry costs on the order of \$1,000 s to \$10,000 s USD to reproduce [e.g. 5]. They often require advanced hardware or custom-made components to create intricate designs. Furthermore, some systems require expertise in software programming or instrumentation [13]. Thus, there is room for progress in expanding the accessibility of ocean acidification research.

The use of Arduino platforms in seawater pH modulation is an important step in advancing this goal [7,19,20]. Arduino platforms are comprised of programmable physical circuit boards and a companion software—Integrated Development Environment (IDE). The physical circuit boards—microcontrollers—are low cost, easy to use, and contain processing, memory, input, and output power. The Arduino IDE is open-source and based in a simplified C++ language, making it easy to learn and operate. Together, the Arduino hardware and IDE provide an accessible way for scientists and individuals alike to design and control circuits of varying sizes and complexities. Low et al. recently demonstrated this by controlling a complex, robust, and high-performing pH modulation system with an Arduino Uno [19]. While these systems have made important advances in increasing their accessibility, they still carry some of the burdens described above, including relatively high system complexity and the need for custom-made hardware. Therefore, there is still sizeable room to increase the convenience of these systems by lowering their cost and complexity.

Here we introduce a method for seawater pH control that is low in cost, simple in design, and requires only a basic understanding of coding and electrical assembly. We rely entirely on commercially available components that can be easily connected with solid hook-up wire, reducing the time and expertise required to establish a working system. This simple design allows for great variety in the experimental designs and study organisms. Our system establishes new minimums in cost and assembly burdens without sacrificing control quality or adaptability, creating accessible and valuable methods that expand the opportunities to participate in this important research.

## 2. Hardware description

The Arduino Aquarium pH Control System (AAPCS) controls seawater pH by injecting small amounts of CO<sub>2</sub> gas directly into an experimental aquarium using an active feedback control mechanism (Fig. 1a). Seawater pH is monitored in real time by pH probes whose data is stored digitally and supplied to an Arduino Nano. The Arduino output is wired to a relay and electric solenoid valve, which controls the flow of pressurized CO<sub>2</sub> gas directly into seawater aquaria (Fig. 1a).

AAPCS is designed to work with any *in-situ* pH probe that monitors continuously and outputs a proportional 4–20 mA current. We used Durafet III ISFET sensors connected to a Honeywell UDA 2182 Dual Analyzer. These probes are precise, stable over extended time periods, and widely used in the oceanographic community. The UDA and any probe which outputs a 4–20 mA current can be directly wired to the Arduino. Glass electrodes and other probes without an internal current output will require an electrode amplifier to attain the required 4–20 mA current. Regardless of the pH sensing technology utilized, the sensor must be calibrated and evaluated for stability during an experiment.

The Arduino receives the pH data from the probe and integrates it into a feedback circuit (Fig. 1b). Arduino's open-source IDE, based in C++, allows for easily programmable code to be directly uploaded to each Arduino and to thus control each circuit. The code contains equation (1) which interprets the electrical current from the pH probe and translates it into a pH value. This pH value is then defined as a float variable used later to control CO<sub>2</sub> additions.

$$\text{float}pH1 = \left( \text{sensorValue1} \left( \frac{5}{1023} \right) \left( \frac{1}{312.5} \right) \right) \left( \left( \frac{pH_{high} - pH_{low}}{20 - 4} \right) (1000) \right) + pH_{low} \quad (1)$$

The Arduino Nano contains a 10-bit analog to digital converter, which translates input voltages between 0 V and the operating voltage of 5 V into integer values (*sensorValue1*) between 0 and 1023. The input voltage is provided by the 4–20 mA electrical current converted by a 249 Ohm resistor. This conversion is calculated in equation (1) using a value of 312.5 (rather than 249), which also accounts for the input impedance of the Arduino's analog ports. This value was determined empirically through manual adjustments until the pH values read by the Durafet probes matched those interpreted by the equation. The

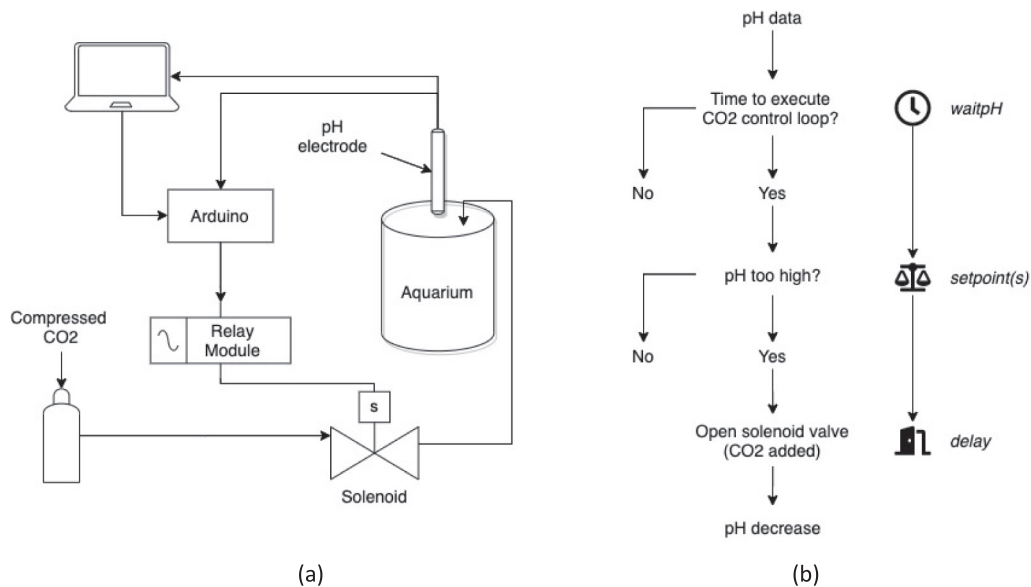


Fig. 1. (a) General system diagram depicting Arduino control of CO<sub>2</sub> gas to individual aquaria. (b) Flow-chart schematic of Arduino IDE code functions.

```

(a)
int delay4 = 32;
int delay3 = 24;
int delay2 = 20;
int delay1 = 16;

(b)
if (pH1 > setpoint1) {
  digitalWrite(6, HIGH);
  delay(delay4);
  digitalWrite(6, LOW);
  Serial.println("CO2 addition (4) pH1 ----- 4");
}

```

Fig. 2. Arduino IDE code for control of (a) delay values, and (b) pH setpoints.

operable pH range—specified by the researcher—is mapped onto the 4–20 mA range of electrical input current from the pH probe, converted to Amps. The product of these terms creates a delta pH value—the proportion of which, within the specified pH range, will be equal to the proportion of the input current within the 4–20 mA range. This delta pH value is then added to the minimum pH value specified to result in the final pH.

