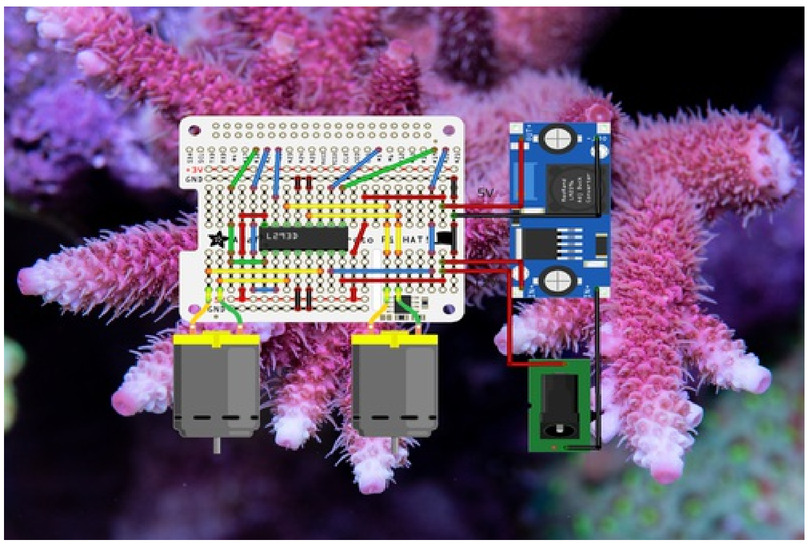
adafruit learning system

reef-pi Guide 7: Dosing Controller

Created by Ranjib Dey



Last updated on 2018-11-19 03:53:16 AM UTC

Guide Contents

Guide Contents	2
Overview	3
Parts	5
Circuit construction	7
Building the Housing	10
Configuration & Testing	14

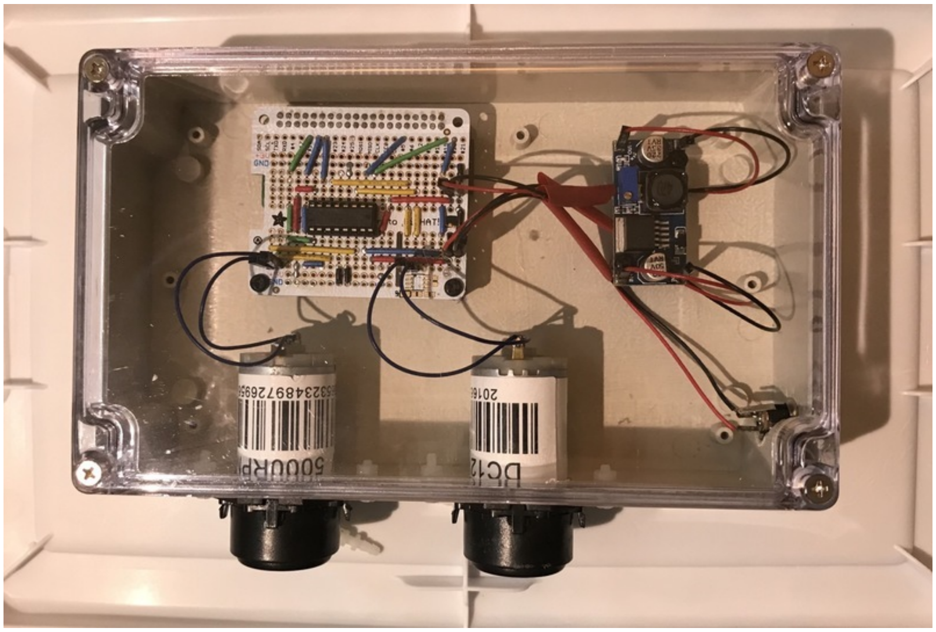
Overview

Welcome to the seventh and final guide in reef-pi series (https://adafru.it/CSr). In this guide, we'll learn how to use reef-pi to build a dosing controller. This guide assumes you have familiarity with reef-pi, if not start by reading the first guide (https://adafru.it/Cx7) in this series.

