**Step 1: Making parts**

3D printed parts: I recommend using PETG for the reaction chamber since it doesn't react with saltwater. Print the chamber at 100% infill to make sure it's watertight.

The pump heads and bearings should be printed in PLA with at least 70% infill for stiffness and resilience. The "0.8" pump parts should be printed once each, and the "1.6" parts should be printed twice. You'll need three of each of the bearing parts as these are the same for all three pumps.

The rest of the parts are pretty forgiving and can be printed in whatever material you like, at whatever settings you would normally use. 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 case panels are designed to be cut from 4mm thick acrylic and bonded together using a solvent-weld adhesive. You may be able to get away with using superglue, but I don't recommend it given the weight of some of the parts. If you have a large-format 3D printer, you can of course print the panels instead of laser cutting them. You might also be able to CNC them out of a machinable plastic like Delrin.

**Step 2: 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 30ml of acid at this strength. 0.05M HCl will produce an accuracy of +/- 1 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 a given reservoir will hold enough for five times as many 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. A 1 litre reservoir lasts several months.

Whatever reagent concentration you choose, you will need to enter the appropriate normality constant in the Arduino code in a later step.

**Step 3: Modifying the nano CNC shield**

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](https://www.instructables.com/Fix-Cloned-Arduino-NANO-CNC-Shield/).

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).

Bending the 3V pin out of the way: You're going to need access to the 3V pin later, but as it sits at the moment anything plugged into it will block the USB port on the Arduino. You can simply bend the pin 90 degrees to the left (toward the Z+/Z- pins), but a more secure idea would be to desolder the pin and solder on a short jumper wire terminating in a male connector.

**Step 4: Assembling the case**

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 panels for the main casing except the front.

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 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](https://hackaday.com/2019/07/25/deducing-stepper-motor-wiring/) 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.

Attach your stepper motors to the front case panel, using M3 screws in the TOP RIGHT and BOTTOM LEFT holes on each to secure them.

**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. You may need to do this by gluing it to the inside of the acrylic, as most of these jacks are not designed to be attached through such thick walls.

**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.

If you're using an analog pH board (recommended and cheaper!), attach it to the adapter plate using 2mm screws and screw the adapter plate to the standoffs near the centre of the case. This plate is designed for Grove pH sensors and may or may not fit other brands. If it doesn't fit yours, you can make your own adapter plate by drilling some holes in the right spots on any piece of flat plastic.

*If you're using a digital pH board, insert it into the isolated carrier and screw it directly into the standoffs. Note that these sensors are no longer supported in the code base so you will have to make some adjustments. That this may return as an officially supported option in the future, but no promises.*

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

**Step 8: Front panel electronics**

Solder some jumper wires to the button (making sure they terminate in female connectors at the far end). Screw the button to the front panel.

Check your OLED screen according to the notes in the Bill of Materials and determine whether or not you need to add resistors between VCC and SDA/SCL. If you do, you can solder them to the solder points on the opposite side from the pins, on the front of the screen unit. Grab some female to female jumper wires and connect them to the four pins on the screen. Use four M2 screws and nuts to attach the screen to the front panel, making sure you place the two screen standoff brackets between the panel and the screen.

**Step 9: Connecting (most of) the electronics**

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.

Connect the pH board GND to a GND pin on the shield, the VCC pin to any 5V pin, and the SIG to the pin marked HOLD. Note that of the two pins next to the HOLD marking, this is the INNER one, away from the edge of the shield (the outer pins on this row 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 or the front panel electronics yet.

**Step 10: Wiring the liquid level sensor**

This component is optional, but recommended. If you're using one, pass the liquid level sensor wires through the hole in the reaction chamber lid and screw on the plastic nut. You'll then need to cut off the connector the sensor came with and wire some resistors in as in the schematic below:

Diagram

Description automatically generated

There is some flexibility in the resistor values here; I used a 440 Ohm in place of the 390 in the diagram with no ill effects.

Note that the red and yellow wires will be bridged by the larger value resistor, and that you will be left with THREE wires as the blue and white should be connected. Either crimp Dupont connectors onto the three ends or just solder them to some cut jumper wires.

**Step 11: Reaction chamber and pH probe holder**

Use a soldering iron to push the four M4 threaded inserts into the holes on the front of the reaction chamber. Run the

Glue the reaction chamber lid on to the chamber using superglue. This doesn't need to be watertight.

Cut your tubing to the lengths required and insert the ends into the three holes on the top of the reaction chamber. The smaller 0.8mm tubing goes in the middle. 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. It's particularly important to make sure the leftmost piece of tubing goes in as far as possible, as that will help it maintain suction when drawing water out.

Pull the tubing through the three holes in the front panel. Attach the reaction chamber to the front of the case using four M4 x 10 screws.