The pH value produced by equation (1) passes through a series of if/then loops, which control the addition of CO<sub>2</sub> (Fig. 1b). The code compares the observed pH with a programmable pH *setpoint*, signaling a CO<sub>2</sub> addition when  $pH_{\text{observed}} > pH_{\text{setpoint}}$ . The pH *setpoints* are chosen within the CO<sub>2</sub> control loops. The magnitude of each CO<sub>2</sub> addition is controlled by independent programmable *delay* variables (Fig. 2a) that specify the duration that the solenoid valve remains open, thus controlling the amount of CO<sub>2</sub> that enters the aquaria and subsequent decrease in pH (Fig. 1b). The frequency of pH feedback and potential CO<sub>2</sub> additions is controlled by the customizable *waitpH* variable, which indicates how often the observed pH is sent through the logic gate structure. The user can thus control all three aspects of pH control: the frequency of potential CO<sub>2</sub> additions, the magnitude of those additions, and the structure of the control loops, as multiple loops can be layered into a customizable hierarchy. In total, this allows the user immense control over the nature and precision of pH control (Fig. 2b). The pH data is stored directly on a microSD card wired to the Arduino and is displayed through the Arduino IDE Serial Monitor on the computer for real-time monitoring. All described sections of code are included in the provided C1 Sys Control file (Design files, Table 1).

When the code indicates a CO<sub>2</sub> addition, the signal is outputted from the Arduino and sent through a 4-channel relay module and finally to a 12 V normally closed electric solenoid valve. The relay module controls the solenoid valve power and ensures that it opens only when signaled by the Arduino output. This regulates the flow of CO<sub>2</sub>, which travels from a pressurized gas tank through the electric solenoid valve and finally to the aquarium (Fig. 1a).

Special care should be taken to keep the circuitry dry and free from seawater spills. The Arduino Nano and relay module circuitry can be optionally covered or encased in a container. However, this reduces accessibility to the hardware. It is easy to contain the aquaria (in an incubator, for example) and cut lengths of tubing that allow for an adequate separation of wet and dry components.

The simple and accessible nature of the AAPCS components allows for customizability and expansion of the system to meet a variety of experimental designs. Additional treatments can be easily incorporated by adding pH probes and either wiring additional solenoid valves or splitting existing CO<sub>2</sub> tubes. A variety of experimental scales can be supported either by increasing the magnitude or frequency of CO<sub>2</sub> additions in the code. The magnitude of pH decrease per CO<sub>2</sub> addition can also be adjusted by the height of the tubing or by adding aquarium stones to increase gas surface area and thus gas equilibration. Seawater flow-through capabilities can be achieved by adding and controlling the chemistry of a header tank which feeds multiple experimental aquaria. Header tanks avoid the impacts of bubbling directly into experimental aquaria, which can negatively impact some organisms [11,21]. However, the degassing and mixing associated with experimental aquaria fed by header tanks makes tight pH control difficult [15] and necessitates direct monitoring of the chemistry of individual experimental aquaria.

The main AAPCS hardware, including Arduino Nano (with ATmega328 microcontroller), breadboard, relay module, solenoid valves, gas tubing and accessories, can be purchased for ~\$150 USD (see Bill of Materials). Compressed CO<sub>2</sub> gas (variable pricing) and a regulator (range \$50 USD – \$400 USD) are also required. Additional pH treatments can be added for \$35 USD. Thus, the accessibility, customizability, and low cost of AAPCS make it a convenient and efficient option for biological researchers seeking a reliable system to control seawater pH.

- Seawater pH control system for biological experiments
- Low-cost alternative to more expensive available options
- Accessible to researchers with little instrument development or coding experience
- Adaptable system to fit a variety of marine species and experimental designs

### 3. Design files

**Table 1**

Summary of supplemental design files.

Design file name	File type	Open-source license	Location of the file
C1 Sys Control Code	Arduino IDE	GNU GPL v3	<a href="https://doi.org/10.17605/OSF.IO/3CDVZ">https://doi.org/10.17605/OSF.IO/3CDVZ</a>
C2 Sys Stop Code	Arduino IDE	GNU GPL v3	<a href="https://doi.org/10.17605/OSF.IO/3CDVZ">https://doi.org/10.17605/OSF.IO/3CDVZ</a>

- Sys Control File: Arduino IDE code that monitors pH and controls CO<sub>2</sub> gas additions
- Sys Stop File: Arduino IDE code that stops system pH control

### 4. Bill of materials

**Table 2**

Bill of materials. Components indicated with (\*) are sold in multi-packs, which will provide excess components if only one pH control treatment is desired.

