PolyPonics Water Quality Monitoring System for Aquaculture and Aquaponics

A User’s Guide for Microcomputing in the Classroom

Mark IV Version

MDSG Publication Number UM-SG-ES-2021-01

The Maryland Sea Grant College communications team provided copyediting and graphic design for this publication.

PolyPonics Water Quality Monitoring System for Aquaculture and Aquaponics

A User's Guide for Microcomputing in the Classroom

Mark IV Version

System Design and Authors
Harry Berman, Glen Hedelson, Jeff Reeser, and William Wolfe
Baltimore Polytechnic Institute High School
1400 West Cold Spring Lane
Baltimore, Maryland 21209

Editors
J. Adam Frederick, Glen Hedelson, and Jacqueline Takacs
Maryland Sea Grant and University of Maryland Extension
University of Maryland College Park
College Park, Maryland 20740

CONTENTS

APPENDICES

INTRODUCTION

History of the PolyPonics Water Quality Monitoring System

The PolyPonics System (PPS) project was conceived at Baltimore Polytechnic Institute (BPI) High School with two goals in mind: first, to provide a way to ensure the success of a water quality monitoring system for aquaculture/aquaponics, and second, to provide a vehicle to introduce the students to a real-world systems engineering and software development project.

BPI instructors William Wolfe and Jeff Reeser put together the original aquaculture lab at BPI, and Harry Berman wrote and developed the PPS water quality monitoring software on a Linux platform using open-source development tools and software. The PPS has been undergoing refinements, expansions, and updates in cooperation with J. Adam Frederick of Maryland Sea Grant College at University of Maryland.

Figure 1. Aquaponics Lab at Baltimore Polytechnic Institute High School in Baltimore City, Maryland.

When students take water quality measurements one parameter at a time, many time-consuming and expensive chemical tests must be conducted. These tests produce no visible trends unless a large number of data points are taken and then plotted by hand.

With real-time 24-hour graphic plots, the PPS will present students with a clear picture of water quality trends in recirculating systems and will allow students to make inferences about water quality parameters and their relationships to each other, i.e., temperature and dissolved oxygen.

The PPS provides real-time water quality monitoring for aquaculture/aquaponics projects and transmits data via Wi-Fi.

The PPS software is copyrighted © by the BPI High School under the Source License, GNU General Public License.

This software and the Vernier® sensors are designed for educational use. These products are not designed or recommended for any industrial, medical, or commercial process such as life support, patient diagnosis, control of a manufacturing process, or industrial testing of any kind.

I.

General Hardware Description

The Raspberry Pi 4B (RPi 4B) is a credit card-sized microcomputer. The RPi 4B has 1 GB of RAM memory, a graphics card that is capable of video playback, a solid-state memory chip slot, four USB ports (two USB 2.0 and two USB 3.0 connectors), and Wi-Fi capability.

There are three of these microcomputers in the PolyPonics System (PPS). The first RPi 4B, acting as the client, is connected to the Arduino, protoboard, and various water testing sensors. The second RPi 4B, acting as the server, is connected to a solid-state hard drive. The server allows the system's data to be displayed on a monitor and downloaded as Excel spreadsheets. A third RPi 4B will act as a web server in conjunction with a separate monitor to display water quality data and video through a Wi-Fi connection.

The Arduino Mega 2560 converts the water testing sensor data from analog form to digital form and sends that data through the client RPi 4B to the server.

The Mark IV version of the PPS can support nine different Vernier® water quality sensors. Each of those sensors is connected to a protoboard through BTA connectors, which are then connected to the Arduino by means of color-coded bits of wire.

Wi-Fi capability allows the client and server to be at some distance from each other.

The Mark IV system uses a 128 GB Fit 3.1 USB flash drive for data storage. The Fit flash drive is fast, simple to install, and agile for data transfer.

The RPi 4B increases processor speed, multimedia performance, memory, and connectivity compared to the RPi 3 Model B+. The RPi 4B retains backwards compatibility, which means software written for RPi 3B+ will work on this latest version.

Note: *For this manual we are using the Raspberry Pi 4B (RPi 4B). All future mention of RPi references this model unless otherwise noted.*

II.

Hardware Components

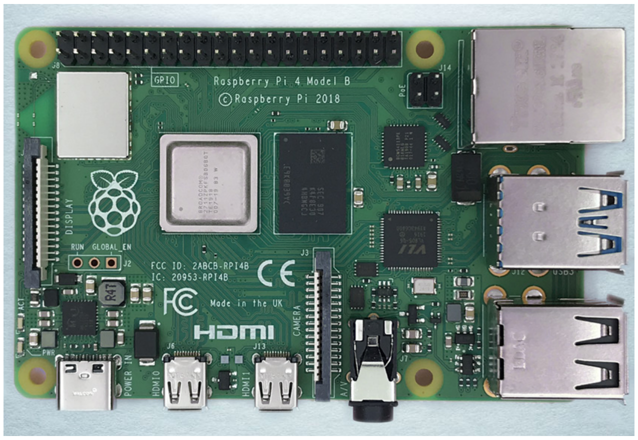


Figure 2. Raspberry Pi (RPi)

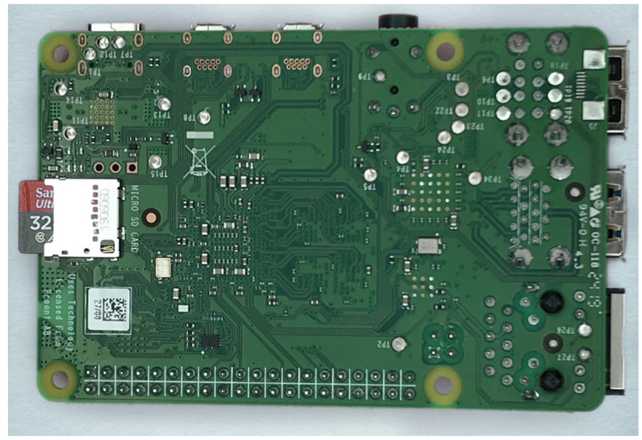


Figure 3. RPi with SanDisk 32 GB microSD card inserted, bottom view

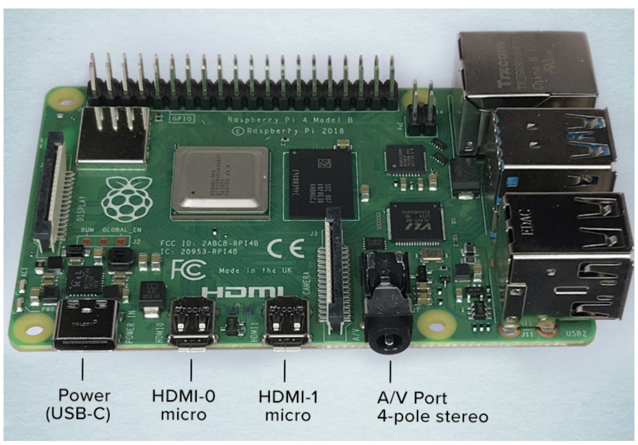


Figure 4. RPi microcomputer ports, oblique view

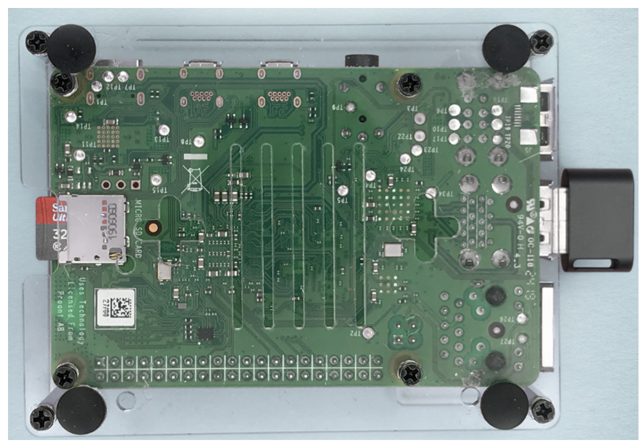


Figure 5. RPi in acrylic case with USB Fit and 32 GB microSD card inserted

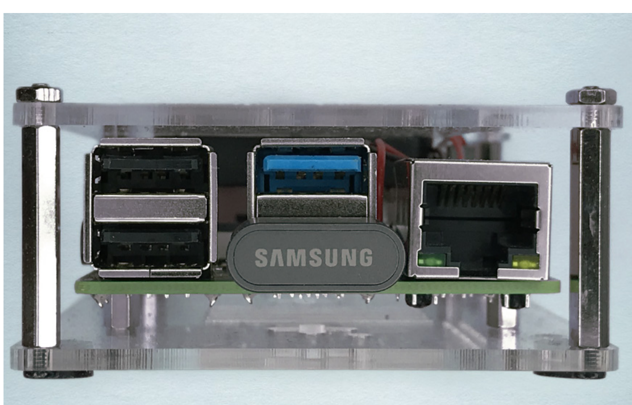


Figure 6. RPi end view, USB 2 ports (black) and USB 3 ports (blue) with USB Fit inserted

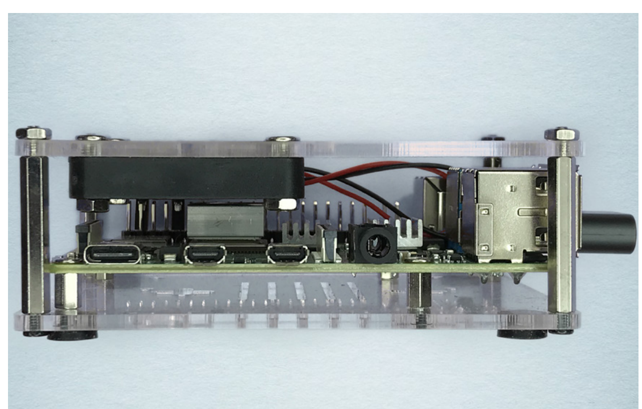


Figure 7. RPi assembled side view with ports

Figure 8. RPi power supply

Figure 9. RPi cooling fan

Figure 10. Two examples of a USB Fit 128 GB flash drive (3.0 or faster preferred) to use as the data storage device

Figure 11. RPi acrylic case hardware

Figure 12. Acrylic case standoff hardware

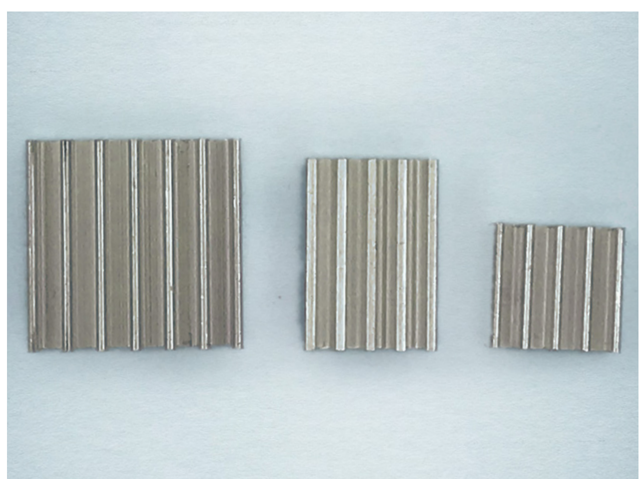


Figure 13. RPi heat sinks

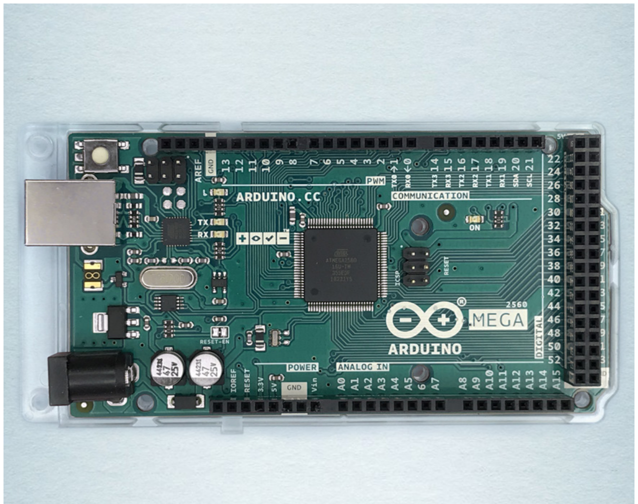


Figure 14. Arduino Mega 2560 microcontroller

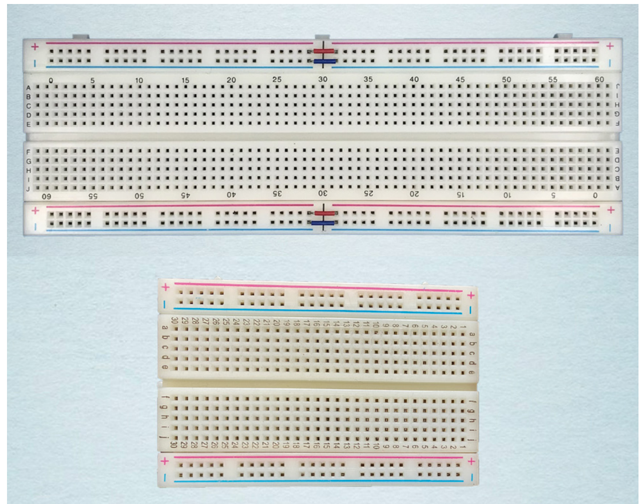


Figure 15. Long protoboard (top) and short protoboard (bottom)

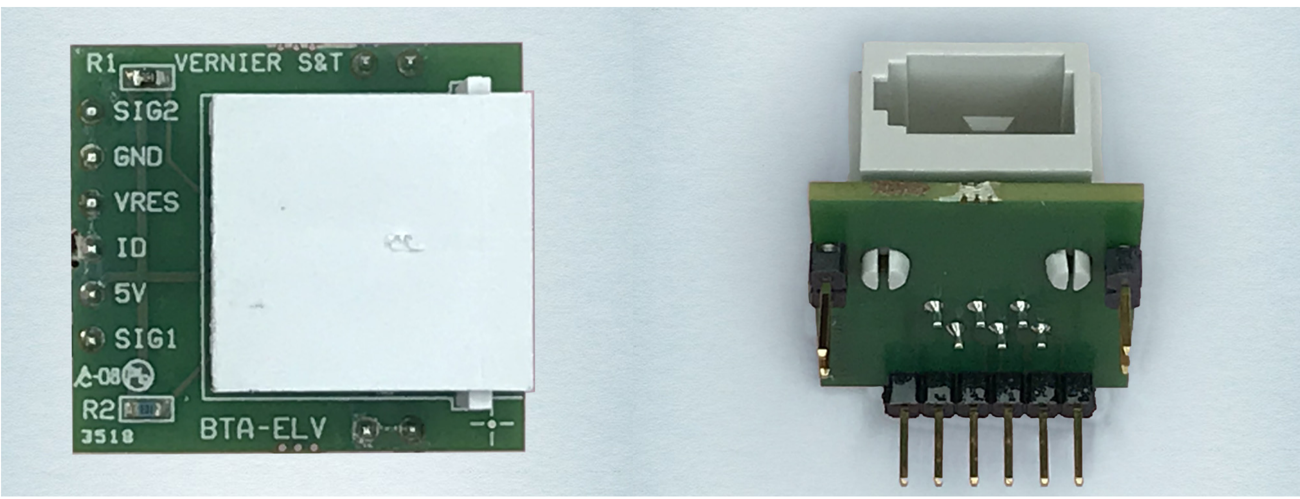


Figure 16. BTA connectors for Vernier water monitoring sensors, top view (l) and bottom view (r)

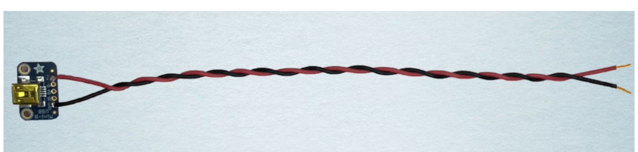


Figure 17. 5-volt breakout board soldered to wires to bring power to protoboard and Arduino

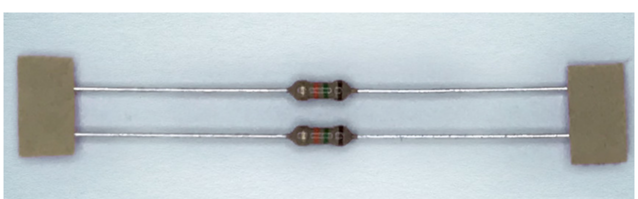


Figure 18. 15K ohm resistors for temperature BTA connections on the protoboard

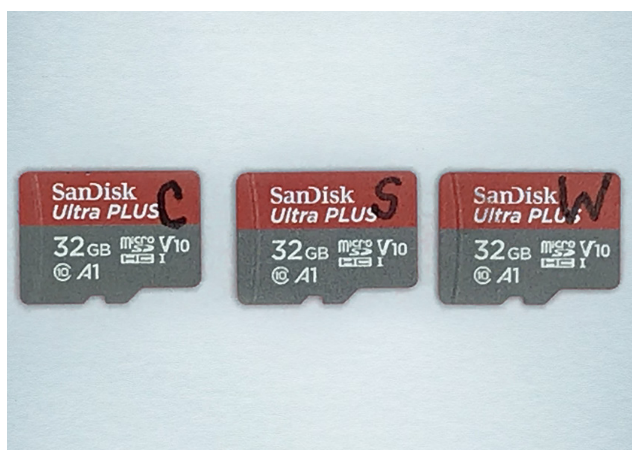


Figure 19. Client(C), server (S), and web server (W) microSD cards

III.