Dosing controllers or dosers as they are popularly known allows automated addition of liquid in reef tanks. These liquids can be chemical reagents aimed for maintaining a certain type of chemistry (such as optimum calcium or alkalinity levels) or microbial population (such as phytoplanktons or nitrifying bacteria) inside the reef tank. Most intermediate to expert corals require very specific levels of nutrients and water chemistry, so a dosing system is a must. Reef tank inhabitants such as corals, clams and other invertebrates consume certain minerals and release biological waste which changes the water chemistry over time. This rate of changes continues to get faster as the reef tank gets populated with more and more inhabitants.

A dosing system address challenges by systematically replenishing needed elements. Often times, a dosing system is used with sensors (such as pH monitors) or manual water tests to deduce the optimum rate of dosing. A popular dosing setup is to monitor for pH and dose Carbonate (to supplement calcium) and bicarbonate (to supplement alkalinity) to support stony coral growth (image above). This is called as two-part dosing. In this guide, we'll detail how to build such a dosing system, but the technique described here is generic and can be extended for general purpose dosing of most liquids.

We'll be building a standalone unit for this. Since dosing systems are often located close to the aquariums, building them as separate units help us keeping all other controllers (light, power etc) completely isolated from it. This also allows isolation of failure domains, i.e. if the light controller fails it will not impact the dosing system. Following is a photograph of a fully assembled dosing system running on reef-pi.



The heart of the dosing systems are peristaltic pumps, which reef-pi uses to pour liquid at fixed interval in a fixed speed.

Parts

Below is a listing all the parts required to build the dosing controller, including housing and electronics. To calibrate the dosing controller, we'll use some measuring cylinders (or anything that can be used to measure liquid volume at milliliter precision), which is not listed as parts here.