Use a soldering iron to push two M4 threaded inserts into the pH probe holder and use two M4 x 10 screws to fasten it to the case front panel.

**Step 12: Uploading the sketch and calibrating your pH sensor**

Connect the pH probe to the sensor board, and plug a 12V power source into your Alkabot. Connect your computer via USB and open up the Arduino software. Open up the KH\_monitor.ino code. Take a look through the constants established in the first section of the code. If you're going to be using your Alkabot with MQTT for automation, you'll need to enter your wifi and MQTT server details. If not, delete or comment out the USE\_MQTT line. You may also want to change your reagent normality if you're using a different concentration of HCl, and you should probably check the screen address for your OLED display.

That's all you should worry about for now; we'll come back to the calibration values in a bit.

For pH calibration, while it's technically possible to do this with any known-pH substances, it's best to use proper buffer solutions, preferably at least three. Upload the sketch, open up serial monitor, and enter a "p" in the send box. This will tell the Arduino to go into constant pH measuring mode. You'll need to put the probe in one of your three buffers, wait for the values to stabilise, and then enter the voltage and pH in the calibration.xlsx spreadsheet. Note that you need to enter the ACTUAL KNOWN pH, NOT the pH reported by the Arduino! Repeat this for the other two buffers. The spreadsheet will generate a graph and a trinomial equation for the calibration curve of your pH sensor. Go back to your Arduino sketch and enter the numbers from this equation as PHCURVE\_A, B, and C. For example, if the equation reported by Excel is 10x2 – 60x + 5, you would change PHCURVE\_A to 10, PHCURVE\_B to -60, and PHCURVE\_C to 5.

Re-upload the code with the new values. Double-check your calibration by dipping the probe into the buffers again and making sure the pH reading is now accurate.

**Step 13: Attaching the front panel**

Clip the pH probe into the holder. 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.

Connect the stepper motors to the three motor outputs on the right side of the Arduino shield – top motor to the top output, middle to the middle, etc. Plug in the 5V (blue/white) lead from the level sensor into a 5V pin, the red (GND) lead to a ground pin, and the yellow (data) lead to the pin marked X+. As before make sure you are using the INNER pin of the X+ pair.

Plug one of the wires from the button into a GND pin and the other into the pin marked D13, again using the INNER pin. It doesn't matter which wire goes to D13 and which one goes to GND.

Plug in the four wires from the OLED screen in as follows: VCC to the 3V power pin you modified in step 2, GND to GND, SDA to SDA, and SCL to SCL.

You're now ready to attach the front panel to the case. You might wish to bundle up some of the excess wiring using a ziptie or elastic band first.

**Step 14: Assembling the pump heads**

Assemble the bearings first. Place four steel dowels in the holes on the bottom bearing plate and slide roller bearings onto them. Continue by clipping the top bearing plate onto the dowels and using four M3 x 16 screws to fix these to 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.

If your motors have circular shafts instead of D-shaped ones, you may need to drill out the central holes in the bearings a bit. D-shaft motors are recommended.

Slide one of the bearing assemblies onto each motor shaft. Push it down until it is as close to the acrylic as you can get it without it actually touching. Line the grub screw up with flat bit of the motor shaft (unless using circular shaft motors), and use a hex key 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.

Use a soldering iron to push four M3 threaded inserts into the outer four holes on the pump bottom plate from the bottom side. Attach the pump bottom plates to the Alkabot using the two remaining mounting holes on each motor. The top pump should be the 0.8mm version, which is denoted by a small circular indentation on the side of each of the three pump parts.

For each pump, run the tubing through the channels at the bottom of the pump plate and over the bearing. The leftmost tubing goes to the bottom pump, the middle tubing to the top one, and the rightmost tube to the middle pump. Position the tube support over the tubing and then push the pump top into place. Use four M3 x 25 screws to hold everything together.

**Step 15: 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; 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 16: Motor calibration**

Assembly is now finished! Reconnect your computer and Arduino and enter a "c" in the serial monitor. Follow the instructions to determine direction and rotations/millilitre constants for the stepper motors and speed settings for the stirrer motor. Note that you should be watching the pumps when you enter the "c" command as the first motor will immediately spin briefly so you can determine if it needs to be reversed.

Once the calibration constants are determined, they will automatically be saved in the Arduino's memory. This means they will survive any reboots or power losses. However, they will be wiped if you ever re-upload your code for any reason (eg, if your wifi details change or you re-run the pH calibration). I therefore suggest entering your calibration values as the defaults in the Arduino code.

**Step 17: Testing**

You can now test the Alkabot using a sample of water from your tank. 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 the motors to cool down – 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 mixing a new batch of reagent.

You'll need to have a working MQTT server and understand the basics of using MQTT if you want to automate measurements. 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.