Sequence of Data Flow Through Hardware

1. Vernier sensor in or above the fish tank
2. BTA connector
3. Protoboard
4. Arduino Mega 2560
5. Client Raspberry Pi (RPi)
6. Wi-Fi (to web server RPi and monitor)
7. Server RPi
    a. Fit flash drive
    b. Monitor
    c. User

Note: *See "Appendix 6: Diagram of Data Path" on page 38.*

Wi-Fi Capabilities of the PolyPonics System

The server can be in a remote location because the system is connected by Wi-Fi. The remote location is often a classroom where data can be monitored in real time. The client or server also can support a Logitech video camera to visually monitor the aquaponics room. The web server will display the live Wi-Fi data feed to another monitor or to a television placed within range of the Wi-Fi signal.

IV.

PolyPonics System Components

The assembly of the PolyPonics System (PPS) is an easy process but requires attention to detail and sequence. Once assembled, the system can be powered up, software can be installed, components initialized and formatted, and water monitoring can commence. The components needed to build the basic PPS include:

3	Raspberry Pi (RPi) computers with power supplies, cooling fan, and case kit
1	Arduino Mega 2560 microcontroller
1	5-volt power breakout board soldered to red and black connector wires for Arduino power
1	5-volt power supply with micro USB connector for Arduino power
2	Protoboards: one long (830 pt) and optional one short (420 pt)
3	32 GB microSD chip loaded with the PPS program software code
1	USB Fit flash drive 128 GB to act as solid-state drive for system

Hardware

2	Clear, acrylic clipboards to use as a system base and cover
1	Piece acrylic 1” x 9” for BTA hold down
4	10–24 x 2.5” machine screws
12	10–24 stop nuts with nylon insert
8	#10 x 1” fender washers
~3 ft.	Red, black, and yellow 22-gauge hookup wire (precut and stripped)
~1 ft.	3M Dual Lock mounting tape to connect components to the acrylic base

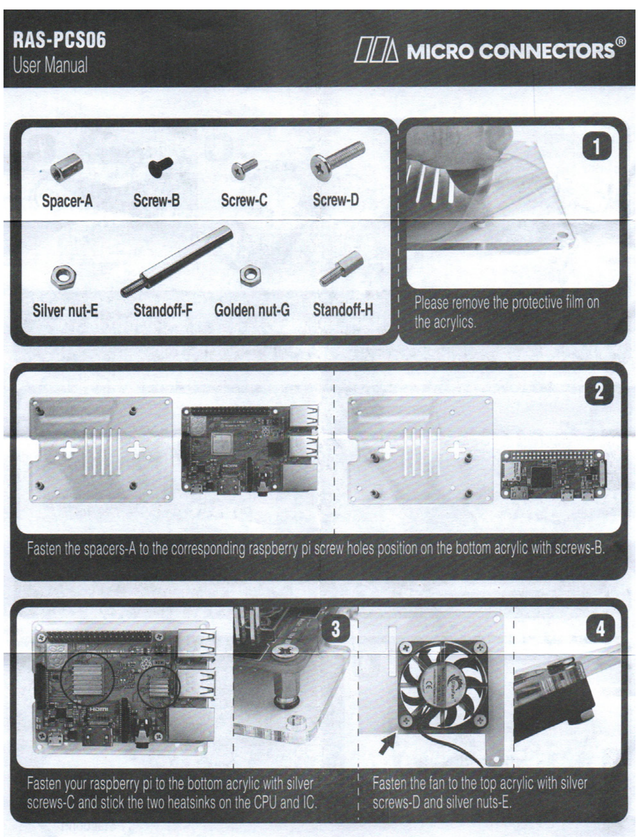
V.

Assembling the Client

Unpack the box containing the acrylic stackable Raspberry Pi (RPi) case kit with USB-C power. Account for all of the component parts. Follow the directions carefully to assemble the case and fan and install the RPi.

Directions for RPi Case and Fan Assembly

1. Remove the protective film from the acrylics.
2. Fasten the spacers-A to the corresponding RPi screw holes on the acrylic base with screws-B.
3. Fasten the RPi to the bottom acrylic with silver screws-C and stick the two heat sinks on the CPU and IC.
4. Fasten the fan to the top acrylic with silver screws-D and silver nuts-E.
5. Fasten the standoffs-F to the bottom acrylic with screws-B.
6. Install top acrylic by putting the standoffs-F into the top acrylic screw holes and fixing with golden nuts-G.
7. Install the red pin of fan on GPIO-4 (5V) and black pin on GPIO-6 (GND). Other 5V and GND GPIO sockets will work on the client RPi if the Arduino uses these sockets. Consult the GPIO diagram for other available sockets.
8. Step 8 in the photo directions is not needed.
9. Step 9 in the photo directions is not needed.
10. Insert the microSD card “C” into the client RPi. Take care not to touch the gold contacts. Insert the card with contacts facing up and label facing down.



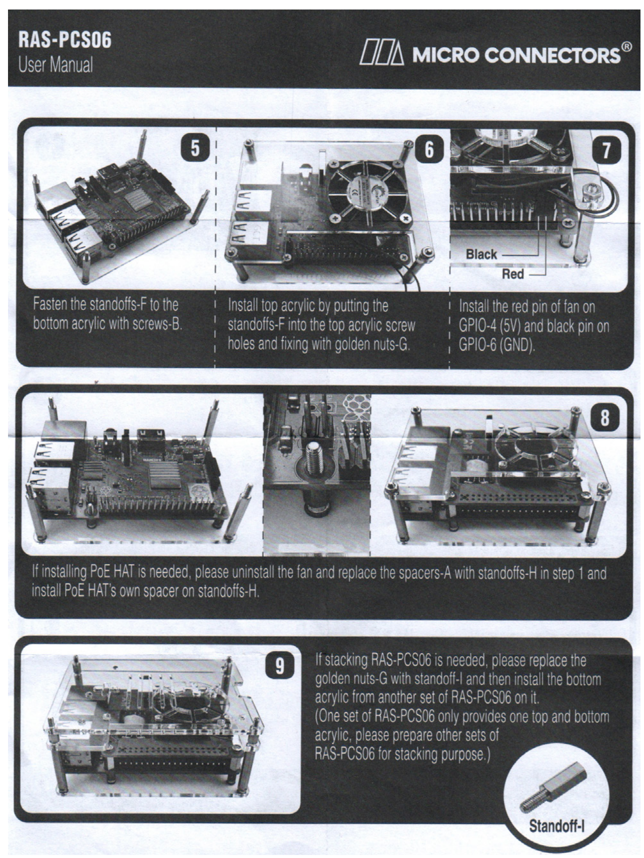


Figure 20. Photo directions for RPi case and fan assembly

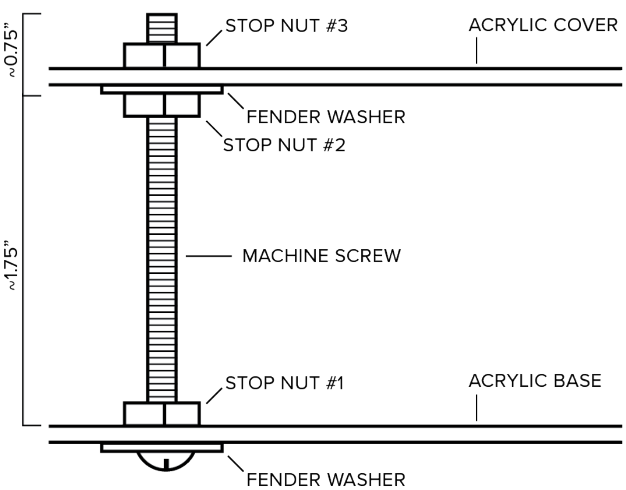
Client Acrylic Base and Cover

The client RPi, due to its function, is in close proximity to aquaria and the potential for splashing and other water intrusion. Because of this, we recommend some level of waterproof protection for the client hardware. Here, we describe how to prepare and mount the client's components in a base and cover that will protect them from water splashes, stabilize them to ensure wiring integrity, and allow adequate RPi ventilation—the primary features of any such base configuration. This method uses clear acrylic clipboards that are readily available, the right size, and reasonably priced. You can also choose other methods; one PPS user has modified a shallow, lidded plastic storage box for all of the PPS client components and mounted the entire client unit vertically near the fish tank. Custom-cut Plexiglas can also work, though it is more expensive and harder to get than the acrylic clipboards. While the server RPi and web server RPi are remote from wet environments, PPS users may want to build protective covers for these components as well using this same method. Directions and hardware quantities below are for a single client assembly.

Acrylic Base and Cover Hardware

4	10–24 x 2.5" machine screws
12	10–24 stop nuts with nylon insert
8	#10 x 1" fender washers

Hardware sequence



Preparing the Acrylic Base and Cover

Locate two acrylic clipboards. You will remove the metal clips on these boards, exposing two holes at one end, and you will drill two more holes at the other end. These four holes will serve as attachments points for the acrylic base and cover.

To remove the metal clips, take the clipboards to a workstation that has an electric drill. Insert a 3/16" bit in the chuck. Place one clipboard on a backer board of scrap lumber to prevent drilling through to the table. Center the drill bit over one of the rivets holding the clip and drill through the clipboard and rivet. Repeat for the other rivet and remove the clip. Then repeat for the second board.

If you cannot drill out the rivets, the board clip will need to be removed with a band saw. To do this, put masking tape on both sides of the clipboards as close as possible to the clip. This will prevent chipping when the acrylic is sawed. Use a sliding miter gauge to insure a straight saw cut.

Once you remove the metal clips, you will drill two more holes at the opposite end of the acrylic boards. On one board, measure and mark two holes: 1” from the end and 1.75” from each side. Use a low-adhesive tape (like blue painter’s tape) to secure both acrylic boards together and drill both at the same time. Use the backer boards and do not apply too much pressure as the acrylic can crack.

You now have an acrylic base and top plate for your system.

Assembling the Acrylic Base and Cover

Locate the 2.5” machine screws, fender washers, and nylon-lined stop nuts in the hardware package. Place one washer onto each machine screw and insert one screw through one of the holes in one of acrylic boards (the base) you have just drilled. Attach a nylon lock nut to the machine screw. Use a wrench and screwdriver to screw it all the way down to the acrylic board. Snug tight is good enough.

Screw the second nylon lock nut onto the machine screw until it is about three-quarters of an inch down from the top of the machine screw (there should be about 1.75” on the screw threads between the acrylic base and the top of this stop nut). (See “Hardware sequence” on this page.)

Repeat this entire process for the other three machine screws. You now have the base with four screws projecting upward, and four stop nuts that will hold the cover once it is installed.

Before installing the cover, you will be attaching the hardware components to the base. Go to Appendix 3 on page 28, and use the template to trace each component's location on the acrylic base. Once these components of the PPS are connected, you are ready to install the cover. Screw a fender washer onto the top of each stop nut. Slide the second acrylic board (the cover) onto the four screws. Install a final stop nut on each screw and tighten until snug.

Mounting the Protoboard

Locate the acrylic base and the long protoboard. Locate the traced location for the protoboard on the acrylic base.

***Note:** The blue, negative bus will be closest to the edge of the acrylic base. The red, positive bus will be oriented toward the interior of the base.*

Remove the protective film from the adhesive backing of the protoboard. ***Caution: Once the adhesive touches the acrylic base, it will be attached. There is no adjustment!***

Carefully position the protoboard at the location marks and fasten it to the base (Figure 21)

Mounting the Arduino

Locate and unbox the Arduino (Figure 22). Locate the 3M Dual Lock tape. Cut eight small tabs of tape for the bottom corners of the Arduino. Stick the tabs together in pairs.