1 x Raspberry Pi Zero
Raspberry Pi zero with headers
ADD TO CART

1 x L293D
Motor driver IC
ADD TO CART

2 x Peristaltic Pump
Dosing pump
ADD TO CART

1 x 12V power adapter
Power adapter
ADD TO CART

1 x Perma Proto HAT
Perma Proto HAT
ADD TO CART

1 x Solid core wire
Solid core wire for jumpers
ADD TO CART

1 x IC Socket
Socket for mounting L293D
ADD TO CART

1 x Male headers
Male headers to connect barrel jack and pumps
ADD TO CART

1 x Panel mount barrel jack
2.1 mm panel mount female barrel jack
ADD TO CART

1 x Male female jumper wire
Male female jumper wire
ADD TO CART

1 x Nylon Standoff
Nylon standoff
ADD TO CART

1 x Project box

Enclosure for project housing

BUY NOW

---

1 x LM2596

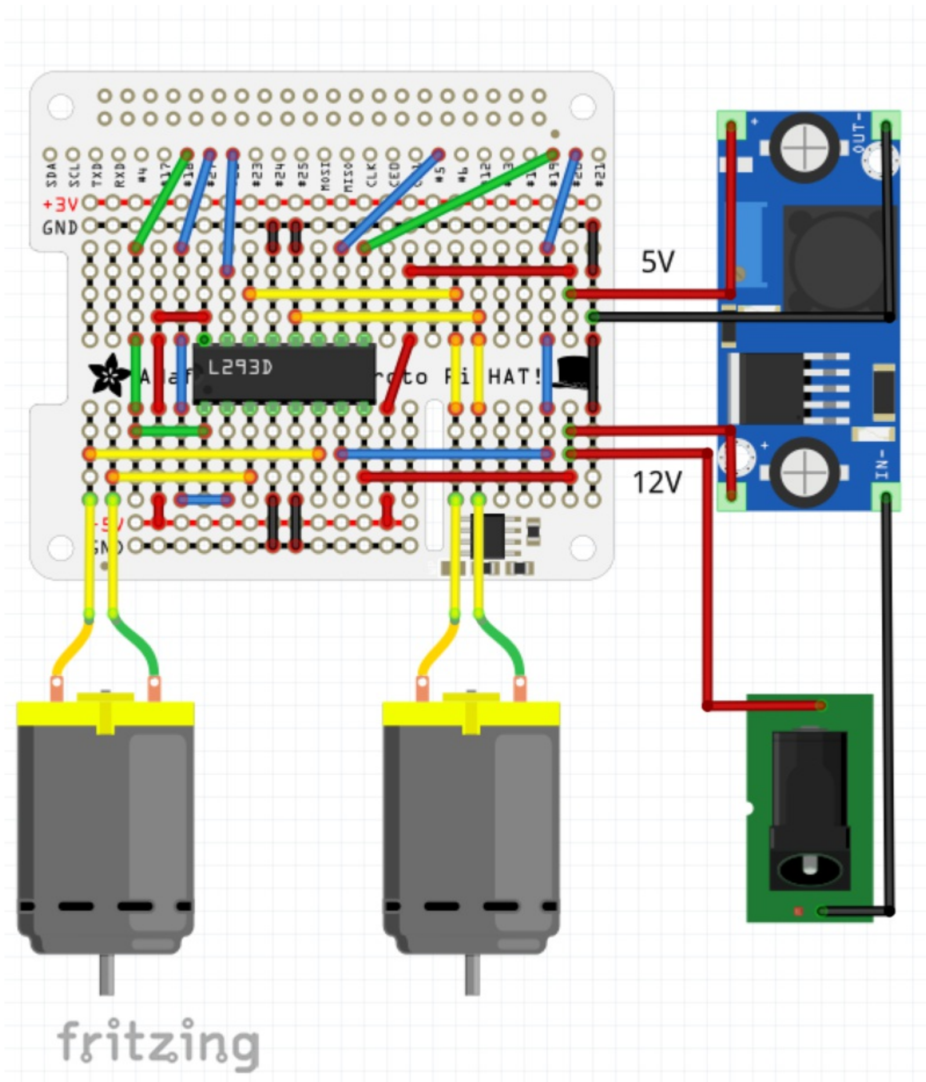
Buck converter

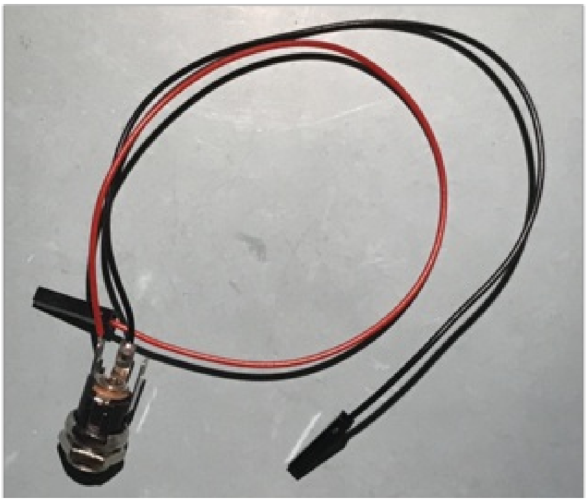
BUY NOW

Circuit construction

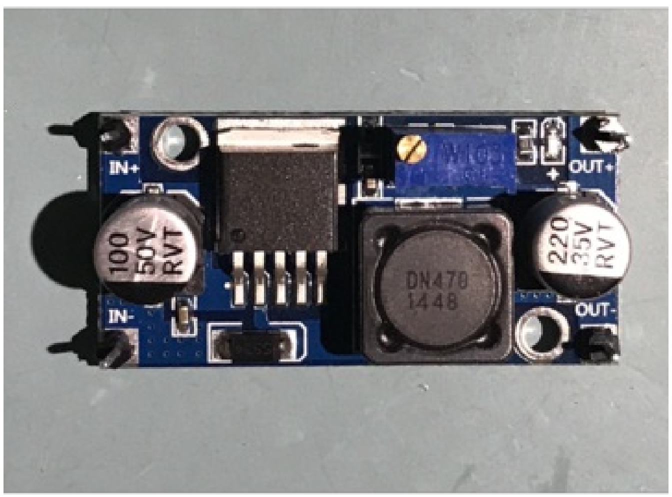
The peristaltic pumps we are using is based on DC motors, which we'll be controlling using L293D ICs. This is a perfect match for us, as this IC support controlling two DC motors, their speed and direction. If you want to know more about these ICs checkout the adafruit guide (https://adafru.it/D3Y) on how to use them with Raspberry Pi. Raspberry Pi comes with two hardware timers (PWMs) that we'll be using as PWM source for L293D IC (each for one pump). These pumps work on 12V, so we'll power the entire build with a 12V 1A adapter and use an LM2596 breakout board to generate 5V power required for Pi. Since this is a standalone build with only two dosing pump control, we can safely run this build with Pi zero, but Pi 3 (or 2) will work just fine.

