

### **Step 1: Making parts**

3D printed parts: I recommend PETG for all parts since it doesn't react with saltwater. Print the reaction chamber at 100% infill and up your extrusion multiplier a little from your standard – this will help make sure this part comes out watertight. The rest of the parts are pretty forgiving and can be printed at any settings that give you good results. You may need to drill out some of the holes a bit. If you choose to wall-mount your Aquabot, the wall mount brackets will need to be printed with supports.

Laser cut parts: The reservoir and case panels are designed to be cut from 4mm thick acrylic and bonded together using a solvent-weld adhesive. You might be able to CNC cut the casing panels since it doesn't matter if they line up 100% perfectly, but you'll almost certainly need the perfectly straight cut lines of a laser cutter to achieve watertightness with the reservoir parts.

### **Step 2: Calibrating your pH sensor**

Your sensor needs to be calibrated before it can be used. Although Atlas Scientific provides an Arduino sketch that's supposed to use SoftwareSerial to communicate with the EZO board, I could never get it to receive commands properly. For simplicity's sake I suggest directly interfacing with the EZO board, using an Arduino as a serial bridge. There are lots of guides on the internet on how to do this, but basically it involves connecting TX of the EZO to TX of the Arduino, RX to RX, VCC to 5V, GND to GND, and then either shorting the Arduino's RESET pin to GND or (if you have a board where this is possible like the Arduino Uno) removing the processor from the Arduino. Then you can plug the Arduino into your computer and use Arduino IDE's serial monitor to communicate directly with the EZO. After that, follow the calibration steps outlined in Atlas' pH EZO datasheet.

### **Step 3: Modifying the nano CNC shield and EZO isolator**

Fixing bad traces on clone boards: If you have a genuine Keyestudio board, you don't need to worry about this. However, many of the cloned boards have bad traces which connect a few pins to ground that should be connected to Vcc. To fix this, follow [this guide on Instructables](#).

12V output pins: The CNC shield has two hole contacts marked "Mot\_VCC" on the right side (as oriented in the project case). We're going to need the full voltage output in a couple of places, so solder some header pins on these outputs so we can use jumper wires to connect to them.

Getting a usable 5V rail: The one downside of the Arduino Nano 33 IoT is its lack of a functional 5-volt output. It has a "5V" pin, but this is only active if you connect two solder pads and power the board over USB. In order to get the 5V power we need for our two breakout boards, we're going to give the CNC shield a 5V supply separate from the Arduino. First, cut the 5V pin off the Arduino (or, if you're soldering your own headers, don't add one on that contact to begin with). Now get your 5V linear regulator, attach it to a piece of perfboard, and solder up some male-female jumper wires to its three contacts (if you don't have any perfboard lying around, you can solder the wires directly to the regulator, but I find the perfboard keeps things neater).

Swapping out pins on the EZO isolator: Your electrical isolator board for the pH circuit probably came with 90 degree jumper pins attached. You'll need to desolder and remove these and replace them with straight pins. (When I purchased my isolator board, it came with extra straight pins in the box.)

#### **Step 4: Assembling the case and reservoir**

Use a solvent-weld adhesive – I recommend Tensol 12 here in the UK, but available brands may differ elsewhere in the world – to join all five reservoir panels and all panels for the main casing except the front. Note that the reservoir must be completely watertight – ideally you should leak test it for 24 hours before doing anything else.

Use a soldering iron set to around 250 degrees C to push the threaded inserts into the back and bottom panels of the casing. The larger ones of these are M4 x 6 (outer diameter) x 4 (length) inserts, and the smaller ones are M3 x 5 x 4.

Continue by pushing threaded inserts into the six front panel mounts. The type 1 mounts get M4 x 6 x 4 inserts in both holes, while the type 2 mounts only get one insert each, in the larger hole.

Attach the front panel mounts to the case. The type 1 mounts go on the sides and top, screwed in from the outside. The type 2 mounts go at the bottom, and are screwed into the threaded inserts in the bottom panel from the inside. (This keeps the bottom of the Alkabot flat so it can be placed on a shelf.)

#### **Step 5: Preparing the motors**

Solder the male ends of two short male-female jumper wires to the contacts on the bottom of the 24mm DC motor. Position them so they come out parallel to the bottom of the motor, pointing in the same direction.

Superglue the magnets into the magnet plate and fit the plate to the DC motor spindle. Note that you may need to drill out the hole for the spindle a bit.

Screw the motor mount plate into the threaded inserts at the bottom right of the case, and clip the motor into it.

If your stepper motors didn't come with their own wires, you'll need to check the coils to figure out how to wire them. [This Hackaday guide](#) is a good place to start. If they did come with wiring but they're terminated in JST connectors instead of Dupont, you'll need to clip off the existing connector and attach a new one. The easiest way to do this if you don't have a crimping tool is to cut some female jumper wires and solder the ends to the motor wires.