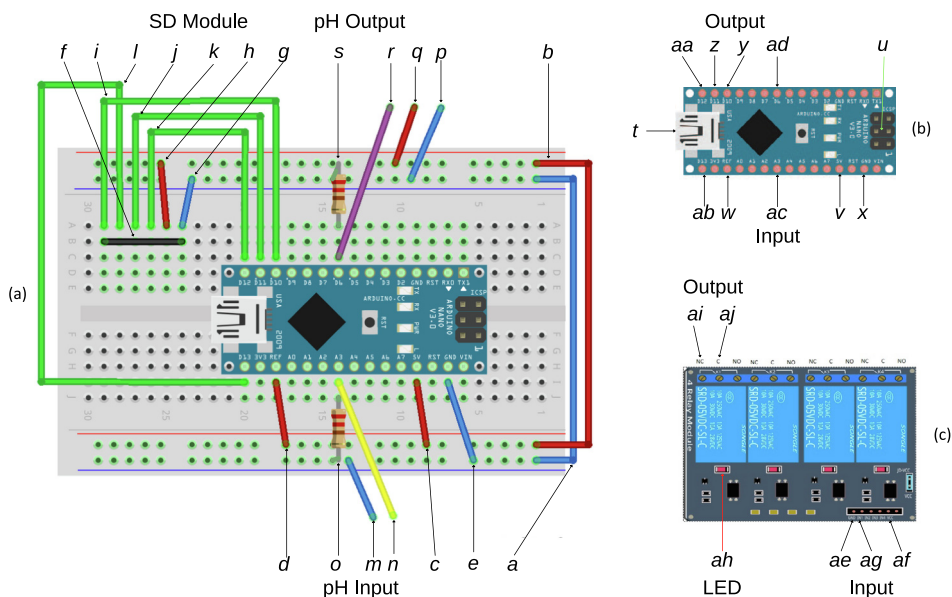
Designator	Component	Number	Cost per unit (USD)	Total cost (USD)	Source of materials
M1	Arduino Nano	1	\$ 19.80	\$ 19.80	<a href="#">Amazon</a>
M2	USB to microUSB cable adapter	1	\$ 5.09	\$ 5.09	<a href="#">Amazon</a>
M3	Solderless breadboard	1	\$ 9.88	\$ 9.88	<a href="#">Amazon</a>
M4	Solid hook-up wire	1*	\$ 8.25	\$ 8.25	<a href="#">Amazon</a>
M5	MicroSD card module	1	\$ 1.99	\$ 1.99	<a href="#">Newegg</a>
M6	2G microSD flash memory card	1	\$ 9.75	\$ 9.75	<a href="#">Amazon</a>
M7	249 Ohm resistor	1	\$ 0.40	\$ 0.40	<a href="#">Elliot Electronics Supply</a>
M8	4.7 k Ohm resistor	1*	\$ 4.99	\$ 4.99	<a href="#">Amazon</a>
M9	Breadboard jumper wires	4*	\$ 6.98	\$ 6.98	<a href="#">Amazon</a>
M10	4-channel relay module	1	\$ 9.99	\$ 9.99	<a href="#">Amazon</a>
M11	Electronic solenoid valve – ¼" 12 V DC	1	\$ 10.99	\$ 10.99	<a href="#">US Solid</a>
M12	DC female adapter	1	\$ 2.99	\$ 2.99	<a href="#">Show Me Cables</a>
M13	12 V AC-DC power converter	1	\$ 10.00	\$ 10.00	<a href="#">TRC Electronics</a>
M14	AC power cord	1	\$ 3.99	\$ 3.99	<a href="#">TRC Electronics</a>
M15	Compressed CO <sub>2</sub> gas	1	---	---	---
M16	CO <sub>2</sub> gas regulator: CGA-320	1	---	---	---
M17	Tubing ¼" OD	1*	\$ 31.14	\$ 31.14	<a href="#">Grainger</a>
M18	Push-to-connect tube fitting	1	\$ 6.99	\$ 6.99	<a href="#">Amazon</a>
M19	Push-to-connect unions/joints	1	\$ 6.99	\$ 6.99	<a href="#">Amazon</a>
M20	Check valve	1	\$ 9.06	\$ 9.06	<a href="#">US Plastic</a>
M21	Aquarium stone	1	\$ 1.30	\$ 1.30	<a href="#">Amazon</a>

## 5. Build instructions

The build instructions below are written for the construction of the described control system for a single pH treatment (spanning the probe to the end of the CO<sub>2</sub> delivery tubing). Instructions for adding separate treatments are embedded below. Indicators with a capital letter and a number (e.g. M1, M3) refer to components in the Bill of Materials (Table 2). Uncapitalized and italicized letter indicators (e.g. location *y*, wire *q*) refer to subcomponents in Fig. 3. Additional step-by-step images are provided in the Supplementary Figures.

### 5.1. Circuitry: Setup

- Test the Arduino Nano (M1) function by connecting microUSB port (*t* in Fig. 3b) to a computer using a general USB to microUSB cable adapter (M2).
  - Green LEDs on Arduino (*u* in Fig. 3b) should light to indicate power supply.
  - Then disconnect.
- Insert Arduino into the center of a solderless breadboard (M3) to allow for organization and expansion of wiring components.
  - The Arduino must be spanning the gap down the center of the breadboard so that it connects the two halves (A-E half and F-J half) and is accessible to the breadboard connections.
  - Note that A-E spots in the same numbered column on the breadboard are connected, as are F-J within the same numbered column. However, the two halves (A-E and F-J) are separate. Thus, a current delivered to position 1A on the breadboard can be received by positions 1B-1E but will not affect any other positions.
  - The ground and power bus strips are separate from each other and connect along the numbered rows. Thus, a wire in either bus strip in row 30 can be connected to any wire in the same bus strip in rows 1–29.
- Connect the breadboard ground bus strips (indicated by thin blue lines on the breadboard) and power bus strips (thin red lines) with individual pieces of 22-gauge solid hookup wire (M4; blue wire *a* and red wire *b*, respectively, Fig. 3a).
  - These wires connect the bus strips across the two halves of the breadboard, so that both sides are powered and grounded.
- Confirm that the Arduino is not powered before continuing. Wire the 5 V and REF Arduino pins (*v* and *w* in Fig. 3b, respectively) to the breadboard power bus strip (red wires *c* and *d*, Fig. 3a).
  - When complete, the circuit should match that shown in Fig. 4.



**Fig. 3.** Schematic diagrams of (a) Arduino circuit and breadboard, (b) Arduino Nano, and (c) relay module. Wires are color-coded by function (blue = GROUND, red = POWER, yellow = pH INPUT, purple = pH OUTPUT, green = SD MODULE). Relevant subcomponents are letter-coded for designation in build instructions. Colored arrows indicate an LED color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 5.2. Circuitry: SD module

6. Insert the microSD card adapter module (M5, left of Arduino in Fig. 5) into six open and adjacent columns on the breadboard (represented by black wire *f*, Fig. 3a). Ensure that the selected breadboard columns are free of any other hardware (wires, resistors, Arduino pins).
7. Then using solid hook-up wire:
  - a. Wire the ground bus strip to the adapter module GND pin (blue wire *g*, Fig. 3a).
  - b. Wire the power bus strip to the adapter module VCC pin (red wire *h*, Fig. 3a).
  - c. Wire Arduino D10 pin to adapter module CS pin (green wire *i*, Fig. 3a).
  - d. Wire Arduino D11 pin to adapter module MOSI pin (green wire *j*, Fig. 3a).
  - e. Wire Arduino D12 pin to adapter module MISO pin (green wire *k*, Fig. 3a).
  - f. Wire Arduino D13 pin to adapter module SCK pin (green wire *l*, Fig. 3a).
8. Insert a 2G microSD flash memory card (M6; small black card on left in Fig. 5a) into the adapter module (M5).