Here is the fritzing breadboard diagram for the entire circuit. In short, we are using two hardware timers (GPIO 18 & 10) from Raspberry Pi as PWM source and 4 GPIO pins to control the direction of individual pumps.



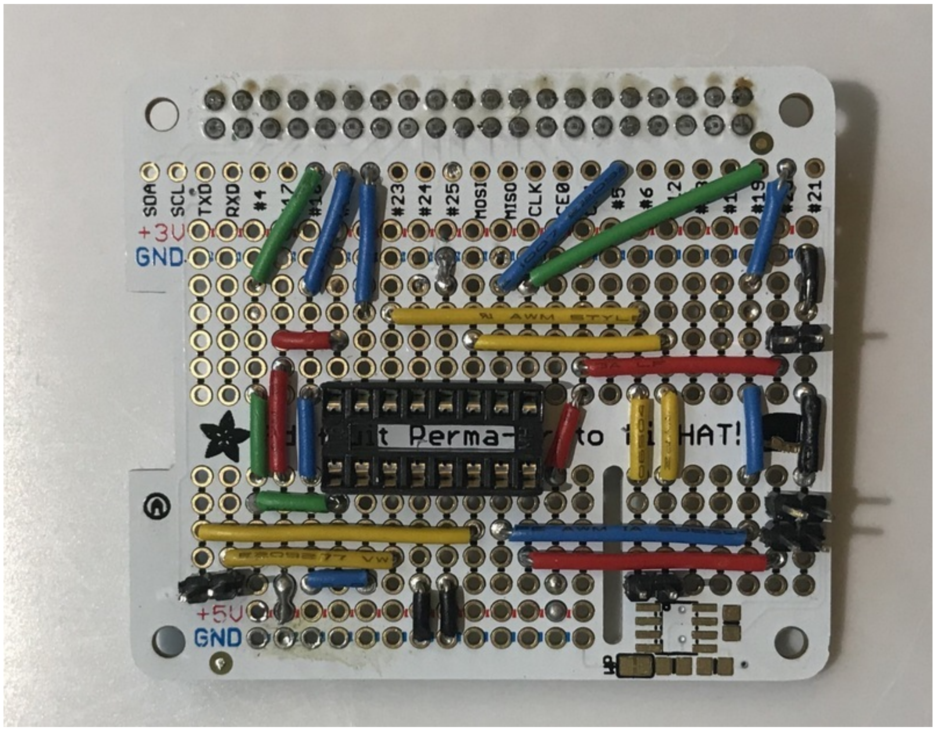


Strip and trim one end of two female jumper wires. Solder them with the barrel jack. We'll use this to connect the power adapter with the rest of the circuit.