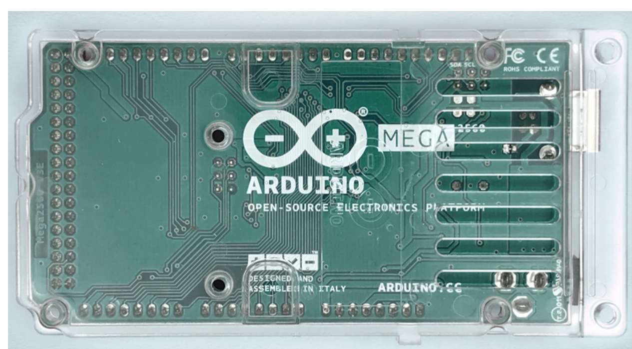


Figure 22. Arduino Mega 2560 microcontroller

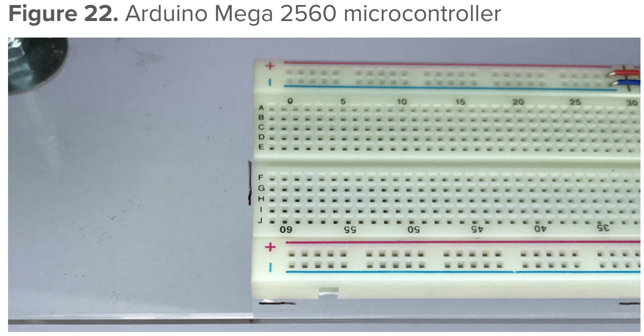


Figure 21. Protoboard measured, marked, and installed on the acrylic base

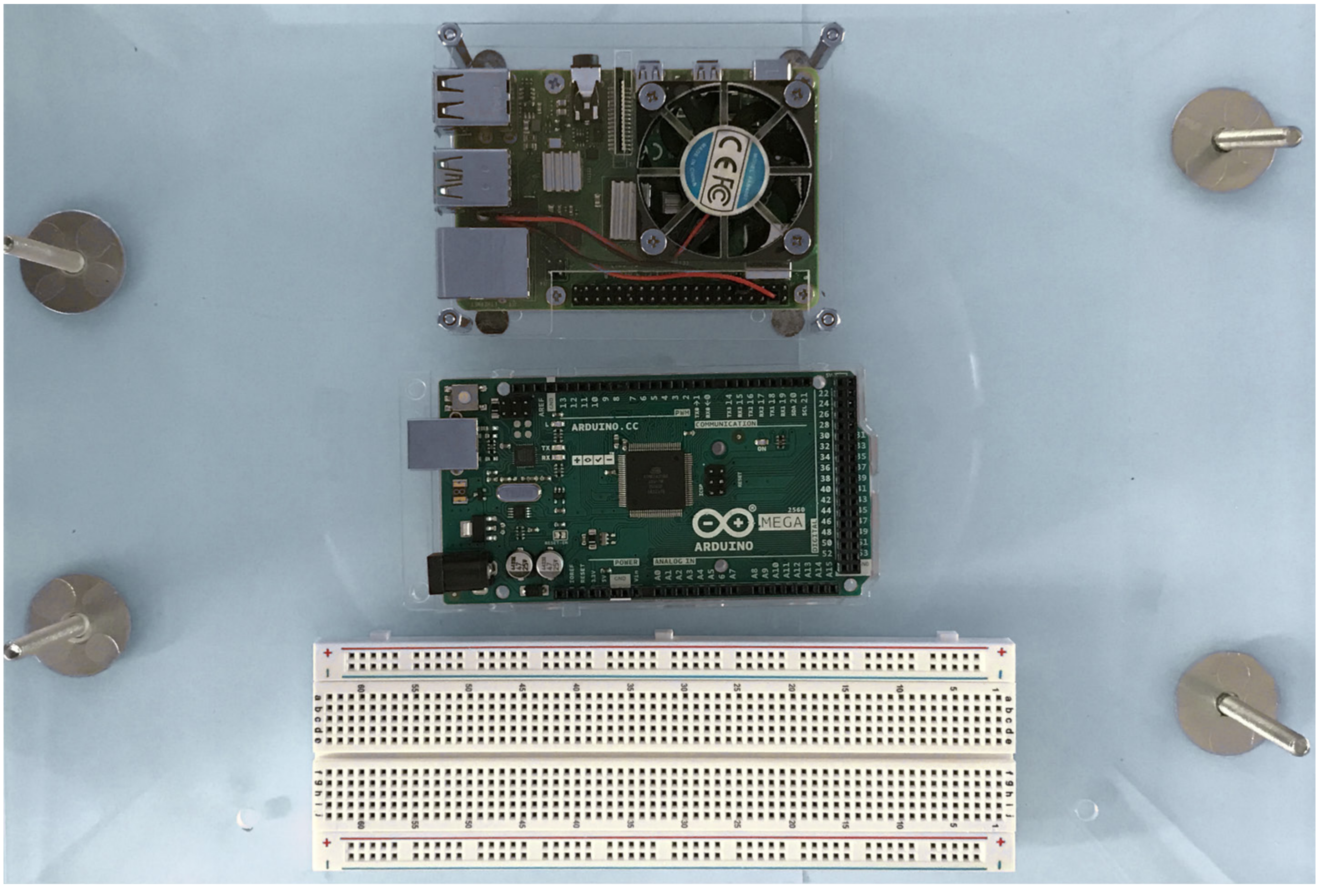


Figure 23. All three components of the client (protoboard, Arduino, and client RPi) properly mounted to the acrylic base

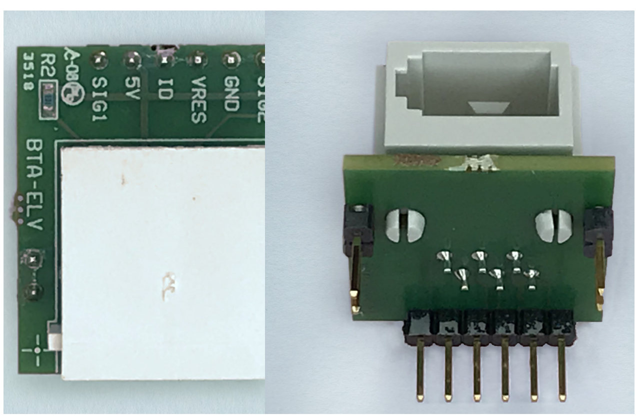


Figure 24. BTA connector, top view (l) and bottom view (r)

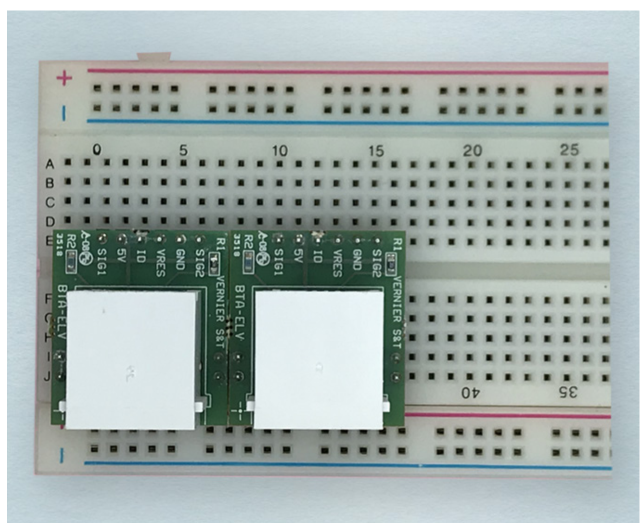


Figure 25. Two BTA connectors installed on protoboard

Remove the protective film and attach the tape to the underside of the Arduino (avoid the ventilation slots).

Locate the traced template location for the Arduino. Remove the other protective film from the 3M Dual Lock tape at the bottom of the Arduino and fix the Arduino to the acrylic base.

Mounting the Client

Assemble a RPi (see page 7) with its cooling fan and case. This will become the client RPi.

Locate the microSD cards. Remove the card labeled “C” and insert the microSD card into the slot with contacts facing up.

Remove the film from the adhesive of 3M Dual Lock tape and attach the tape to the bottom corners of the client RPi case. Cover the existing black rubber feet.

Locate the traced template for the client RPi. Remove the protective film and center the client RPi over the template with the USB ports facing to your left. Remove the protective film and adhere the client RPi to the base. (Figure 23)

Installing BTA Connectors on the Protoboard

Locate the BTA connectors (Figure 24). One at a time, remove the pink protective foam. There is frequently a bit of residual molding material on the edges of the green BTA board. Use either a fine file or strips of sandpaper held flat on a tabletop to abrade those bits of material from the edges.

The long protoboard will hold seven BTA connectors.The pins toward the opening of the BTAs will be inserted in protoboard rows I and J, and the other pins should insert in row E. Begin at the left-most edge of the protoboard and insert the BTAs in a line (Figure 25).

Wiring the Protoboard/Arduino Power Connection

The Arduino will be powered by a single 5-volt power supply. That power supply will be connected through the 5-volt power connector. The 5-volt power connector will have a red and a black hookup wire soldered in place. The wires need to be twisted about one another to reduce interference (Figure 26). The wires should be bent so that they lay flat along the acrylic base and extend from the attachment location to column 25 of the protoboard.

Figure 26. 5-volt power supply that connects to (+) and (-) sockets on the protoboard

***Note:** The red wire always connects to the positive and the black wire to the negative sockets on the protoboard!*

Mounting the Client’s 5-volt Power Connector

Cut two pieces of 3M Dual Lock tape, stick them together, peel off the protective tape from one side and attach the tape to the bottom of the power connector. Peel off the other protective tape and fasten the 5-volt power connector to the base between the machine screws on the left. Attach the 5-volt power connector to the protoboard by inserting the black wire into the blue, negative socket near the center of the protoboard. Insert the red wire into the red, positive socket near the center of the protoboard (Figure 27a).

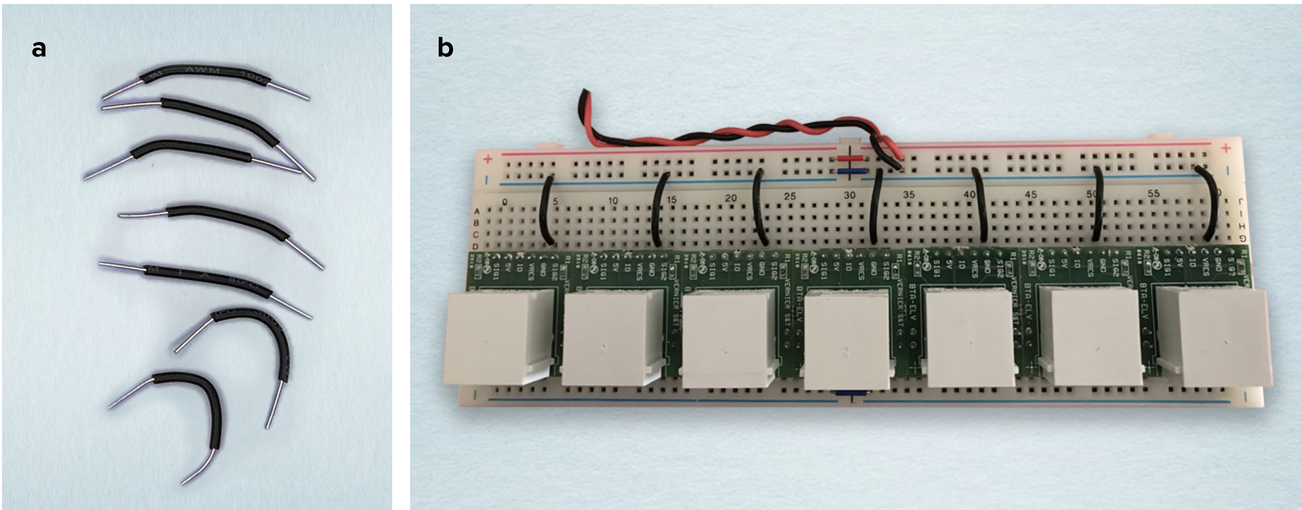


Figure 27. a) Black connecting wires **b)** Black connecting wires properly inserted into the protoboard with a full set of BTA connectors

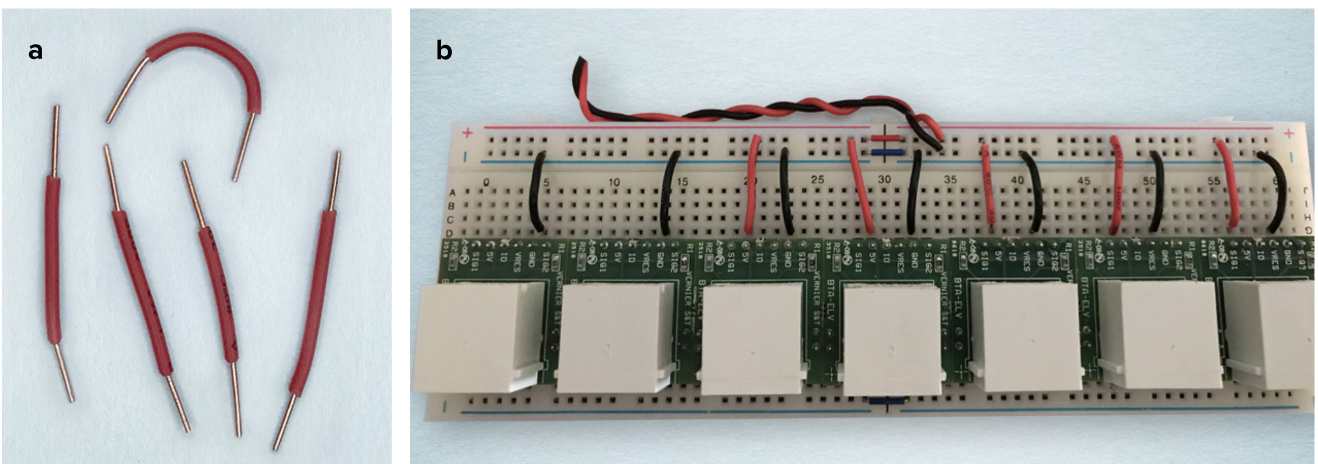


Figure 28. a) Red connecting wires that connect the 5V sockets of BTA to the + bus on the protoboard **b)** Red and black and 5V power connector wires properly installed

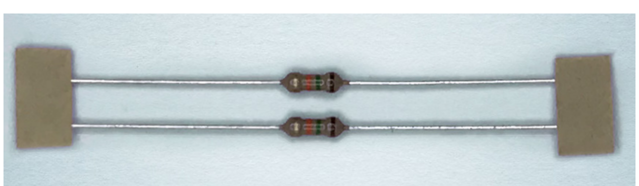


Figure 29. 15K ohm resistors that connect the two temperature BTAs: SIG1 to + bus on the protoboard

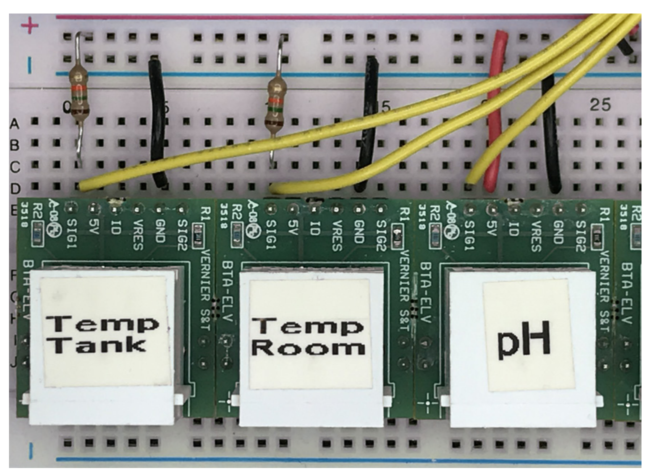


Figure 30. 15K ohm resistors in place on the protoboard for the temperature sensor BTAs

Wiring the BTA Connectors to the Protoboard

It is time to begin wiring the BTA connectors to the protoboard. The black and red connecting wires have been pre-cut and stripped for this workshop. From your equipment container, remove all seven of the black connecting wires (Figure 27a).

Note: *Black and red connecting wires should be 4 cm long.*

These are the ground (GND) wires for each BTA connection. On each BTA connector, locate the GND pin (second pin from the right of the BTA). Insert one end of the black wire into the protoboard column aligned with GND at row D.

Insert the other end of that black wire into the corresponding column in the negative (blue) row at the edge of the protoboard. Repeat this with all seven black wires for all seven BTA connectors (Figure 27b).

Note: ***If installing fewer than seven BTAs, the long protoboard can still be wired for future BTA installation. The black wire will be inserted into the ninth socket to the right of the preceding black wire.***

Locate and remove the five pieces of red hookup wire from your equipment container (Figure 28a).