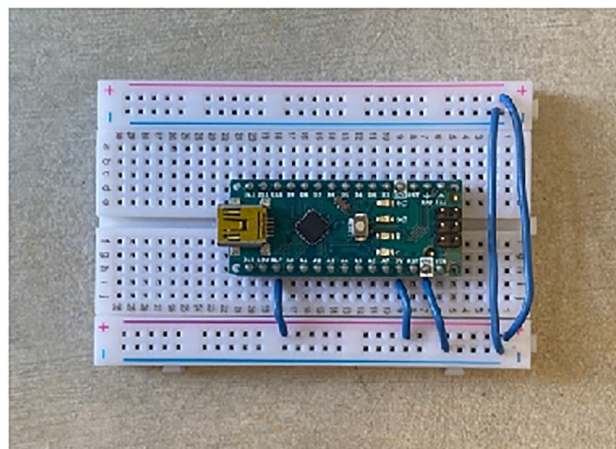
### 5.3. Circuitry: pH input

9. Connect the pH probe to the Arduino as follows:
  - a. Connect the ground wire from the pH probe or amplifier to the ground bus strip (blue wire *m*, Fig. 3a).
  - b. Connect the signal wire from the pH probe (yellow wire *n*, Fig. 3a) to Arduino pin A3 (*ac* in Fig. 3b).
10. Insert one end of a 249 Ohm resistor (M7) to the ground bus strip on the input side (resistor *o*, Fig. 3a) and the opposite end to the breadboard column aligned with the A3 input pin (Fig. 3a).
  - a. When complete, the circuit should resemble that displayed in Fig. 6.

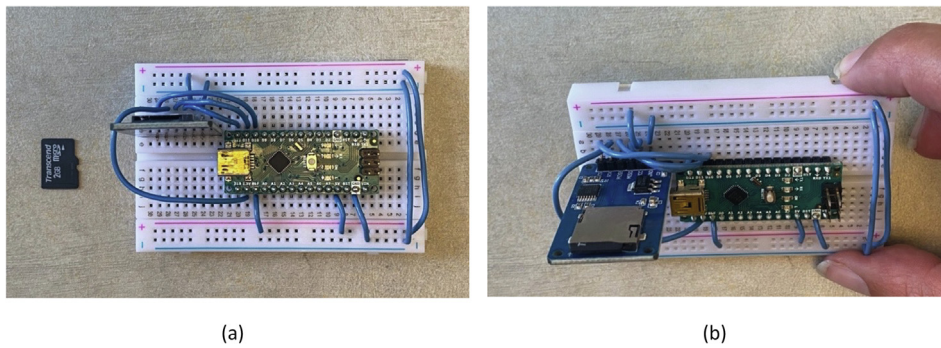
For an additional treatment, simply repeat steps 9 and 10. Wire another probe to an Arduino input pin (A0-A7) with a signal wire, ground wire, and 249 Ohm resistor.

### 5.4. Circuitry: pH output to relay module & solenoid valve

11. Connect the pH Output from the Arduino to a 4-channel relay module (M10) as follows:
  - a. Connect the male end of a male-to-female breadboard jumper wire (M9) to the breadboard ground bus strip (blue wire *p* in Fig. 3a, orange wire in Fig. 7). Then connect the female end to the GND pin on the relay module (*ae* in Fig. 3c).
  - b. Connect the male end of a male-to-female breadboard jumper wire (M9) to the breadboard power bus strip (red wire *q* in Fig. 3a, yellow wire in Fig. 7). Then connect the female end to the VCC pin on the relay module (*af* in Fig. 3c).
  - c. Connect the male end of a male-to-female breadboard jumper wire (M9) to the D6 output pin (purple wire *r* in Fig. 3a, black wire in Fig. 7). Then connect the female end to the IN1 pin on the relay module (*ag* in Fig. 3c).
  - d. Insert one end of a 4.7 k Ohm resistor (M8) into the breadboard ground bus strip on the output side (resistor *s* in Fig. 3a) and the opposite end to the breadboard column aligned with the D6 output pin (*ad* in Fig. 3b).



**Fig. 4.** Image summary of build instructions steps 1–5. Displays setup of Arduino Nano and initial wiring to establish the circuit.



**Fig. 5.** Image summary of build instructions steps 6–8. Displays setup and wiring of microSD card adapter module from (a) overhead and (b) side-angle view.

For an additional treatment, repeat steps 11c and 11d: connect another Arduino output pin (D2–D9) to an open INX pin on the relay module, with the X being the number of an open channel (ex: IN2). Then add a 4.7 k Ohm resistor. The ground and power wires are only necessary when establishing a new relay module. Each relay module has four channels and can therefore support four treatments.

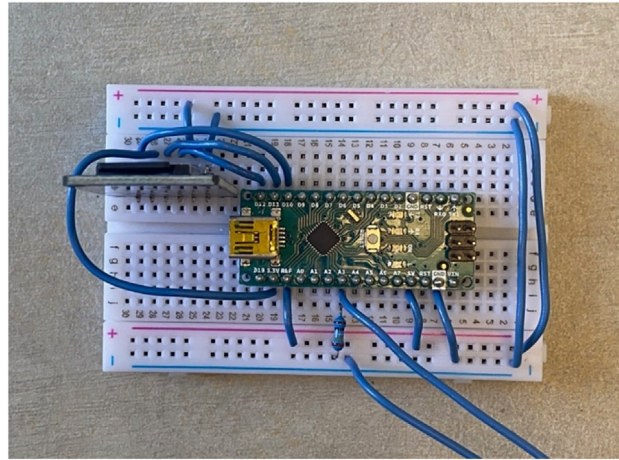
12. Test the Arduino-to-relay module connection by powering the Arduino with a computer through the USB to microUSB cable adapter (M2).
  - a. The green LED on the Arduino (*u* in Fig. 3b) should again be lit to indicate Arduino power (see Fig. 7).
  - b. The power should also be transferred to the relay module, as indicated by red LEDs at the IN1 channel (*ah* in Fig. 3c, see Fig. 7)).
  - c. Then disconnect by unplugging the Arduino from the computer.
13. On channel K1 of the relay module, loosen the screws at positions NC and C (*ai* and *aj*, respectively, Fig. 3c). Then, with the Arduino disconnected from the computer, wire an electric solenoid valve (M11) to the relay channel as follows:
  - a. Insert the white wire from the solenoid to the NC position (far left screw) on the relay module (*ai* in Fig. 3c) and gently tighten the screw (see white wire in Fig. 8).
  - b. Insert the black wire from the solenoid to the DC female adapter (M12, see black wire in Fig. 8).
  - c. Use solid hook-up wire to connect the other port on the DC female adapter to the C position (middle screw; K1) on the relay module (*aj* in Fig. 3c) and gently tighten the screw (see blue wire in Fig. 8).
14. Power the solenoid externally by connecting the DC female adapter (M12) to the 12 V AC-DC power converter (M13). Finally, insert the AC Cord (M14) into a wall outlet and plug into the AC-DC power converter (M13).
  - a. When complete, the circuit up to this point should resemble that shown in Fig. 8.