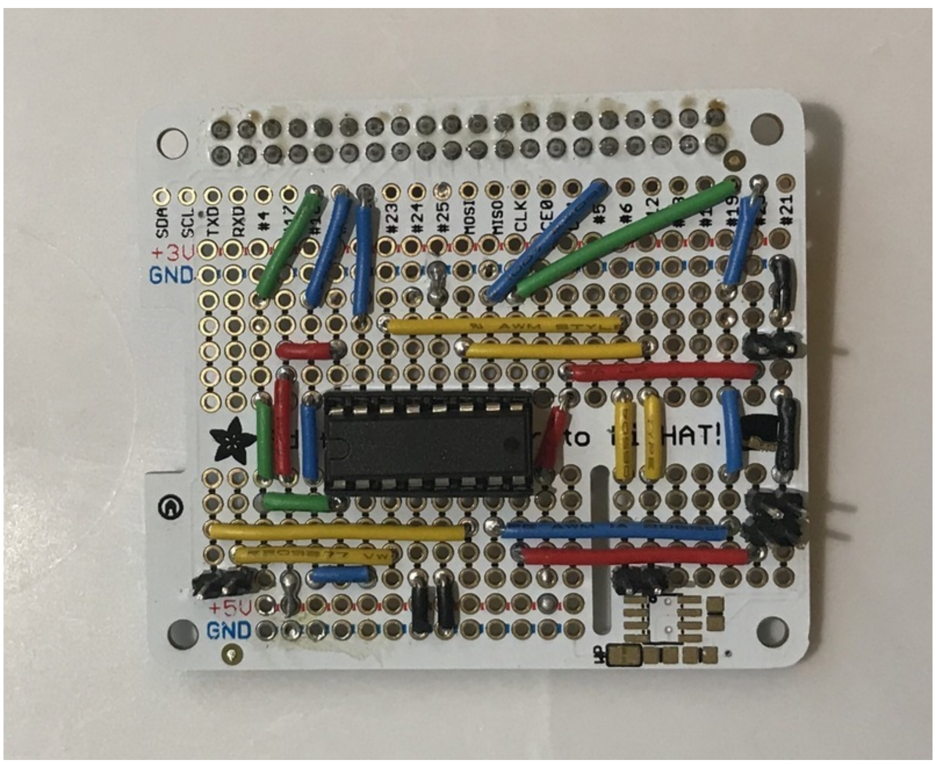


Solder male header pins on LM2596 module. We'll connect this with 12V rails on the perma proto board using female-female jumper pins

I am using a perma proto hat to mount an 18 pin IC socket header, where L293D IC will be inserted. All jumper are made using 22 AWG solid core wires. The blue wires represent GPIO (polarity or direction control) outputs, while green wires represent PWM/hardware timer outputs from Pi and yellow wires represent 12V PWM output from L293D that finally powers the peristaltic pumps.



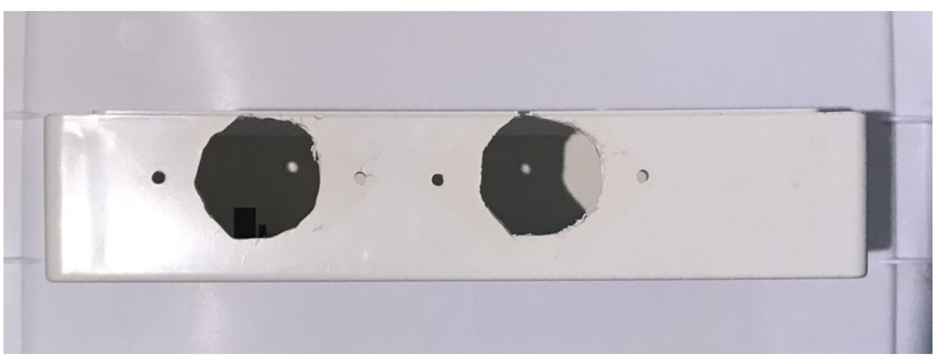
That's all, insert the L293D IC and we are good to go for mounting the circuit inside the enclosure and test it out.



Building the Housing

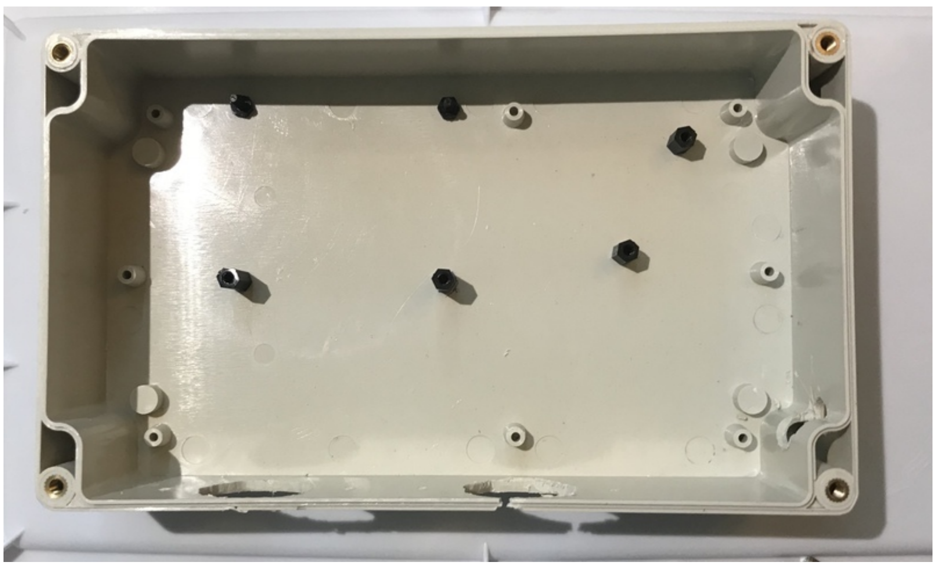
We'll be using an uxcell 7x4x2" plastic enclosure to mount the entire circuit.