Starting at the right end of the protoboard, insert one end of the red wire into the socket aligned with the “5-volt” pin (second pin from the left on each BTA). The other end of each red wire is inserted into the corresponding column in the positive (red) row at the edge of the protoboard (Figure 28b). Note that the red wires will not be used with the first two BTAs but can be installed in the third socket to the left of the black wires.

Connect Resistors to Temperature BTA Connectors

The two BTA connectors at the left end of the protoboard (temp-air and temp-tank) will be connected from the SIG1 socket in row D to the positive (red) row with 15K ohm resistors. The colored stripe pattern to denote a 15K ohm resistor is brown, green, orange, gold (Figure 29).

Locate two of these resistors. Cut both wires’ leads for a total length of 4 cm. Carefully bend the wire ends into a wide “U” shape. Insert one end into the SIG1 bus in row C or D to the corresponding socket in the positive (red) row. Do this for both temp-room and temp-tank (Figure 30).

Connect Power from Arduino to the Protoboard

There are two 5-cm wires, red and black, in your container. They have a small bit of blue tape holding them together. Remove the blue tape and discard it. These two wires will bring the 5-volt power from the protoboard to the Arduino.

Locate the power bus on the Arduino. The red wire is inserted into the 5-volt socket and the black wire is inserted into the GND socket in the power bus. Insert the other ends of these two wires into the appropriate (+) and (-) sockets of the protoboard at about socket 26 (see Figure 31).

Wiring Protoboard to Arduino Analog-In Bus

The data from the sensors pass through the BTA connectors into the protoboard, which is connected to the Arduino by yellow hookup wire. The approximate lengths of the wires needed are listed here:

Temp (tank)	12 cm
Temp (room)	10 cm
pH	8 cm
ODO	7 cm
Flow	7 cm
Light	8 cm
Conductivity	10 cm

Measure the distance from SIG1 socket to the appropriate location on the Arduino analog-in bus to double check these numbers. Two centimeters have been added to account for the stripped ends of the wires.

Use the stripping pliers to remove ~1 cm of insulation from each end of each yellow wire. This procedure may have been done for you.

Use yellow wire you just cut and stripped to connect the BTAs on the protoboard to the Arduino analog-in bus (Figure 31).

Temp-tank SIG1 connects to Arduino AØ
Temp-room SIG1 connects to Arduino A1
pH SIG1 connects to Arduino A2
ODO SIG1 connects to ArduinoA3

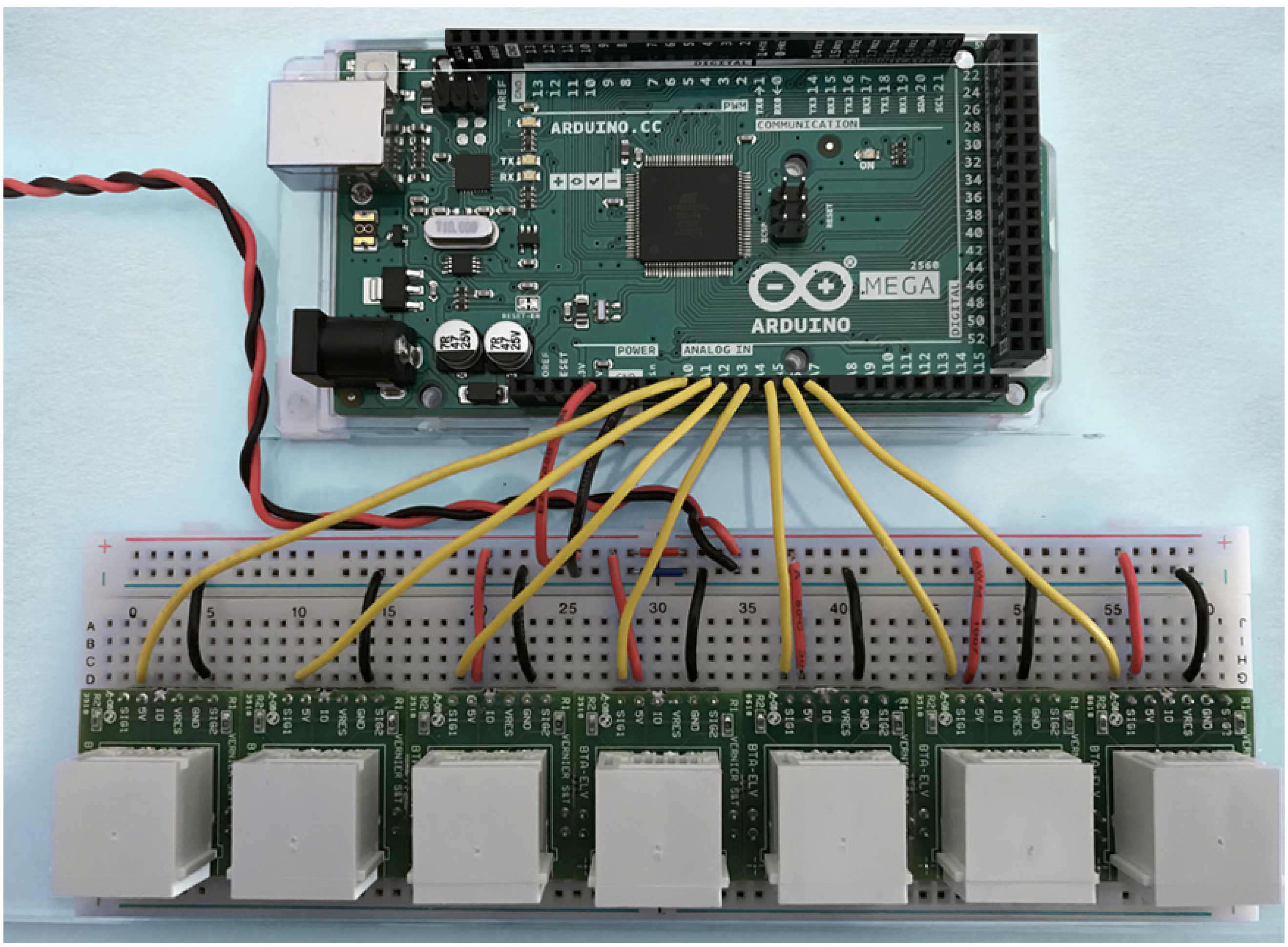


Figure 31. Long protoboard BTA connectors properly wired. Note the red and black wires from protoboard sockets #25 and #26 to the 5V and GND sockets of the power bus on the Arduino.

Flow SIG1 connects to Arduino A4
Light SIG1 connects to Arduino A5
Conductivity SIG1 connects to Arduino A6

***Note:** If assembly involves fewer than seven sensors/ BTA connectors, the position of each connection to the Arduino is prescribed. Do not make any connections other than those listed here. If you do not use a flow or light sensor, those BTA connections and Arduino wiring positions will remain unused.*

***Note:** The yellow wire position on the protoboard is one socket to the left of the red wire (Figure 32).*

If you are planning on using nitrate and/or ammonium sensors, continue with this section. If you're not, skip ahead to the next section "Connecting the Arduino to the Client RPi General Purpose Input/Output Bus (GPIO)"

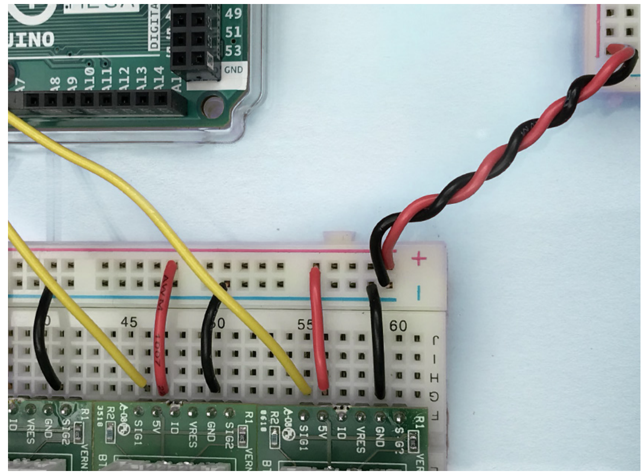


Figure 32. Twisted power connector from long protoboard to the short protoboard

Install the short protoboard on the template to the right of the long protoboard. Peel the protective covering off of the adhesive and place the short protoboard on the acrylic base. If BTA connecters are provided, insert the BTA connectors as you did with the long protoboard. Measure, cut, strip, and twist a red and black set of jumper wires to connect the (+) and (–) bus on the long protoboard to the (+) and (–) bus on the short protoboard. Use the end sockets in each protoboard for this connection (Figure 32). If ORP and Flow sensors are to be used, cut a red and black wire for each. Strip each wire and install it as you did with the BTAs on the long protoboard. Yellow wires (measured, cut, and stripped) connect as indicated below:

Nitrate SIG1 connects to Arduino A7
Ammonium SIG1 connects to Arduino A8

Connecting the Arduino to the Client General Purpose Input/Output Bus (GPIO)

From the parts container, locate the three colored connector wires that have a pin on one end and a socket on the other end. Select a black jumper wire and any two other colors.

The black jumper wire will connect the Arduino bus socket Digital GND to the client RPi pin #6 (or #9 or #14 GND if fan GND is in #6). Consult photo of RPi pin numbering system in the manual (Figure 34).

Choose one of the other colors you selected. The pin is inserted in the Arduino bus socket, SLC 21, and the socket end is connected to client RPi pin #5, I2C1 SCL (Figure 35).

The remaining colored jumper wire will be connected from Arduino socket, SDA 20, to the client RPi pin #3, I2C1 SDA. ***Note:** Pins #1 and #2 are not connected to any wire (Figure 35).*

The client assembly is now complete.

Client Location in Your Lab

Try to locate your client with acrylic base and cover where it won't be subject to water splash. Sensor extensions wires are available from Vernier. Alternative methods of housing the client assembly (e.g., acrylic Sterilite box) are possible to prevent water damage.

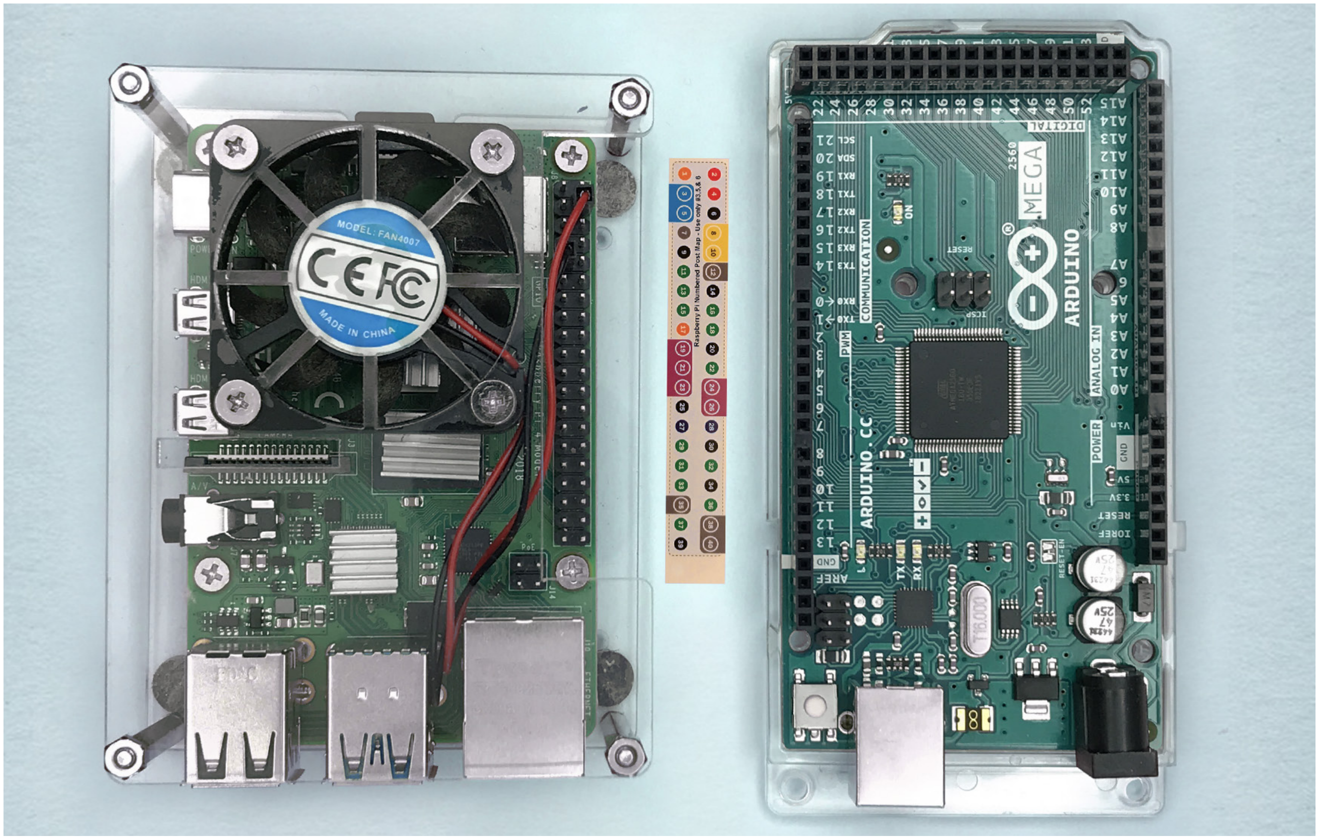


Figure 33. Connection map for wiring Arduino to the server RPi

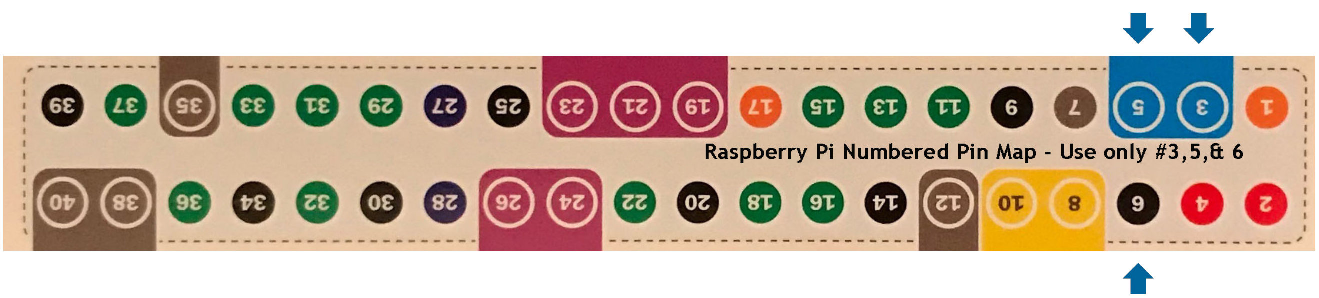


Figure 34. The Arduino socket #20 connects to Pi pin #3; Arduino socket #21 connects to RPi pin #5; GND connects to pin #6; pin #5 is fan red wire; pin #9 is fan GND.