For additional treatments, extra solenoids can be powered by the same 12 V AC-DC power converter, or by separate power. To do this, modify step 13 as follows: Splice the black solenoid wires (step 13b) together, then connect to the DC female adapter. Then, split the solid hook-up wire (step 13c) from the DC female adapter to the desired number of channels on the relay module.

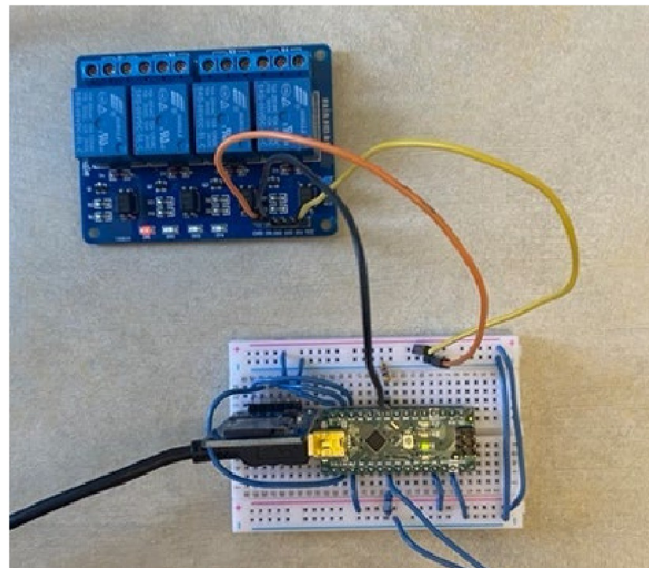
The above steps complete the circuitry for one treatment. Instructions for adding treatments are embedded as sub steps, when relevant.

### 5.5. Arduino code

15. Download and install Arduino IDE 1.8.13, available [here](#).
16. Within the IDE, click *Tools* and make the following selections:
  - a. *Board*: “Arduino Nano”
  - b. *Processor*: “ATmega328P”
  - c. *Port*: select COM port in use.
    - i. An active COM port is indicated by a green check mark.
17. Open the provided CONTROL code (**Design Files, Table 1, C1**).
18. Customize the encoded variables as follows:



**Fig. 6.** Image summary of build instructions steps 9 & 10. Displays wiring of input from pH probe (two wires leaving picture).



**Fig. 7.** Image summary of build instructions steps 11 & 12. Displays wiring of pH outputs from Arduino Nano to relay module and power testing of the circuit.

- a. The *waitpH* variable (line 5; milliseconds) controls the duration between when the CO<sub>2</sub> control loops are executed. The default value is 180,000 ms (3 min).
  - b. The input and output pins are designated in lines 34 & 35, respectively. If the default pin definitions are used, ensure that your wiring is accurate as described above.
    - i. To change the input pin, replace the default value (A3) in line 34. Pins A0-A7 must be specified with the letter A (e.g. A3, Fig. 9).
    - ii. To change the output pin, replace the default value (6) in lines 35, 36, and 152–176. Pins D2-D13 can be encoded simply with the number (e.g. 6, Fig. 9).
  - c. Indicate pH electrode range in use and encode this into line 101 by updating the *pH low* and *pH high* terms (see pH equation (1) in Hardware description).
  - d. Update the *delay* (lines 142–145; Fig. 2a) and pH *setpoint* (lines 152, 158, 164, and 170; Fig. 2b) values to match desired experimental conditions.
19. Click the arrow icon in the top right marked “upload” to upload the code to the Arduino.
- a. Flashing red and green LEDs on the Arduino indicate that the upload is in progress.
  - b. Solid green LEDs indicate upload completion and return to normal state.



20. Toggle solenoid valve power to OFF by disconnecting 12 V AC-DC power converter (M13) from DC female adapter (M12) (see Build Instructions, Step 14), to avoid accidental acidification of the aquarium upon startup.

To add pH treatments, select unutilized input and output pins and wire them as specified (steps 9–10 and 11–14, respectively). Then, update the code file by adding pin definitions (step 18b) and CO<sub>2</sub> control loops (step 18d, see Fig. 2b) when needed.