Start with cutting two large holes for mounting the peristaltic pumps at the bottom panel. I uses a Dremel tool for this work. You can also use large drill bits and a file to smooth things out.

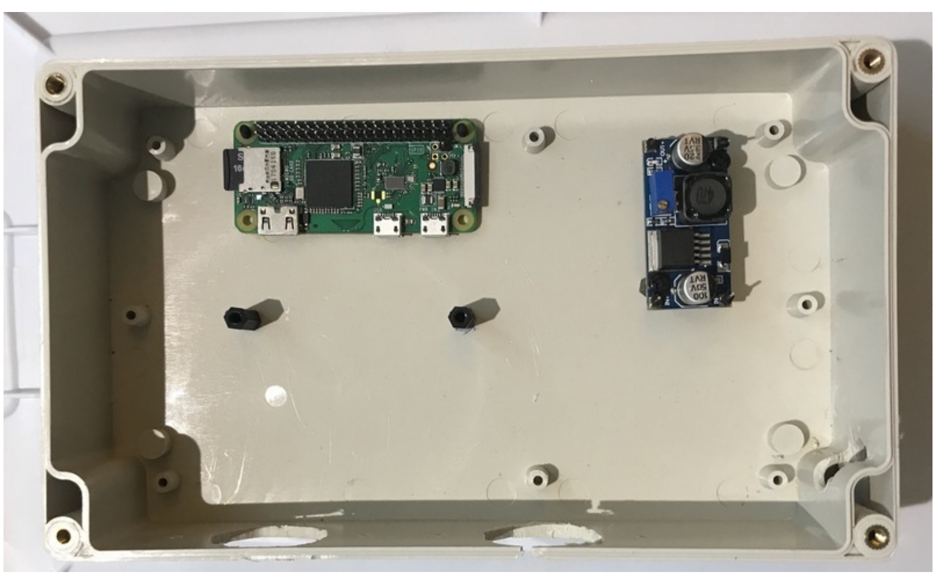


Drill a hole on the side panel for the barrel jack connector that will be used to plug in the power adapter.

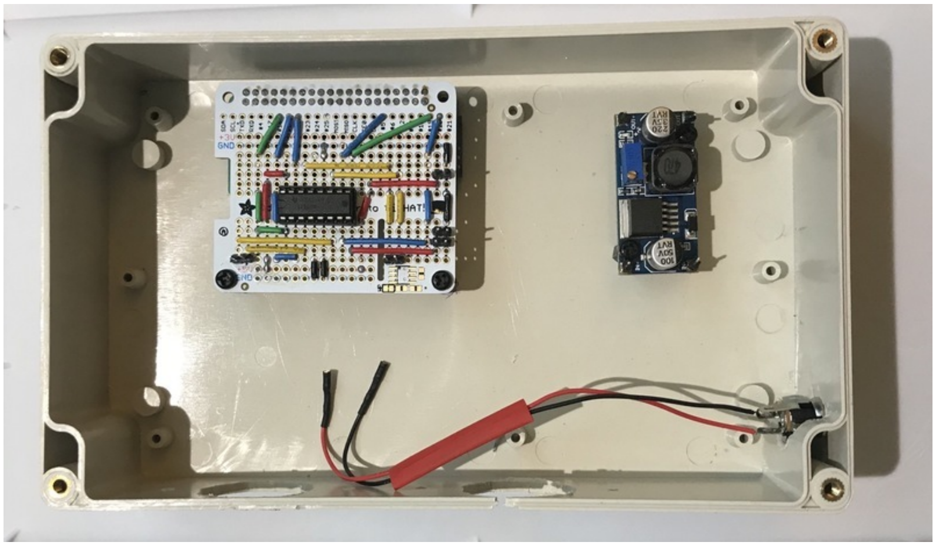
Next, place the circuit components (LM2596 board, perma proto hat etc) and mark the holes required to placing nylon standoffs. Drill the holes and screw in the standoffs.