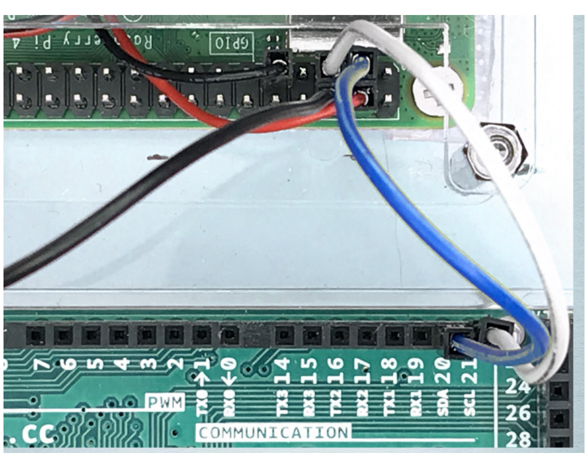


Figure 35. Close up view of connection between Arduino communication bus socket #20, #21, and GND and pins #3, #5, and #6 (GND) of the GPIO bus of the server RPi. The red wire is fan on pin #4, the thin black wire is fan on pin #9 (GND), the white wire is Arduino SCL 21 to GPIO pin #5, the blue wire is Arduino SCL 20 to GPIO pin #3, the thick black wire is Arduino digital (GND) to GPIO pin #6, and GPIO pins #1, 2, 7, 8, and 10 are empty. (See connection map, Figure 34)

VI.

Assembling the Server

Assemble the server Raspberry Pi (RPi) as you did the client RPi (page 7). Once assembled, insert the microSD card labeled “S” into the server RPi.

Take care not to touch the gold contacts. Insert the microSD card with contacts facing up and label facing down (Figure 36).

Connecting Server to Keyboard, Monitor, Mouse, and Client

Connect the keyboard and mouse to the server RPi via USB 2.0 jacks. Connect the server to the back of the monitor with a HDMI-to-micro-HDMI cable. Adapter plugs may be needed depending on the connections on your individual monitors.

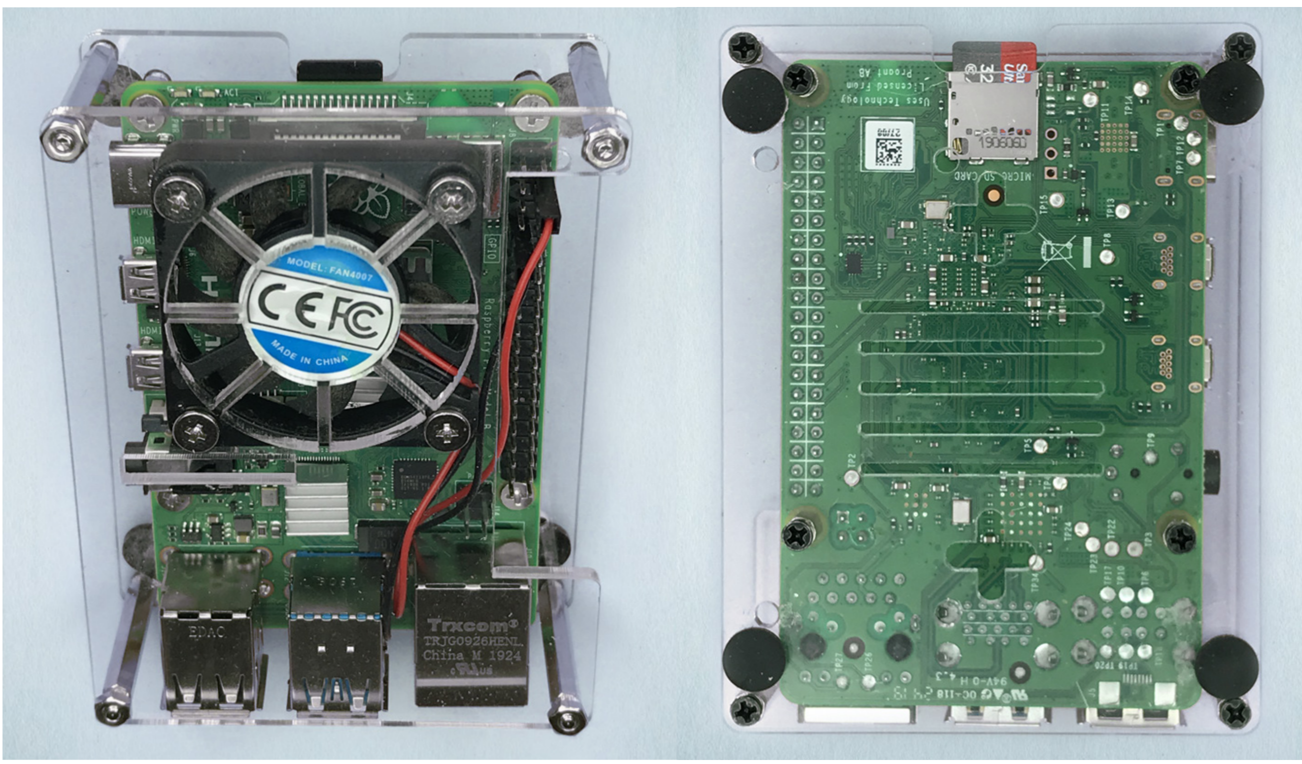


Figure 36. RPi with microSD card inserted, top view (l) and bottom view (r)

VII.

Assembling the Web Server

Assemble the web server RPi as you did the server RPi. Once assembled, insert the microSD card labeled "W" into the web server RPi. Connect the web server to any monitor or TV with a micro-HDMI cable. Connect the 5-volt power supply to the web server RPi and plug monitor/TV and RPi into a power source. Power up the system.

Note: *The PolyPonics (PPS) server must be up and running before adding power to the web server.*

VIII.

Connecting the PolyPonics System to Power

Note: *No Vernier sensors should be connected to the PolyPonics System (PPS) during the initial connection to power.*

Connect the multi-outlet power strip to an electric outlet. Be certain the power strip switch is in the “Off” position.

Connect the micro-USB plug of a 5-volt power supply to the client 5-volt power connector that is attached to the client’s acrylic base. Plug it into the power strip.

Connect the micro-USB plug of another 5-volt power supply to the client RPi and plug it into the power strip. Connect the micro-USB plug of another 5-volt power supply to the server RPi and plug it into the power strip.

Plug the monitor power cord into the back of the monitor and into the power supply.

Powering Up

1. Move the power strip switch to “On.”
2. Turn on the monitor.
   The PPS is powered up and running.
3. Click on time icon.
4. Enter the day, month, date, and time (24-hour).
   Note: *Day and month are three-letter abbreviations.*
5. Click “Enter.”
6. Click “Quit.”

The PPS is nearly ready for use, but first the Fit must be formatted and the database must be initialized (IX: Formatting the Fit Flash Drive and Initializing the Data Base, page 19), the software must be loaded on the Arduino Mega 2560 (X: Installing Arduino Software Through the Client, page 20), and active Vernier sensors must be selected (XI: Selecting the Active Sensors Connected to the Client, page 21). Once this is complete, sensors can be plugged into the BTA connectors on the protoboard.

IX.

Formatting the Fit Flash Drive and Initializing the Data Base

Click on the terminal icon in the desktop window. You will now enter the command scripts that identify and format the Fit USB Flash Drive to the server RPi and initialize the data base.

Identify and Format the Fit USB Flash Drive (Hard Drive)

Follow the steps below:
1. cd /home/pi (Click "Enter")
2. sudo ./FdiskSda1.sh (Click "Enter")
3. sudo ./PiDriveFormat.sh (Click "Enter")
4. sudo ./DiskLabel.sh (Click "Enter")
5. sudo reboot (Click "Enter")

***Note:** For steps 2–5, the format is "sudo [SPACE] [PERIOD] [FORWARD SLASH] Command Script")*

When system has restarted:
1. cd /home/pi (Click "Enter")
2. sudo ./FmtExHd.sh (Click "Enter")

***Note:** The format is "sudo [SPACE] [PERIOD] [FORWARD SLASH] Command Script")*

Initialize the Data Base

***Note:** This **must be** done the first time when the system is installed. This function can also be used any time to clear the data base.*
1. sudo ./create
2. sudo ./create_ext_drive_dba (Click "Enter")

***Note:** This **must be** the last script entered in this process.*

3. sudo ./ChownChmod.sh (Click "Enter")
4. Close terminal window by clicking on "X"

***Note:** For steps 1–3, the format is "sudo [SPACE] [PERIOD] [FORWARD SLASH] Command Script")*

X.

Installing Arduino Software Through the Client

This installation need only occur once at initial system start up. It can be accomplished using the client Raspberry Pi (RPi) or by clicking on the client icon on the server main page and entering the password. The hardware needed includes a keyboard, mouse, monitor, and USB-B to USB-C cable (a typical printer connection cable).

Preliminary Client Configuration

1. Power is on to client RPi.
2. Unplug power to the Arduino Mega 2560 at the 5-volt breakout board.
3. Unplug all Vernier sensors from the protoboard.
4. Connect monitor, keyboard, and mouse to the client RPi or use the client icon on the server.
5. Power up the Arduino Mega 2560 by plugging the power cord back into the 5-volt breakout board.

Arduino Mega 2560 Software Install

***Note:** This procedure is only needed one time when the system is initially installed.*

1. On the client desktop, double click on the Arduino IDE icon.
2. A window should open with "Arduino_client.ino."
3. If it does, skip to Step 8.
4. If file "Arduino_client.ino" is not displayed, continue to Step 5.
5. Select "File."
6. Select "Open Recent."
7. Select "arduino_client.ino."
8. Be certain that all Vernier sensors are unplugged.
9. Plug the Arduino Mega 2560 (USB-C) into the client RPi USB-B socket.
10. Select "Tools."
11. Select Board "Arduino/Genuino Mega or Mega 2560."
12. In the Board Manager window, be sure that "Arduino/Genuino Mega or Mega 2560" is selected.
13. Select "Tools" again.
14. Select "Port."
15. Select "tty/ACM0" (Arduino/Genuino Mega or Mega 2560).
16. Select the "check mark" to compile code.
17. Select "➡" to upload the program.
18. When uploading is complete, unplug the interface cable connecting the Arduino to the client RPi.
19. Reconnect the monitor, keyboard, and mouse to the server RPi or close the client window on the server main screen.

XI.

Selecting the Active Sensors Connected to the Client

This will allow you to select only the Vernier sensors that are connected to the system.

1. On the main screen, click on the ActiveSensors icon.
2. Select the sensors you have connected to the client. (Click the button to change the color: green is active / red is inactive.)
3. Click “Save,” then click “Quit.”
4. Control C to exit the server window (if it is open).
5. Click on the Raspberry icon in the upper left corner of the screen.
6. Select the “Shutdown” icon then click “Reboot.” Plug in the sensors indicated as active. The server and GUI will now appear with only active sensor data visible.

XII.

System Operation

- Always close the server with Control C command while the server window is visible and active.
- Set Time format is: Three-letter abbreviation for day of the week and the month, time is 24-hour, seconds can be set as 00 or set exactly.
- Set Time is required every time power has been disconnected but not on reboot.
- Always use Raspberry icon to shut down the system before disconnecting power.
- Connect power to the server before connecting power to the web server.
- Graphs on GUI display will require 24-hour cycle to complete a full display.
- Graphs and dials on GUI will register 0 value if they are not selected as “active.”
- Select Vernier sensors as “active” before the sensors are added, otherwise a reboot will be needed.

XIII.

System Passwords

For cybersecurity considerations, passwords should not be shared with anyone other than the actual system operators.

Server
poly2020Server#! (note upper case S)

Client
poly2020user#! (all lower case)

Web server
poly2020user#! (all lower case)

APPENDIX 1

Raspberry Pi (RPi) Hardware Description

- The processor is a Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC@ 1.5GHz
- The built-in memory is 1GB (or 2GB or 4GB).
- Connectivity is 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE gigabit ethernet, 2x USB 3.0 ports, 2x USB 2.0 ports.
- GPIO is standard 40-pin GPIO header that is fully backwards-compatible with previous boards.
- The video and sound include: 2x micro-HDMI ports, 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port, 4-pole stereo audio and composite video port.
- Multimedia capabilities include H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), OpenGL ES, 3.0 graphics.
- One microSD card slot is available for loading operating system or data storage.
- Input power options include 5-volt DC via USB-C connector, 5-volt DC via GPIO header, or Power over Ethernet (PoE)-enabled, which requires a separate PoE HAT.
- The client RPi interface with Arduino will be same as RPi 3B+.
- The RPi uses a SanDisk or Samsung Fit 3.1 USB 128 GB Flask Drive for data storage.

APPENDIX 2

Software Backup Protocol

Materials needed:
3 microSD cards 32 GB (SanDisk is preferred brand)
1 SD/microSD card reader with male USB connection

Label the new microSD cards: “S” for server, “C” for client, “W” for web server.

Copying the Server MicroSD Card

1. Insert “S” SD card into the card reader.
2. Insert the USB card reader into the server Raspberry Pi (RPi).
3. Window opens “Removable Medium is Inserted.” Click “Cancel.”
4. Select the Raspberry Pi icon in the upper left corner of the window.
5. Select Accessories tab.
6. Select “SD Card Copier.”
7. In the “SD Card Copier” window:
   a. For “Copy From Device,” select SL32G (dev/mmcblk0).
      ***Note:** This is the microSD card already in the server RPi.*
   b. For “Copy To Device,” select the USB Card Reader.
      ***Note:** This would be your generic storage device “/dev/sdb” or “/dev/sda”*
8. Click “Start.”
   ***Note:** This process may take 8–10 minutes.*
9. Wait until “Copy Complete” message appears.
10. Remove the microSD card and store it a safe place.

Copying the Client MicroSD Card

***Note:** This process is slightly more involved.*
1. Insert the “C” microSD card into the card reader but do not insert it into the client RPi yet. ***Note:** Sequence is important! Only insert the card reader into the client RPi after signing into the client (Step 11).*
2. On the monitor’s main screen, double click on “Client” icon.
3. You will see a turquoise screen labeled “rdesktop – 192.168.4.3.” This is the Remote Desktop Viewer window.
   a. Login to PolyPonicsClient
   b. Session: Xorg
   c. Username: pi
   d. Password: poly2020user#!
   e. Click “OK”
4. Within the Remote Desktop Viewer window, select the terminal icon.
5. Type “xhost +” (Click “Enter”).
   ***Note:** The format is “xhost[SPACE]+”*
6. This will present a message “Access control disabled. Clients can connect from any host.”
7. Insert the microSD card reader into any client RPi USB port.
8. An authentication window will pop up, requesting: “Enter Password Again.” Enter password.
9. “Removable medium is inserted.” Click “Cancel.”
10. Within Remote Desktop Viewer window:
    a. Select “Raspberry icon”

b. Select “Accessories”
c. Select “SD Card Copier”

11. In the “SD Card Copier” window, fill in the following information:
    a. For “Copy From Device,” select SL32G (/dev/mmcblk0).
    ***Note:** This is the microSD card already in the client RPi.*
    b. For “Copy To Device,” select your generic storage device (/dev/sdb) or (/dev/sda).
    ***Note:** One of these is the card reader and the other is the actual microSD card (usually “/dev/sdb” but trial and error will tell which one will work).*
12. Click “Start” and wait until “Copy Completed” message appears. Click “OK.”
13. Remove the USB card reader.
14. Disconnect the Remote Desktop Viewer window by clicking on the disconnect icon (just to the right of the connect icon near the top of this window).
15. Close the terminal window by clicking on “X.”
16. Close the Remote Desktop Viewer window by clicking on “X.”
17. Remove the microSD card and store it a safe place.

Copying the Web Server MicroSD Card

When powering up the web server, sequence is important: 1) Power to the server (and client), and then 2) power to the web server.