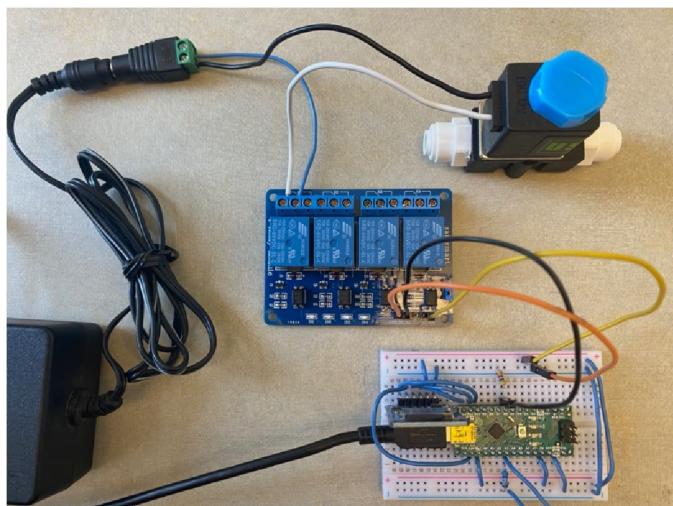
## 6. Operation instructions

### 6.1. CO<sub>2</sub> tubing and aquaria

1. Attach a CO<sub>2</sub> gas regulator (M16) to a compressed CO<sub>2</sub> gas tank (M15).
2. Insert ¼" push-to-connect tube fitting (M18) into the solenoid valve to allow for easy tubing connection.
3. Connect the CO<sub>2</sub> regulator to the solenoid input at the push-to-connect tube fitting (M17) using an appropriate length of ¼" tubing (M17).
  - a. If adding multiple treatments, cut this length of tubing, add a push-to-connect Y union (M19) to split the line, and connect to the desired number of solenoid valves.
4. Connect another length of tubing to the solenoid output and deliver to the aquaria.
  - a. The CO<sub>2</sub> delivery tubing should be long enough to provide adequate separation of the dry circuitry components (Arduino, relay, solenoid) and wet components.
5. To the end of the tubing, add a check valve (M20) above the water line to prevent back-flow (Fig. 10).
6. If an aquarium stone is desired to increase gas exchange of each addition, add a short piece of tubing to the check valve and insert an aquarium stone (M21, Fig. 10).

### 6.2. Startup

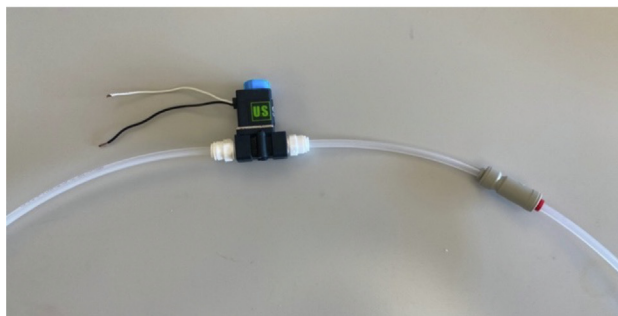
1. Insert the microSD card into the SD module.
2. Power the Arduino by inserting the cable adapter (M2) into the microUSB port (*t* in Fig. 3b).
  - a. This should result in green LEDs on the Arduino (see *n* in Fig. 3b) and a red LED on relay module K1 (see *w* in Fig. 3c).
3. Open Arduino IDE, then ensure that Arduino COM port is active by clicking *Tools > Ports*.
  - a. Green check mark and selection of Arduino Nano indicate an active COM line.
4. Open CONTROL code (C1).
5. Click *Upload* button, indicated by rightward arrow in top left corner.
  - a. Observe red-green flashing LEDs on Arduino.
  - b. Click *Tools > Serial Monitor* to observe the Serial Monitor and validate code upload.
6. Perform a final check:



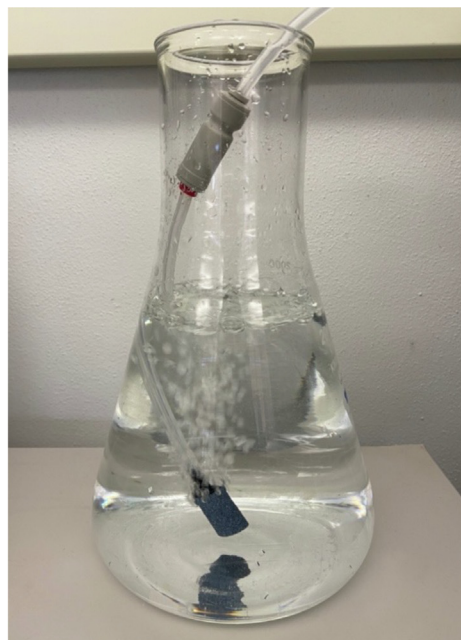
**Fig. 8.** Image summary of build instructions steps 13 & 14. Displays wiring of pH outputs from relay module to solenoid valve and subsequent powering of solenoid valve. The white, black, and blue wires from the relay module to the solenoid valve are of special interest to these steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

```
pinMode(A3, INPUT);
pinMode(6, OUTPUT);
digitalWrite(6, LOW);
```

Fig. 9. Definition of input and output pins in Arduino IDE code.



(a)



(b)

Fig. 10. Image summary of (a) operation instructions steps 2-5 and (b) step 6. Displays solenoid, check valve, and aquarium stone on CO<sub>2</sub> delivery tubing. Length of tubing between solenoid and check valve will often be longer than shown, to reach the experimental tank.

- a. Tubing present in aquaria. Aquarium stones present (if desired).
  - b. Arduino and Relay module are lit and powered (see Step 1).
  - c. Compressed CO<sub>2</sub> tank is open with normal pressures.
7. Toggle solenoid valve power to ON by connecting 12 V AC-DC power converter (M13) to DC female adapter (M12) (see **Build Instructions**, Step 14).
    - a. This should always be the last step taken when starting up and the first step taken when shutting down the system in order to prevent aquaria from being acidified. The Arduino must be powered in order to keep the relays open and prevent the powered solenoids from actually opening and allowing CO<sub>2</sub> to enter.
    - b. Thus, it is most wise to simply depower the solenoids in all times of adjusting, starting up, or shutting down the system to prevent any accidental acidifications (depowering the solenoids returns them to their normally closed state).

### 6.3. Shutdown

1. Toggle solenoid valve power to OFF by disconnecting 12 V AC-DC power converter (M13) from DC female adapter (M12) (see Build Instructions, Step 14).
2. Open STOP Code (C2). Then, click *Upload* button, indicated by rightward arrow in top left corner, to upload.
  - a. Observe red-green flashing LEDs on Arduino.
  - b. Click *Tools > Serial Monitor* to observe Serial Monitor and ensure the CO<sub>2</sub> control code is no longer running.
3. Depower Arduino by disconnecting microUSB to USB cable adapter.
4. Remove microSD Card from SD Module, then store in microSD to SD adapter (M6).

The Arduino will continue executing the code that was most recently uploaded, even if the Arduino IDE is closed or if the Arduino is depowered, then repowered. Thus, to stop the code from being executed, it is best to upload the STOP code (C2), which clears the void loop.

Extreme care should be taken with the solenoid power to prevent accidental CO<sub>2</sub> additions into the aquaria, which will result in rapid acidification. The solenoids are normally closed (NC) in their depowered state. Solenoids should ONLY be powered if the Arduino and relay module are both powered. The powered relay module closes all channels by opening gates in the circuit and preventing the powered signal from reaching and opening the solenoid. When the Arduino indicates, the relay closes, completing the circuit and powering the solenoid, which opens it for the duration specified by the relevant *delay* value in the code. In any circumstances wherein the Arduino and relay module may need to be depowered, the solenoids should also be depowered. This is best achieved by powering the solenoid at the *end* of startup procedures and depowering at the *beginning* of shutdown procedures.