#### **Step 6: Power connector**

Powering the CNC shield: You have two options for running power from the DC jack on the side of the casing to the CNC shield. You can either solder some wire directly to the connections on the back of the shield where the barrel jack connects to the PCB traces, or use a spare barrel jack connector. Either way,

the other end of these wires will need to be soldered to the DC jack to be fixed to the left panel of the case. Make sure you're not reversing the connections when you do this! You can check by using a multimeter to check continuity between the pin on the board's DC jack and the pin on the one you've just attached. Once the wiring is complete, attach the connector to the case.

### **Step 7: Attaching the electronics**

Screw M3 x 6mm standoffs into the inserts in the back panel. Make sure your standoffs have 4mm stems or use nylon standoffs so you can trim off the excess length.

Seat the Arduino Nano 33 IoT and TMC2208 stepper drivers in the CNC shield. Note that the Arduino's micro-USB port should face the edge of the board, and the "DIR" pins on the TMC2208s should be on the side closest to the Arduino.

Seat the pH EZO board in the electrically isolated carrier, lining up the GND, TX, and RX pins on the small pH board with the identically labelled pins on the carrier.

Place the CNC shield, 5V regulator, EZO board assembly, and DC motor driver on their respective standoffs and screw them in with M3 x 6 screws.

### **Step 8: Wiring**

From the 5V regulator, the input voltage wire goes to one of the Mot\_VCC pins you soldered onto the shield, the ground goes to any free GND pin on the shield, and the output wire goes to any 5V pin. This gives the shield a functional 5V rail for powering other components.

Use five long female-female jumper wires for the pH board. Connect them to the CNC shield as follows: RX to Z+, TX to D12, OFF to Y+, GND to GND, VCC to 5V. Note: with Z+, Y+, and D12, you'll need to use the INNER pins as the outer ones are all grounds.

Use seven female-female jumper wires to connect the DC motor driver to the CNC shield. Connect them as follows: VM+ to your second Mot\_VCC pin, Vcc to 5V, GND to GND, PWMB to CoolEN, BIN1 to Resume, PWMA to X+, and AIN1 to D13. Note: for CoolEN and Resume, use the INNER pins.

Connect the two wires you soldered to the 24mm motor to the two MOTORB pins on the DC driver board; it doesn't matter which way around they go.

Do not connect the stepper motors yet.

### **Step 9: Reaction chamber and pH probe**

Use a soldering iron to push the threaded inserts into the reaction chamber: four M4 x 6 x 6 inserts in the side and one M3 x 5 x 6 in the bottom. Note that the bottom hole is undersized to ensure watertightness; go slowly and make sure the insert is pushed straight in, then clean up any extra plastic that has been pushed out inside the chamber with some flush cutters.

Though it's generally not necessary, some superglue around the outside of the threaded insert may help keep things watertight.

Screw the right-angle tubing fitting into the M3 insert. Make sure the fitting is pointing the correct direction to face the front of the casing once the chamber is in place.

Insert the ends of the 0.8mm ID silicone tubing and one of your two pieces of 1.6mm ID tubing into the holes on the top of the reaction chamber. The holes are different sizes, make sure you're putting the tubes in the right ones. These should be tight enough that you'll need to twist the tubing downwards a bit to get it to go in, but not so tight they pinch off the flow of water. Push the last piece of 1.6mm tubing onto the right-angle fitting at the bottom. You may optionally wish to secure this with a ziptie.

Attach the reaction chamber to the case using four M4 x 10 screws.

Connect the pH probe to the pH sensor board.

Use a soldering iron to push two M4 x 6 x 6 threaded inserts into the pH probe holder. Clip the pH probe into the holder, then position the probe so that the tip is inside the reaction chamber and attach the probe holder to the case using two M4 x 10 screws.

Slide the pH probe up or down to position the tip around 8-10mm above the bottom of the reaction chamber. Drop the magnetic stirrer bar into the chamber.

Gather up the pH probe wire and ziptie it in a bundle to keep it out of the way.

### **Step 10: Assembling the motors**

Assemble the bearings first. Use a soldering iron to push threaded inserts into the bottom bearing plates: three M3 x 5 x 4 inserts into the holes facing the flat side, and one M3 x 5 x 6 into the long hole that starts on the outer edge. Place three steel dowels in the holes on this plate and slide three roller bearings onto them. Clip the top bearing plate onto these and three M3 x 16 screws to fix these to the threaded inserts on the bottom plate. Screw a grub screw into the side of the bearing, just until it is flush with the outer perimeter. Repeat this process for each of the three bearing assemblies.

Slide the pump top onto the pump bottom plate, but only by a few millimetres, leaving the screw holes clear. Use M3 x 12 screws to attach one motor plate and stepper motor to opposite sides of the front panel in the bottom position. Position the motor plate so that the small hole in the middle of one side is on the left. Slide one of the bearing assemblies onto the motor shaft. Line the grub screw up with the hole in the side of the motor plate, and use a hex key inserted through the hole to tighten the grub screw onto the motor shaft. Make sure the bearing assembly does not come off the motor shaft if pulled with moderate force.

Do the middle motor next, followed by the top one.