1. You will need a blank SanDisk 32 GB microSD card. Label it “W.”
2. Remove the microSD card from the web server RPi.
3. Insert the blank microSD card into a microSD card adapter or into a USB card reader.
4. Insert the original “W” card into a microSD card adapter or into a USB card reader.
5. Insert both into a desktop or laptop computer.
6. Use a standard application to copy the original “W” card onto the blank “W” card. The blank microSD card may need to be “Quick Formatted.”
7. Place the newly copied “W” microSD card somewhere safe in case you need a duplicate.
8. Put the original “W” microSD card back in the web server RPi and return power to that system.

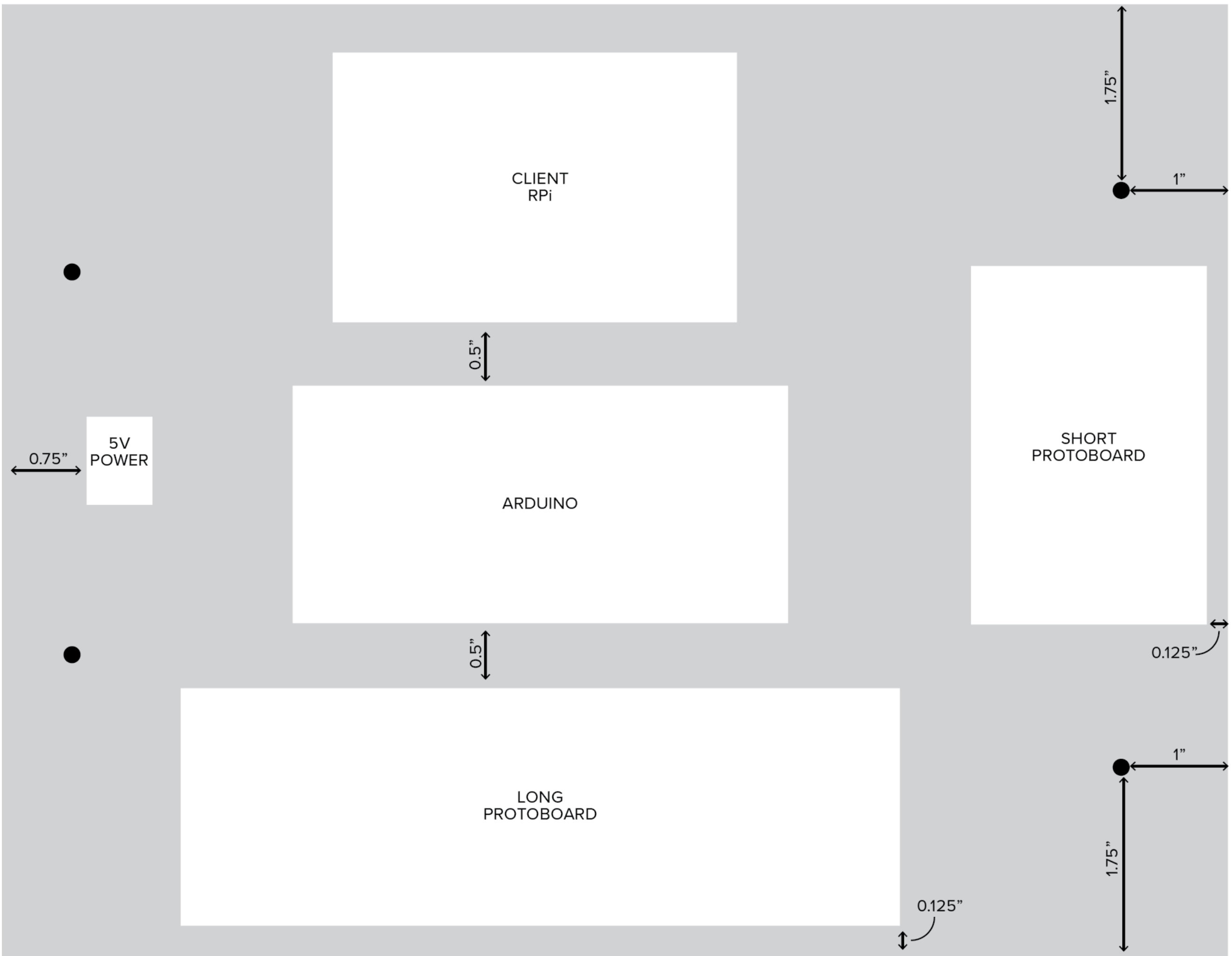
Changing the Server Names

1. Click on the terminal window.
2. Type “EditHostApd.”
3. In the new window, look for a code line “ssid=PolyPonics1.”
4. Type the new name for this server (e.g. ssid=PolyPonics2 or BAPonics3 or PolyHawks2, etc.).
5. Write down the new server name exactly because you will have to do the same for the client and the web server RPi.
6. Go to “File” and click “Save” and close window but do not reboot at this time.
7. Plug in a monitor, keyboard, and mouse to your client RPi.
8. Click on the terminal window.
9. Type “cd” to change the directory.
10. Type “sudo ./home/pi/PolyPi/EditWpaSupplicant.sh”.
    ***Note:** The format is “sudo[SPACE]./home/pi/PolyPi/EditWpaSupplicant.sh*
11. Locate the code line “ssid=PolyPonics1” and change the name to exactly the name you entered for the RPi server.
12. Go to “File,” click “Save,” and close window. Then, from the server, reboot the system.
13. Power up your web server RPi. Plug in a monitor, keyboard, and mouse.
14. Place the cursor in any black part of the screen.
15. Click “Terminal Emulator.”
16. Type “EditWpaSupplicant”.
17. Locate the code line “ssid=PolyPonics1.”

18. Change the name to exactly the name you entered for the RPi server.
19. Type "sudo /home/pi/BestWiFiChannel.sh" and note which channel is least used. Select that one.
    Note: *The format is "sudo[SPACE]/home/pi/BestWiFiChannel.sh*
    Note: *Select Channel=5 or least used available channel.*
20. Type "sudo /home/pi/EditHosstApd.sh" and enter the channel from above.
    Note: *The format is "sudo[SPACE]/home/pi/EditHosstApd.sh*
21. Go to "File," click "Save," and close window.
22. In Terminal window, type "sudo reboot" for the web server.

APPENDIX 3

Template for Client Acrylic Base



APPENDIX 4

Vernier Sensor Calibration Options

The Mark IV version of the PolyPonics System (PPS) can support nine different Vernier® analog sensors to measure water quality for aquaculture/aquaponics. For the initial setup of the PPS, all of the sensors will use the Vernier default slope-interface calibration values.

The PPS offers the user several alternative methods to calibrate the computed sensor values. Most of the calibration methods listed in Appendix 4 use Vernier's LabQuest, a handheld device that can interface with the sensors and is used for sensor calibration.

Below is a table displaying the sensor accuracy of Vernier default calibration values versus the PPS calibration values. Maintenance, use, calibration, and storage information for all Vernier sensors can be found at www.vernier.com/support/manuals.

***Note:** For best results, nitrate (NO_3), ammonium (NH_4), and pH sensors should be recalibrated once a month. Also, due to their sensitivity to micro-voltage electrical interference, ammonium and nitrate sensors should be positioned in the tank 12 inches away from the conductivity and ORP sensors, if possible.*

***Note:** LabQuest and LabQuest 2 were used to develop the photos and procedures for this appendix. They both have been replaced with LabQuest 3, the newest interface platform from Vernier. While we have not tested these procedures with LabQuest 3, the sensor calibration procedures should remain essentially unchanged.*

Sensor Default Versus User Calibration Values

Parameter	LabQuest (default)	PolyPonics Calibration	Delta	Percent Error
pH	7.57	7.69	0.12	1.58%
Dissolved oxygen (DO)	12.23 mg/L	11.763 mg/L	0.467 mg/L	3.8%
Conductivity	848.0 µS/cm	849.6 µS/cm	1.6 µS/cm	0.19%
NO_3	4.7 u	4.562 u	0.138 u	2.9%
Light	98.8 Lux	93.0 Lux	5.8 Lux	5.9%
Tank Temp.	27.6 °C	26.47 °C	1.13 °C	4.1%
Room Temp.	26.8 °C	25.63 °C	1.162 °C	4.4%

Sensor Specifications

The following are specifications regarding precision (reproducibility), accuracy, range, and calibration values for the various Vernier sensors used with the PPS.

Precision

NO_3 ± 10%
NH_4 ± 10%

Accuracy

Temperature: ± 0.2 °C at 0 °C, ± 0.5 °C at 100 °C
pH: ± 0.2 pH units
Optical DO: ± 0.2 mg/L if below 10 mg/l
± 0.4 mg/L if above 10 mg/l

Flow: ± 1% of full-scale reading
Conductivity: ±8% of full-scale reading for low range
±3% of full-scale reading for middle range
±4% of full-scale reading for high range

Light Intensity Sensor

Calibration: Slope/Intercept Values

Light Sensor Ranges
0–6000: slope = 1692 lux/v and intercept = 0 lux
0–600: slope = 154 lux/v and intercept = 0 lux
0–150000: slope = 38424 lux/v and intercept = 0 lux

Conductivity Sensor Ranges
0–20000: slope = 9000 µS/cm and intercept = 0
0–200: slope = 65.7 µS/cm and intercept = 0
0–2000: slope = 960 µS/cm and intercept = 0

Conductivity Selector

Factory Calibration Values

pH Sensor
slope = -3.838
intercept = 13.720

Vernier Optical DO (mg/l)
slope = 4.444
intercept = -0.4444

Flow Rate Sensor
slope = 1.0
intercept = 0

Frequency of Sensor Calibration

The Vernier sensors in the PPS were not necessarily designed for continuous operation as they are used in the PPS. However, experience at Baltimore Polytechnical Aquaculture Lab has shown that the sensors provide accurate data even when used over an extended period. Should any data set appear to be distinctly different from previous data in range or precision, an alternative method of sampling (e.g., titration) should be used to check on the accuracy of the sensor’s calibration.

Should the difference in readings be significant, then the sensor may need to be recalibrated. Bear in mind that the PPS is not designed to provide laboratory-quality data suitable for publication, but it will provide data that can be used to illustrate the complex relationships between the parameters calculated/measured and the organisms living in the tanks and biofilters.

PPS uses three categories of sensor from Vernier: linear sensors (pH, conductivity, light, and flow) where the amount of voltage is proportional to the level of the parameter being measured, non-linear sensors (temperature) where the sensor uses a thermistor and requires a three-point calibration, and ion-selective electrode sensors (ammonium and nitrate) where electrical potential in the water is being measured and logarithmically converted to concentration. The ion-selective electrode sensors should be recalibrated once each month. The ion-selective electrode sensors use a PVC membrane, which has a limited life expectancy. When calibrating these sensors, it is probably time to replace the membrane modules when significantly different voltages or voltage ranges are noticed after calibration.

PolyPonics Vernier Sensor Calibration

By default, the PPS is hard coded to use the Vernier sensor default calibration values for slope and intercept for linear calibrated sensors. These values can be obtained from the manufacturer's documentation.

As mentioned above, the PPS offers the user several methods to calibrate the computed sensor values. You might ask, why offer multiple means of calibrating if any one method will provide the same level of accuracy, precision, and resolution? The PPS was conceived and designed to provide a real-world application of STEM in the classroom. By demonstrating that there are many ways to approach a problem, this system adds insight to the science research and teaching environment.

For sensors that use linear calibrations, the voltage from the sensor varies in direct proportion with the sensor reading. Voltage can be converted into an appropriate measurement unit using the slope-intercept equation:

sensor reading = slope x voltage + intercept

where slope is the rate of change between the sensor reading and the voltage, and intercept is the offset (difference) of sensor value at zero volts.

The PPS calibration can use the LabQuest calibration results to provide the reference frame for linear calibrated sensor slope, intercept and base, and coefficient for ion-selective electrode sensors (i.e., NO_3 and NH_4). These parameters are obtained through the "Equation" function of the LabQuest.

The PPS calibration uses the Vernier-provided reference slope and intercept with a user-provided reference value. That value is obtained by connecting the sensor into the LabQuest and noting the reading. Reconnect the sensor back into the PPS client and enter that reading into the "New Value" field of the specific sensor GUI window. Click "Calibrate," then "Compute," then "Save," and then "Quit."

The calibrator will use the new value to perform a single-point calibration function that will recompute a new intercept. This will bring the PPS into alignment with the LabQuest. All future data readings will use this new calibration.

Alkalinity and carbon dioxide (CO_2) are the only water quality parameters that are not directly measured by the Vernier LabQuest. Instead, alkalinity is mathematically calculated by the PPS software from KH hardness data that the user enters daily for about 30 days. CO_2 is computed based on pH and carbonate hardness levels.

Sensor Calibration Procedure

Sensor Calibration Using Vernier's LabQuest Device

This section goes through various procedures in calibrating your sensors. Please note: the color of the screen in the photo indicates which piece of hardware is active.

- Gray screens are data entry GUI while logged onto the client (or server for nitrate and ammonium)
- Black uxterm screens are client sensor selection screens to select which sensor to calibrate.
- Dark blue with red are LabQuest screens with measured value of a sensor that is connected to LabQuest.

Note: *That value will be entered into the gray client data entry screens when the sensor is plugged back into PPS.*

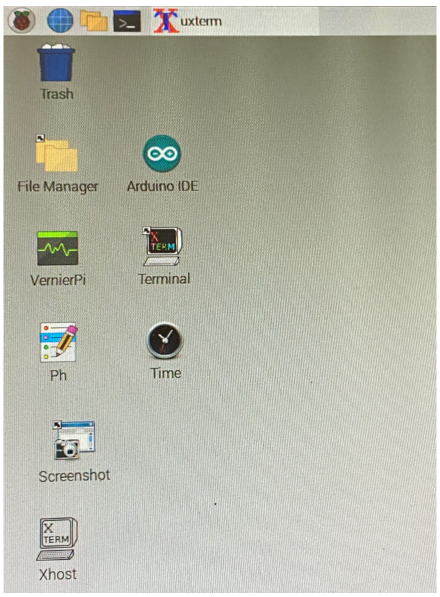
- Pale blue screens are also LabQuest. After clicking on the red sensor value on the screen, click on calibrate.

LabQuest's submenu at the top is "calibrate," "equation," "storage," "source," and "sensor info." Use the "equation" submenu to obtain the "pH slope/intercept data" or "nitrate/ammonium base/coefficient" data. This data will be entered into the gray "client data entry" screen (or gray "server data entry" screen for NO_3 and NH_4) when the sensor is reconnected to the PPS.

Log into the client through the server RPi

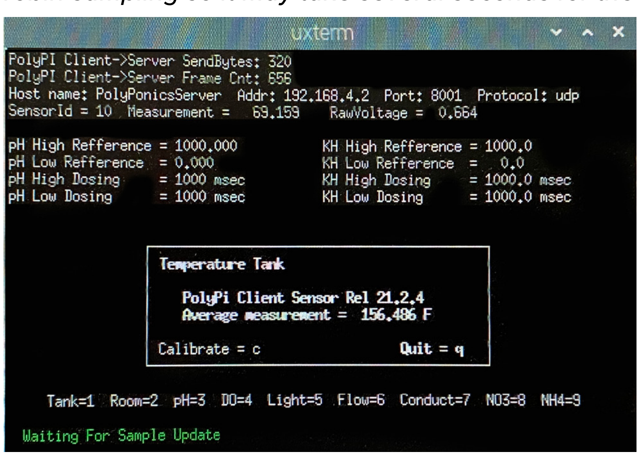
1. Log into the PPS client by clicking on the client icon on the server main page. Enter password (see Section XIII: System Passwords for a list of passwords).