## 7. Validation and characterization

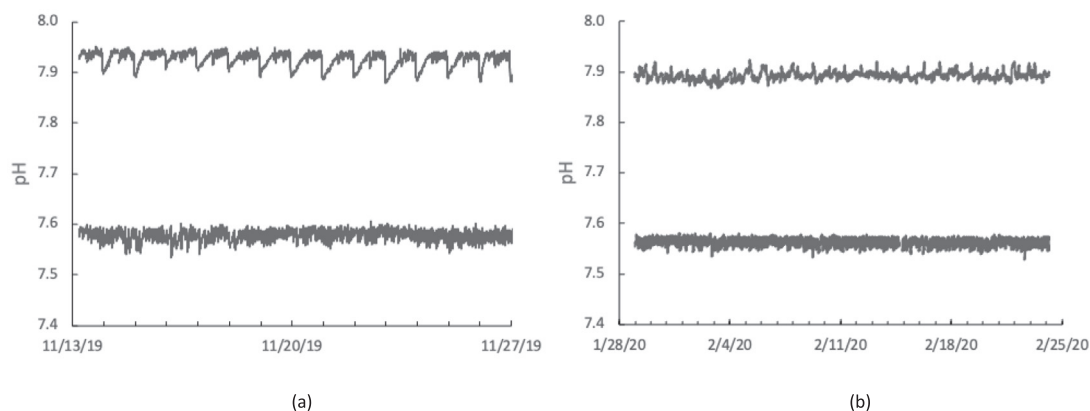
The control quality of AAPCS was tested in two individual experiments. For both experiments, two separate small-scale aquaria—5.0L aquaria in Exp1 and 20.0L aquaria in Exp2—were controlled at pH values 0.35 units apart (Table 3). Temperature control for both experiments was achieved by storing the aquaria in an incubator, thus ensuring a constant relationship between DIC and pH under constant total alkalinity and salinity conditions [22]. This also contained the aquaria and protected the dry circuitry components from any water damage. Seawater pH and temperature were monitored by Durafet III probes connected to a Honeywell UDA 2182 Dual Analyzer. Data was sent from the UDA to a laptop, where LabVIEW software displayed and stored the data [18]. The pH sensors were calibrated with a Tris buffer suitable for high ionic solutions and verified with spectrophotometric analysis of discrete samples using a purified *m*-cresol purple dye.

Seawater pH control was stable, with a maximum range of 0.07 pH units (Table 3), and the desired difference of ~ 0.35 pH units between the two aquaria was maintained for both experiments (Fig. 11). AAPCS thus approaches the pH control quality of expensive, sophisticated and high-performing systems [7,14,15], but with a simple, low-cost and accessible design.

- System controls seawater pH within  $\leq 0.07$  pH units.
- Capable of maintaining control on month-long timescales.
- Displays pH and temperature for real-time monitoring.
- Stores all data on a microSD card.
- Simple nature of setup provides for high adaptability of experimental design and scale.

**Table 3**  
Validation results for pH control experiments.

Experiment	Treatment	Average pH	Range
Exp1 (5 L)	High pH	7.93	0.07
	Low pH	7.58	0.07
Exp 2 (20 L)	High pH	7.89	0.06
	Low pH	7.56	0.05



**Fig. 11.** Time-series of pH data including results for the high and low pH treatments for (a) Experiment 1 using 4L beakers and (b) Experiment 2 using 20L aquaria. Tick marks on x-axis represent 24-hour periods. On 2/14/20 the pH probe was removed from the Exp 2 low pH treatment; thus, several hours of data have been removed from the dataset.

## 8. Conclusions

Existing methods for controlling carbonate chemistry are often expensive, sophisticated, and may come with large burdens related to carbonate chemistry expertise, technical assembly, and operation. This makes them inaccessible to the larger community of ocean acidification researchers and educators. The threat ocean acidification poses to species and communities requires broad investigation and thus the need for widely accessible methods. Our system fulfills this need, providing robust pH control without high expenses and technical burdens. This is achieved by lowering entry barriers regarding the technical knowledge, experience, and time required to establish a working system. Furthermore, the simplicity of the design and the utilized components provides potential for adaptability and customizability of the system to fit a variety of biological species and experimental designs. Our system will increase access to this important research and accelerate the discovery of new knowledge by the ocean acidification research community.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to acknowledge the preliminary design and testing work accomplished by students Julio Sierra, Natalie Weisenburger, and Kat Turk as a part of their Senior Design Project at Cal Poly, San Luis Obispo. We would also like to thank Genevieve Rice, Jackie Killam, and Hannah Allen for their work on the project and Kevin Dunham for his technical assistance. This material is based upon work supported by the California State University Council on Ocean Affairs, Science & Technology (COAST) under Award No. COAST-GDP-2018-005 and the William and Linda Frost Fund at Cal Poly, San Luis Obispo.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.ohx.2021.e00247>.