### **Step 11: Cooling fan**

This is optional, but recommended. Any cheap 12V 40mm fan will do; cut any special connector it might have off the end and solder on some female jumper wire ends. Attach the fan to the inside or the outside of the front plate (outside gives more room for wires inside the case, but some may consider it the less aesthetically valid option...) using M3 screws and nuts. Run the wires through the hole below and to the left of the fan mount point if you're attaching it on the outside. Plug them into the DC driver board's MOTOR A ports, with the positive lead on the RIGHT (towards the corner).

### **Step 12: Fixing the front plate**

Take your three pieces of silicone tubing coming off the reaction chamber and push the far ends through the corresponding holes on the front panel. Pull them through until about 10cm of each tube remains on the inside. Connect the stepper motors to the CNC shield – the top motor to Z, the middle one to Y, and the lower one to X. Now you can place the front panel on the casing and screw in the six screws that attach it to the front panel mounts.

### **Step 13: Tubing**

Now that the front plate is fixed in place, take your thin reagent tubing and lay it across the top pump head, arced over the bearing assembly. Slide the pump top downwards and clamp this in place by threading zipties through the holes down the sides and tightening them as far as you can. YES, ZIPTIES. I have tried using screws here, and the accuracy is better with zipties. You should also place an extra ziptie around the silicone tubing on each side of the pump, as this will prevent it from sliding back and forth through the pump head during operation.

### **Step 14: Installation**

The best place to install your Alkabot is next to your cabinet, slightly above the sump waterline, with the intake line run to the sump. Don't position it too high, or it may be unable to properly draw water in due to a lack of head pressure.

Theoretically, the waste products of the tests performed by the Alkabot are not harmful to pump back into your tank in amounts of a few millilitres a day; however, I would not suggest doing this. I personally run my waste line to a small jar, which I also use every week to empty my skimmer cup.

### **Step 15: Motor calibration**

Assembly is now finished! Insert a micro-USB cable into the Arduino through the hole in the top of the casing, and connect it to your computer. Upload the pump\_calibration.ino code and follow the instructions to determine direction and rotations/millilitre constants for the stepper motors and speed settings for the stirrer motor. During this step, it is best to have your reagent line run to a cup of water that is in the same place as the reagent reservoir will be (ie, sitting on top of the Alkabot).

## Step 16: Programming

Open up the KH\_monitor.ino code. Enter the constants determined in the motor calibration step and your wifi and MQTT details.

You'll need to have a working MQTT server and understand the basics of using MQTT for this bit. There are plenty of tutorials on setting up MQTT on the internet if this is new to you. I use Mosquitto and Node-Red on a Raspberry Pi 4 to run my home automation.

By default, the Alkabot will run a test when it receives a "1" on the topic KHmonitor/start, and will publish the result of the test to KHmonitor/data when it completes the test. You can, of course, change these topics.

## Step 17: Preparing reagent

There is a tradeoff between accuracy, frequency of testing, and duration of "hands-off" use for the Alkabot. The highest accuracy is obtained by using weak acid. Tests using 0.01 M hydrochloric acid, for example, will be accurate to within +/- 0.2 dKH. However, each test may use up to 10ml of acid at this strength, meaning the reservoir will only hold enough reagent for about 20 tests. 0.05M HCl will produce an accuracy of +/- 0.6 dKH, which is sufficient to warn of significant swings in a home aquarium but may not be useful for other purposes. However, at this concentration the reservoir will hold enough reagent for 100 tests. I use 0.02 M acid (obtained by buying 1.0 M acid and diluting it 1 part per 50 in RODI water) and run tests every two days, and find a full reservoir lasts about two months.

Whatever reagent concentration you choose, you should change the REAGENT\_MOLARITY constant in the KH\_monitor.ino code to match.

## Step 18: Testing

You can now upload your code and test the Alkabot using a sample of water from your tank. It is best to have the tubing length and the relative heights of the Alkabot and the water sample as close as possible to what the conditions will be in use. Testing while keeping the device connected to your computer with a serial monitor window open will allow you to keep tabs on the process and identify any problem points.

If your alkalinity result is not what you expect, run the test a few times – spaced at least half an hour apart, to allow the probe to be properly cleaned – and compare the results. If you get wildly varying results, you have a *reliability* problem, which is probably caused by insufficient clamping pressure in the pump heads, movement in the tubing, or some other mechanical problem. If you get results which are consistent but incorrect (eg, your true alkalinity is 9.5 but all your Alkabot results are between 8.3 and 8.6), you have a *validity* problem, which is most likely caused by incorrect pump calibration, by over-priming your reagent pump, or by incorrectly diluted reagent. Run the calibration again, redo the dilution, and if you still have a problem, try adjusting the REAGENT\_PRIME and REAGENT\_PURGE constants. Because the pump exerts less force pushing reagent upward than it does pulling it downward (working against gravity is hard!), PRIME will need to be less than PURGE.

Note that apart from the signal to start a test, there are two debugging signals you can send to the Alkabot over MQTT: sending a "0" to KHmonitor/start (or your custom MQTT topic) will purge the reaction chamber, and sending a "2" will take a pH reading and print it to the serial monitor.