2. You will see the client screen. Click on the Xhost icon, then click on the VernierPi icon.

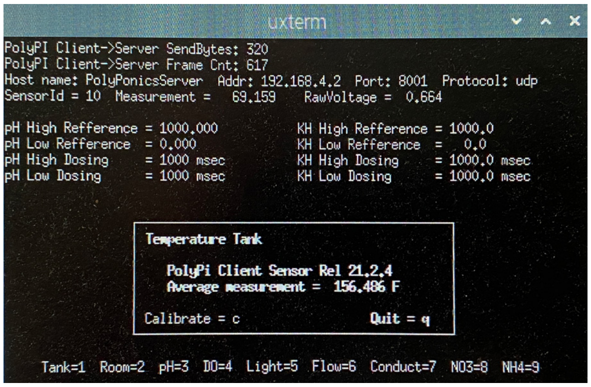


PPS client screen

3. The Calibration GUI (labeled "uxterm") will open.
Note: *By default, the GUI opens with "Waiting For Sample Update." The sensor data is processed by round-robin sampling so it may take several seconds for the data to appear.*



PPS calibration GUI, black screen with green "waiting" text

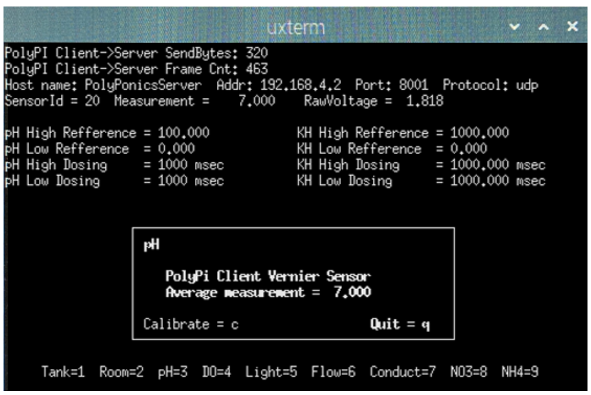


PPS calibration GUI. black screen with “Temperature Tank” sensor selected (default)

***Note:** The default sensor is “Temperature Tank.”*

4. Select which sensor to calibrate. For this example, we will use the “pH” sensor.

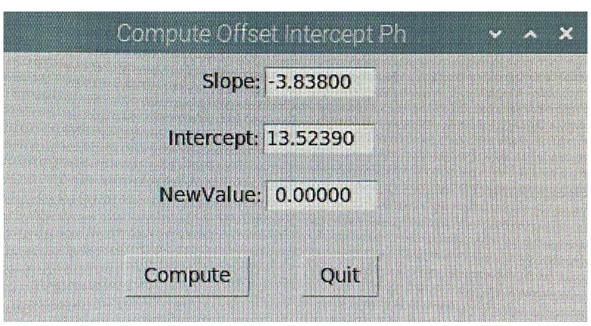
5. Press the number “3” on the keyboard. “Waiting For Sample Update” will be displayed.



PPS calibration GUI with “pH” selected

6. When the selected sensor data is displayed, press the letter “c” on the keyboard for “calibrate.” The text color will turn green until the entry GUI is displayed.

   After several seconds the calibration data entry GUI (labeled “Compute Offset Intercept Ph”) will be displayed.



PPS calibration data entry GUI for pH

   The display shows the active “Slope” and “Intercept.” NewValue = 0. (If “Quit” is pressed, the system will use these values for all computations.)

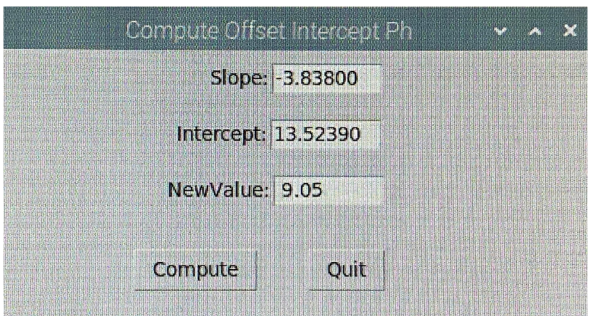
Calibrate using the LabQuest's measured value

7. Unplug the selected sensor from the PPS client protoboard and plug in to the LabQuest. Note the reading displayed on the LabQuest. ***Note:** For this example, we are measuring pH.*

8. Plug the sensor back into the PPS client.

LabQuest's pH display

9. In the PPS interface, enter the reading displayed on the LabQuest into the "NewValue" field.



PPS calibration data entry GUI for pH with LabQuest's pH value entered into "NewValue" field

10. Press "Compute," then "Quit."

    After several seconds the PPS will be in sync with the readings obtained with the LabQuest. This value will be used to compute a new intercept point.

Calibrate using slope and intercept values

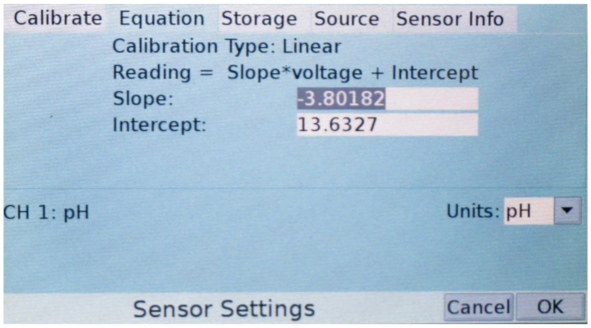
It is possible to use the computed values for slope and intercept obtained using LabQuest's "Equation" function to calibrate PPS.

Follow steps 1–6 starting on page 33 to log into the client through the server RPi.

With the sensor plugged into the LabQuest:

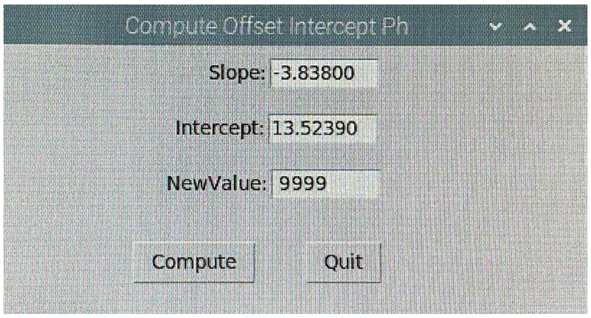
7. Click on the red value displayed on the screen, then click "Calibrate."

8. Select the "Equation" bar at the top of the window. Make note of the values.



LabQuest's pH equation screen with "Slope" and "Intercept" readings

9. Plug the sensor back into the PPS client.



PPS calibration data entry GUI for pH

10. Enter the LabQuest "Slope" and "Intercept" values and "9999" into the "NewValue" field.

11. Press "Compute," then "Quit."

Calibrate Linear Sensors: Fast Mode

***Note:** This method follows the calibration sequence "Calibrate Using Measured Value" on page 32.*

1. Log in to the PPS client (password = poly2020user#!).
2. Click on the Xhost icon.
3. Open an x terminal window.
4. Type "cd PolyPi" then press "Enter."
5. Type "cd SensorCalibration" then press "Enter"
6. The user must select the sensor to calibrate by typing the appropriate command. ***Note:** The format is "[PERIOD][FORWARD SLASH]Calibrate_____.sh" then "Enter."*
   They are:
   ./CalibratePh.sh
   ./CalibrateCond.sh
   ./CalibrateDo.sh
   ./CalibrateFlow.sh
   ./CalibrateLux.sh

The client calibration window will be displayed. Connect the selected sensor to the LabQuest.

7. Click on the red window.
8. Click on "Calibrate."
9. Click on "Storage."
***Note:** Each linear sensor has a tiny microchip incorporated in the sensor. Vernier puts in the "default value" when it is assembled. If a user wants to create a new/specific calibration to use in a particular situation, that new calibration will be called "calibration1" or anything the user wishes, i.e., tilapia1 or trouttank2, etc.*
10. Click on "OK." This stores the new calibration in the sensor which will then be transferred to the PPS, and those new calibration numbers will travel with the sensors to the PPS.
11. Plug the sensor back into the PPS.

Calibrate Linear Sensors: Even Faster Mode

1. Log in to the PPS client (password = poly2020user#!).
2. Click on the Xhost icon.
3. Click on Raspberry icon in the upper left corner of the "rdesktop-192.168.4.3" window.
4. Select "System Tools."
5. Select the sensor you wish to calibrate and enter the LabQuest information for that sensor—either "NewValue" or "Slope" and "Intercept."
6. Click "Compute," then "Quit." In a few seconds your sensor will have a new calibrated value.

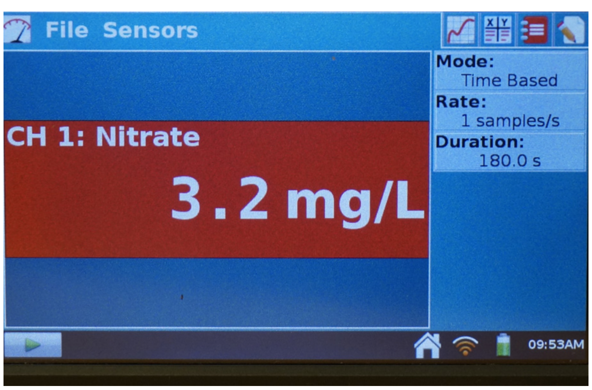
Calibrate NO_3 and NH_4 Using Base and Coefficient Values

***Note:** Ion-selective electrodes nitrate (NO_3) and ammonium (NH_4) are calibrated from the server RPi.*

Calibrating NO_3 or NH_4 for PPS use requires that you first calibrate each sensor following the Vernier LabQuest calibration procedure.

Initial sensor calibration with LabQuest

1. Insert the sensor into the LabQuest. Copy down the LabQuest displayed data value. (i.e., Nitrate: 1.0)



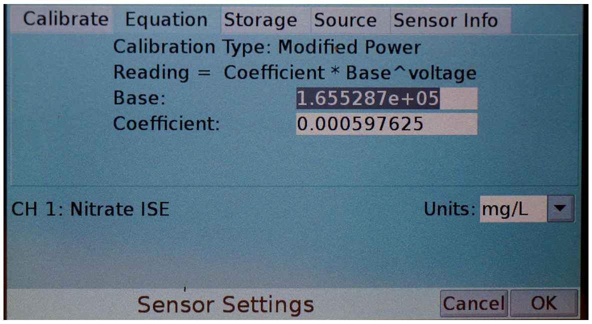
LabQuest's nitrate display

2. Calibrate using the device calibration procedure supplied by Vernier for your particular LabQuest device. Click anywhere in the red sensor value window.

3. Select "Equation" from the submenu at the top of the window.

4. Note the "Base" and "Coefficient" values from the display.
***Note:** These values will be needed for entry into the server calibration of that selected sensor.*

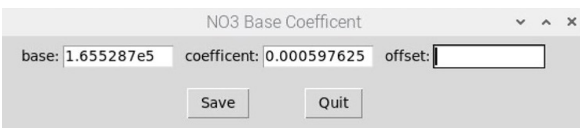
5. Connect the sensor back into the PPS.



LabQuest's nitrate equation screen with "Base" and "Coefficient" values

Calibrating NO_3 and NH_4 sensors for PPS

6. On the PPS server, open the GUI window.
7. Select "Cal NO_3" or "Cal NH_4."
   Note: *The data entry window will be displayed.*



PPS data entry screen with "Base" and "Coefficient" fields

8. Enter the values that were obtained during the LabQuest calibration into the "Base" and "Coefficient" fields of the data entry window.
9. Press "Save" and then "Quit."

After a short time, the NO_3 or NH_4 values will be updated. If there is a difference between the LabQuest reading and the PPS readings, enter the difference in the "offset" field, i.e.:

LabQuest = 9.08
PolyPonics = 10.3
offset = LQ–PP (9.08–10.3)
offset = -1.22

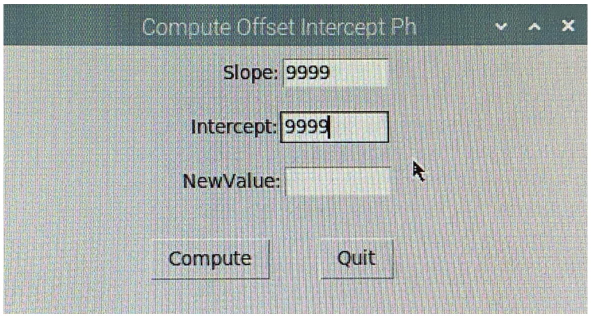
Note: *It is recommended that NO_3 and NH_4 sensors be recalibrated once per month.*

Restore Sensor Default Values

The LabQuest values will now become the active data. If you want to restore the default Vernier sensor hard-coded calibration values for slope and intercept obtained from the manufacturer's documentation:

1. Enter "9999" in the "Slope" field.
2. Enter "9999" in the "Intercept" field.
3. Leave the "NewValue" field blank.

4. Press “Compute” and then press “Quit.”



PPS data entry screen

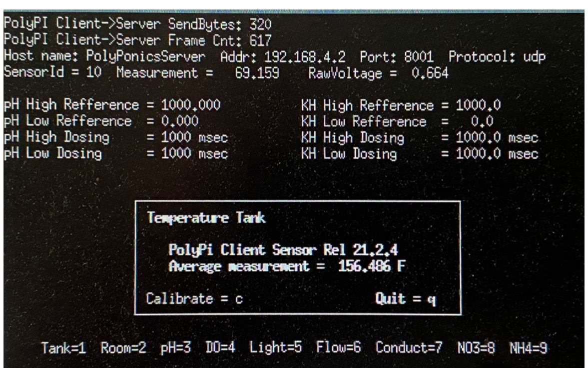
The sensor default data will be restored.

Non-Linear Calibration Method

Temperature Sensor Calibration

***Note:** Calibrating the temperature sensor is only needed when the system is installed or if the temperature readings are not within an acceptable range. This procedure is not necessary if your temperature readings are within acceptable range.*

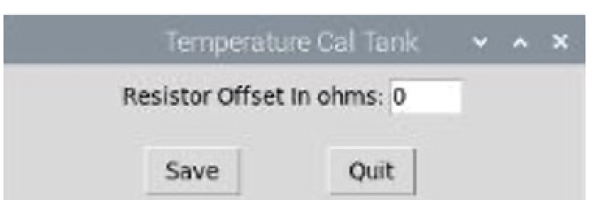
1. Log into the client. Press “1” for “Temperature Tank” or “2” for “Temperature Room,” depending on the sensor you would like to calibrate. A “Waiting For Sample Update” message will appear.
   ***Note:** Might take up to 10 seconds for sensor data to display.*



PPS calibration GUI with “Temperature Tank” selected

2. When the selected sensor data is displayed, press the letter “c” on the keyboard for “Calibrate.” The text “Waiting For Sample Update” will turn green until the data entry GUI appears.