## References

- [1] T. Wernberg, D.A. Smale, M.S. Thomsen, A decade of climate change experiments on marine organisms: procedures, patterns and problems, *Global Change Biol.* 18 (5) (2012) 1491–1498, <https://doi.org/10.1111/j.1365-2486.2012.02656.x>.
- [2] A.J. Lemasson, S. Fletcher, J.M. Hall-Spencer, A.M. Knights, Linking the biological impacts of ocean acidification on oysters to changes in ecosystem services: a review, *J. Exp. Mar. Bio. Ecol.* 492 (2017) 49–62, <https://doi.org/10.1016/j.jembe.2017.01.019>.
- [3] H. Baumann, Experimental assessments of marine species sensitivities to ocean acidification and co-stressors: How far have we come?1, *Can. J. Zool.* 97 (5) (2019) 399–408, <https://doi.org/10.1139/cjz-2018-0198>.
- [4] K.J. Kroeker, R.L. Kordas, R.N. Crim, G.G. Singh, Meta-analysis reveals negative yet variable effects of ocean acidification on marine organisms, *Ecol. Lett.* 13 (11) (2010) 1419–1434, <https://doi.org/10.1111/j.1461-0248.2010.01518.x>.
- [5] E.E. Bockmon, C.A. Frieder, M.O. Navarro, L.A. White-Kershek, A.G. Dickson, Technical Note: Controlled experimental aquarium system for multi-stressor investigation of carbonate chemistry, oxygen saturation, and temperature, *Biogeosciences* 10 (9) (2013) 5967–5975, <https://doi.org/10.5194/bg-10-5967-2013>.
- [6] P.L. Jokiel, K.D. Bahr, K.S. Rodgers, Low-cost, high-flow mesocosm system for simulating ocean acidification with CO<sub>2</sub> gas, *Limnol. Oceanogr. Methods* 12 (MAY) (2014) 313–322, <https://doi.org/10.4319/lom.2014.12.313>.
- [7] G. Duarte, E.N. Calderon, C.M. Pereira, L.F.B. Marangoni, H.F. Santos, R.S. Peixoto, A. Bianchini, C.B. Castro, A novel marine mesocosm facility to study global warming, water quality, and ocean acidification, *Ecol. Evol.* 5 (20) (2015) 4555–4566, <https://doi.org/10.1002/ece3.1670>.
- [8] K.R.N. Anthony, D.I. Kline, G. Diaz-Pulido, S. Dove, O. Hoegh-Guldberg, Ocean acidification causes bleaching and productivity loss in coral reef builders, *Proc. Natl. Acad. Sci. USA* 105 (45) (2008) 17442–17446, <https://doi.org/10.1073/pnas.0804478105>.
- [9] E. Wilcox-Freeburg, A. Rhyne, W.E. Robinson, M. Tlusty, B. Bourque, R.E. Hannigan, A comparison of two pH-stat carbon dioxide dosing systems for ocean acidification experiments, *Limnol. Oceanogr. Methods* 11 (9) (2013) 485–494, <https://doi.org/10.4319/lom.2013.11.485>.
- [10] S.C. Talmage, C.J. Gobler, The effects of elevated carbon dioxide concentrations on the metamorphosis, size, and survival of larval hard clams (*mercenaria mercenaria*), bay scallops (*argopecten irradians*), and eastern oysters (*crassostrea virginica*), *Limnol. Oceanogr.* 54 (6) (2009) 2072–2080, <https://doi.org/10.4319/lo.2009.54.6.2072>.
- [11] J.P. Gattuso, H. Lavigne, Technical Note: Approaches and software tools to investigate the impact of ocean acidification, *Biogeosciences* 6 (10) (2009) 2121–2133, <https://doi.org/10.5194/bg-6-2121-2009>.
- [12] L.J. Falkenberg, B.D. Russell, S.D. Connell, Design and performance evaluation of a mesocosm facility and techniques to simulate ocean acidification and warming, *Limnol. Oceanogr. Methods* 14 (4) (2016) 278–291, <https://doi.org/10.1002/lom3.10088>.
- [13] R.B. Burrell, A.G. Keppel, V.M. Clark, D.L. Breitburg, An automated monitoring and control system for flow-through co-cycling hypoxia and pH experiments, *Limnol. Oceanogr. Methods* 14 (3) (2016) 168–185, <https://doi.org/10.1002/lom3.10077>.
- [14] C.M. McGraw, C.E. Cornwall, M.R. Reid, K.I. Currie, C.D. Hepburn, P. Boyd, C.L. Hurd, K.A. Hunter, An automated pH-controlled culture system for laboratory-based ocean acidification experiments, *Limnol. Oceanogr. Methods* 8 (12) (2010) 686–694, <https://doi.org/10.4319/lom.2010.8.0686>.
- [15] N.A. Fanguie, M.J. O'Donnell, M.A. Sewell, P.G. Matson, A.C. MacPherson, G.E. Hofmann, A laboratory-based, experimental system for the study of ocean acidification effects on marine invertebrate larvae, *Limnol. Oceanogr. Methods* 8 (8) (2010) 441–452, <https://doi.org/10.4319/lom.2010.8.441>.
- [16] X. Xu, F. Yang, L. Zhao, X. Yan, Seawater acidification affects the physiological energetics and spawning capacity of the Manila clam *Ruditapes philippinarum* during gonadal maturation, *Comp. Biochem. Physiol. -Part A Mol. Integr. Physiol.* 196 (2016) 20–29, <https://doi.org/10.1016/j.cbpa.2016.02.014>.
- [17] H.O. Pörtner, M. Langenbuch, B. Michaelidis, Synergistic effects of temperature extremes, hypoxia, and increases in CO<sub>2</sub> on marine animals: From Earth history to global change, *J. Geophys. Res. C Ocean.* 110 (9) (2005) 1–15, <https://doi.org/10.1029/2004JC002561>.

- [18] L. Kapsenberg, E.E. Bockmon, P.J. Bresnahan, K.J. Kroeker, J.P. Gattuso, T.R. Martz, Advancing ocean acidification biology using Durafet® pH electrodes, *Front. Mar. Sci.* vol. 4, no. OCT (2017) 1–9, <https://doi.org/10.3389/fmars.2017.00321>.
- [19] N.H.N. Low, C.A. Ng, F. Micheli, A low-cost modular control system for multistressor experiments, *Limnol. Oceanogr. Methods* 18 (10) (2020) 623–634, <https://doi.org/10.1002/lom3.v18.1010.1002/lom3.10389>.
- [20] J.Z. Milanovic, P. Milanovic, R. Kragic, M. Kostic, J. Chalmers, 'Do-It-Yourself' reliable pH-stat device by using open-source software, inexpensive hardware and available laboratory equipment, *PLoS One* 13 (3) (2018) e0193744, <https://doi.org/10.1371/journal.pone.0193744>.
- [21] D. Shi, Y. Xu, F.M.M. More, Effects of the pH/pCO<sub>2</sub> control method on medium chemistry and phytoplankton growth, *Biogeosciences* 6 (7) (2009) 1199–1207, <https://doi.org/10.5194/bg-6-1199-2009>.
- [22] K.G. Schulz, J.B. Ramos, R.E. Zeebe, U. Riebesell, CO<sub>2</sub> perturbation experiments: Similarities and differences between dissolved inorganic carbon and total alkalinity manipulations, *Biogeosciences* 6 (10) (2009) 2145–2153, <https://doi.org/10.5194/bg-6-2145-2009>.