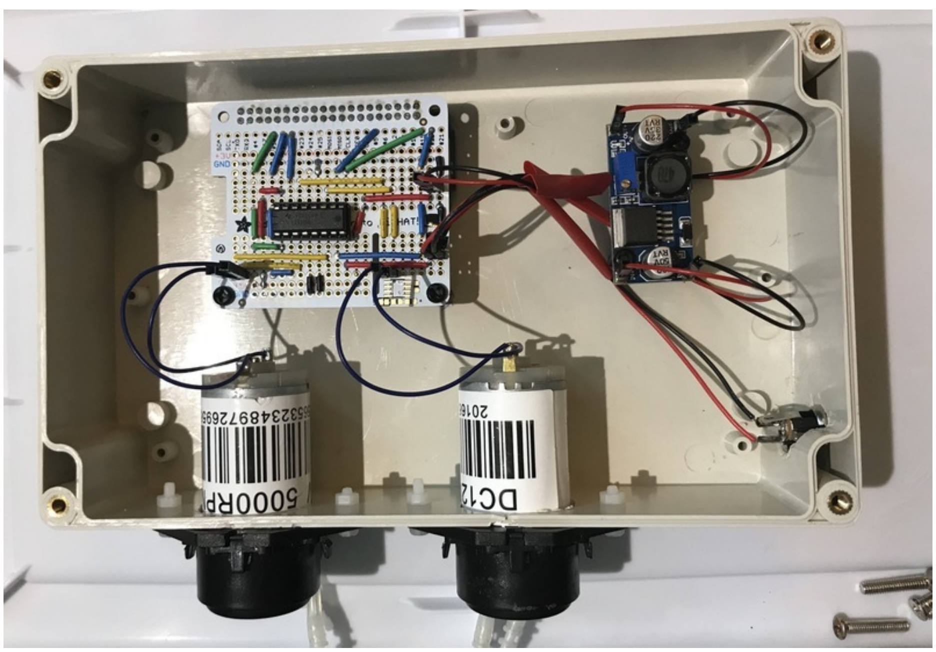
Once standoff are fixed, mount the LM2596 module and the Raspberry Pi Zero



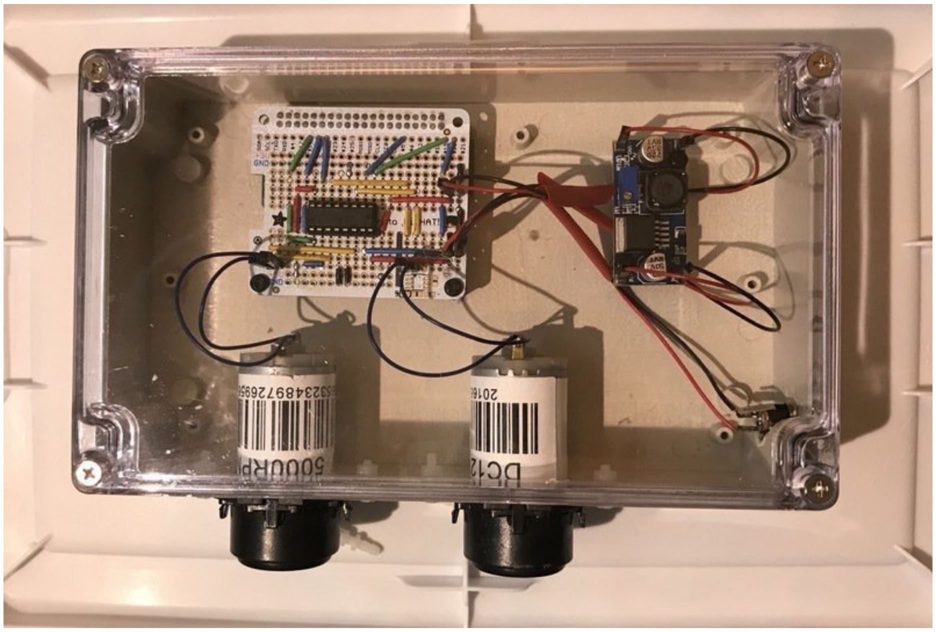
Next, attach the Perma Proto HAT on top of the Pi Zero. Notice I am using a Pi 3 Hat with the Pi Zero, since it gives more space to work with and it works just fine. Affix the barrel jack connector as well.