3. After several seconds, the “Temperature Cal Tank” GUI or “Room” GUI will appear.



PPS “Temperature Cal Tank” GUI; to adjust temperature sensor, the user will enter either +1000 (to make temperature value lower) or -1000 (to make temp value higher)

4. Measure the tank or room temperature with a digital thermometer. Note the reading.

5. You will manually calibrate the temperature sensors by entering these values into "Resistor Offset in ohms" window: 1000 ohms = approximately 1 °F.
   Note: *The values have an inverse effect on the temperature reading. Entering (-)1000 ohms will increase temperature and (+)1000 ohms will decrease the temperature reading.*
   For example, if the PPS temperature reading is three degrees higher than the digital thermometer, you will want to lower the PPS reading to match it. To do this, enter (+)3000 in the "Resistor Offset in ohms;" this should reduce the PPS reading by three degrees to match the digital thermometer reading.
   Users may also enter fractions of (+ or -) 500 or 250 to fine-tune the system. Sometimes, this calibration takes a few tries and some experimentation with the number of ohms entered.

6. Press "Save" and then "Quit."

The new values will now be used to compute the temperature.

APPENDIX 5

Oversampling in the PolyPonics System

Oversampling, a type of mathematical modeling, can help produce better resolution of data (e.g. enable smoother lines in graphical analysis), improving the accuracy and interpretation of the data. A conversion of analog to digital data (binary) combined with oversampling can help the Arduino Mega 2560 improve the lines of resolution of the data in the PolyPonics System (PPS). The Arduino is the microcontroller in the PPS that performs the work in the conversion of analog to digital data, and it's also the workhorse for interpreting and converting the data coming from the water quality probes and then transferring that information to the Raspberry Pi microcomputer.

The default hardware resolution of the Arduino for analog to digital conversion is 10 binary bits (or a conversion to a series of 1 and 0 or "on" and "off"). Ten bits in binary is represented as 1024. The value of 10 bits is an additive function of the following values or bits (1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 210) and totals 1024. In binary, counting can start at 0 instead of 1. Therefore, the count starting at 0 will total 1023 (0, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 210). The maximum for our 10-bit Arduino is 1024.

To explain the conversion of analog to digital to discreet bits starting with 20, consider the following example below. Starting from the right positional notation, the values illustrate a conversion (or equivalent):

power of 2s	23	22	21	20
digital value	8	4	2	1
bits	0	0	1	0

Total count of significant digits is 2

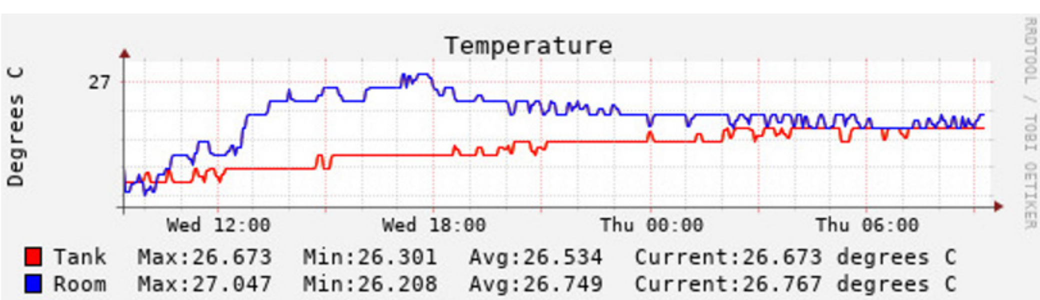
The DC supply voltage (Vc) of the Arduino is essential to the conversion from analog to digital. Five volts is the operating DC voltage of the Arduino. The measurement, noted as Vc, needs to be converted to millivolts (mV) in the conversion from analog to digital. For the calculation to mV, the binary count starts at 0 to include all the possible combinations of bits. The conversion can be represented by the following two-part calculation:

Part I. Vc/1023 = 5 V/1023 = 0.00488759 V = 4.88759 mV

Part 2. For the conversion to digital value, use the 4.88759 mV x digital value.

Example of a complete calculation:

5 V/1023 = 0.00488759 V = 4.88759 mV
4.88759 mV x 2 = 9.77 mV = 0.00977518 Vc



The graph above represents plotting 10-bit data for temperature. Notice the stair-stepping in the graph; this is a result of the Arduino capacity at 10-bit resolution.

Shifting Binary Numbers

Shifting binary numbers right or left is equivalent to multiplying or dividing integers. Shifting by N bits to the left on an unsigned binary number has the effect of multiplying it by 2n (if N =1 then shift left by 1 bit is multiplying by 2).

Shifting by N bits to the right on an unsigned binary number has the effect of dividing it by 2n (N=1 then shifting right divides by 2).

Question: *What would changing to a 12-bit digital sample do to our sampling accuracy? (12 bits = 4096)*

Answer: *This gives a resolution of (5 V/4095 = 1.221001 mV) x (digital value).*

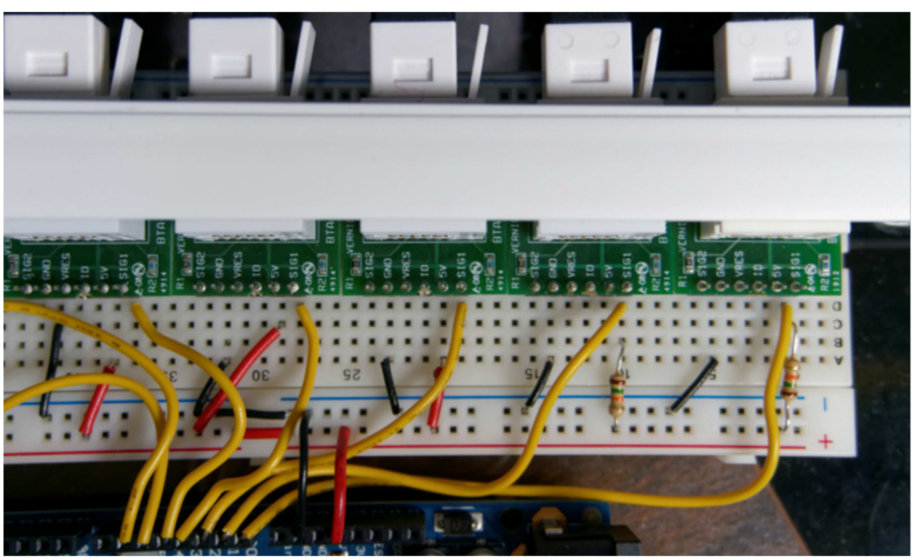
Each digital binary bit = 0.00122100 V. This is a 25% increase in the sampling resolution compared to the Arduino default hardware (10 bits at 1024). So, for a 12-bit sample that is increased from 10 bits to 12 bits, a 2-bit increase is required.

Then the sample is "decimated," dividing by N=2 or shifting right (2) bits. Decimated Data = summed data (shifted right two spaces). The n2 is the two extra bits. Binary 10000 shifted two places to the right becomes Binary 100. That means the two least significant bits, bit 20 and 21, are shifted out making the sample 12 bits.

Increasing the Resolution of the Data

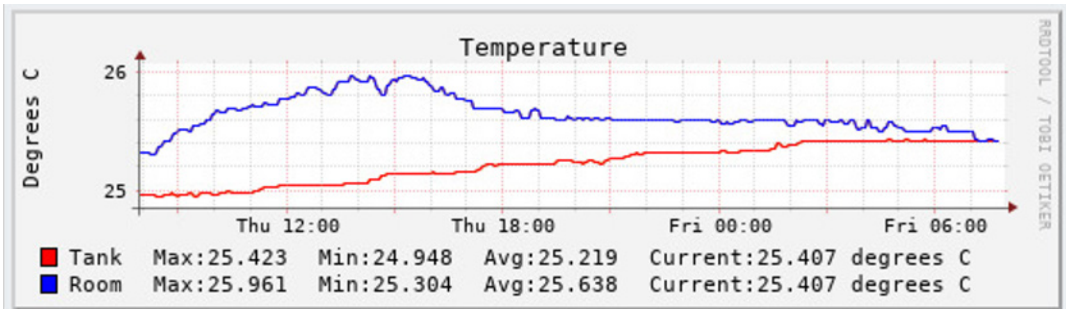
If we were to stop here, we would have a 12-bit sample that would still only have the resolution of the original 10-bit sample, and stair-stepping would still be seen in the graphical representation, as above. Oversampling requires the values to have the least significant bit (LSB), 0 or 1, "dither" or change randomly. That means the LSB (bits 20 and 21) of the data must vary.

Data variations can be ensured by introducing an electrical noise function that will result in the necessary variation of LSB. The electrical noise is known as "white noise," comprised of random signals that cover a broad spectrum. Typically, white noise is avoided in engineering. However, white noise can be used in a non-traditional way. White noise is picked up intentionally in the interface connection between the analog sensor (water quality sensor) and the Arduino Mega 2560. This approach picks up the random electrical noise through the yellow wires connecting the protoboard to the Arduino. In this case, we use this random white noise to our advantage.



The yellow wiring above is used to pick up the white noise from the water quality sensor to the protoboard and Arduino.

By using slightly longer yellow wires to connect the analog water quality sensor to the Arduino, we allow the wires to act like antennae and pick up the random white noise generated by the components of the PolyPonics System and other electrical sources. The magic of 10-bits to 12-bits sampling happens because this random white noise causes the LSB to vary. The graph plotting the data shows that oversampling really works.



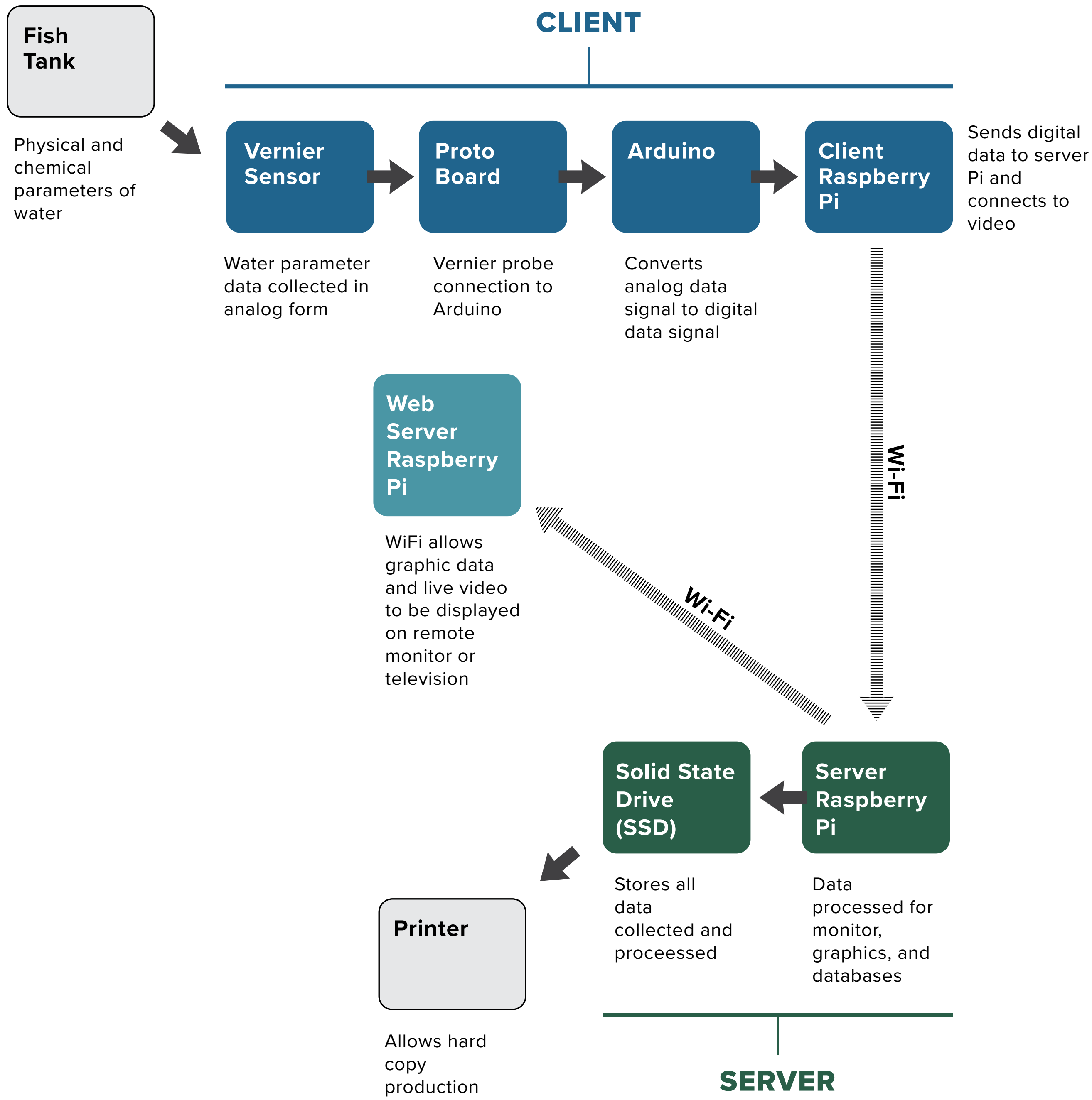
Plotting the 12-bit data is shown above. Notice the reduced stair-stepping in the graph with 12-bit data when compared to the graph of 10-bit data.

Question: *Can the resolution be increased even further, from 10 to 16 bits?*

Answer: *The answer is yes. But, remember that each additional bit requires four additional samples. So, processing time is the problem, and a high-speed processor would be required. The Arduino Mega 2560 has only a 16 MHz clock which is not fast enough to perform that increased level of oversampling. So, this application would not be practical.*

APPENDIX 6

Diagram of Data Path



APPENDIX 7

Glossary

Acronym	Terminology
ADC	Analog to Digital Converter A system that converts a continuous analog signal into a discrete, quantifiable binary signal. Performed by the Arduino.
ARM	An ARM processor is one of a family of CPUs based on the RISC (Reduced Instruction Set Computer) architecture developed by Advanced RISC Machines (ARM). 64-bit RISC multicore processors.
BAS	Bourne Again Shell (BASh) A command processor that runs in a text (terminal) window where the user types commands that cause actions to occur.
bus	Bus is a communication system that transfers data between components inside a computer or between computers. It is a row of pins or sockets where data can be transferred between computer components or within an individual computer.
Client-server	The client-server relationship describes the relation between the client and how it makes a service request to the server and how the server can accept these requests, process them, and return the requested information to the client.
GIMP	A free, open-source raster graphics editor used for image retouching and editing, free-form drawing, and converting between different image formats.
GTK (Formerly GIMP)	A cross-platform widget toolkit for creating a GUI.
GUI	Graphical User Interface Allows the user to interact with electronic devices via visual representations of the data—graphs, meters, icons, etc.—instead of with text.
GUNPLOT	A portable command-line driven graphing utility for Linux.
GNU	Recursive acronym, standing for GNUs Not UNIX. GNU Project was a free, open-sourced software developed at MIT for use by all computer users.
GPIO	General Purpose Input/Output The double rows of GPIO pins connect the client RPi to the outside world. These pins will receive data from the Vernier sensors through the Arduino and transfer them to the Client RPi.
HH	Henderson-Hasselbalch This equation is used to calculate the pH of a solution. In chemistry, it describes the derivation of pH as a measure of acidity (using pKa, the negative log of the acid dissociation).
HTML	Hypertext Mark Up Language The language for defining web-based content and display.
Oversampling	Enhancing ADC resolution by introducing additional signals to increase the sampling rate and improve accuracy.