Once all the electronic components are mounted, attach the peristaltic pumps at last, and connect all the wires. Notice we are using the perma proto hat to create a small 12 V rail (on the right side) from which the LM2596 and the L293D (Vmotor) draw their power.

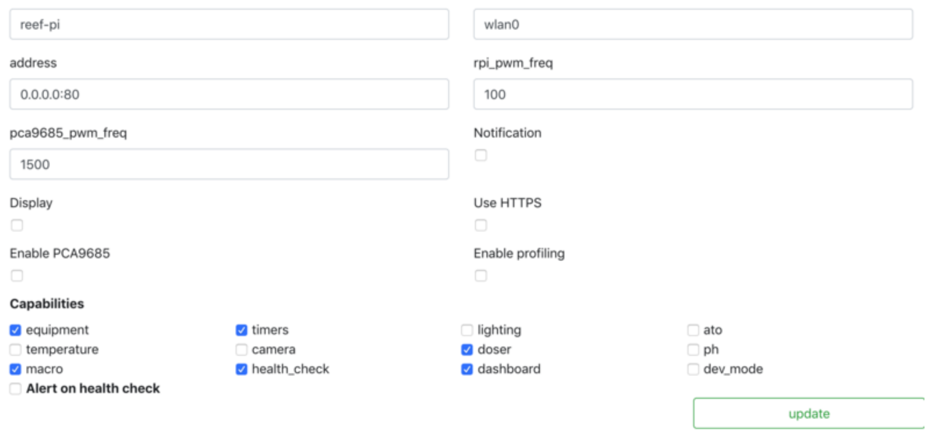


That's it, once all the components are wired, attach the clear case top and our dosing controller is ready for some testing and calibration



Configuration & Testing

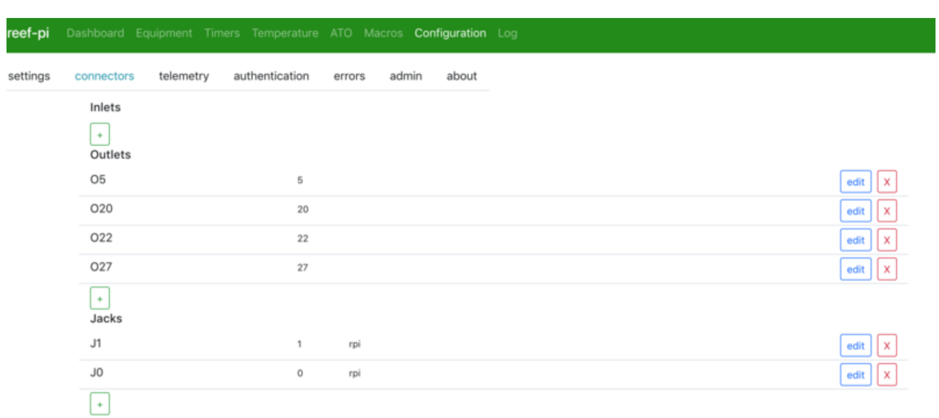
With circuit and housing completed, let's proceed with configuring and calibrating our new dosing system. Power up reef-pi and head to the reef-pi web UI. First, verify you have the doser capability enabled in settings section under configuration tab. Since this is a doser only build, I am disabling the unrelated modules, such as lighting, temperature etc. We'll be using the equipment tab for a trick, I'll discuss that shortly.



Once settings are updated, reload reef-pi from the **admin** section under **configuration**.

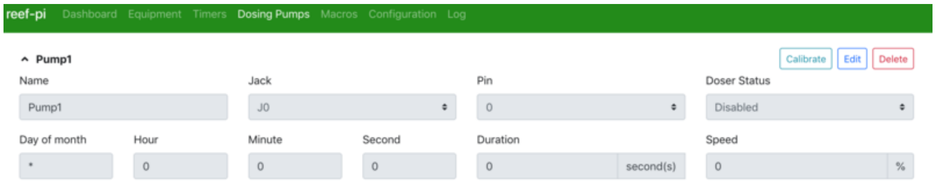
Lights, dosing pump etc are all controlled by PWM pins which are represented as **jacks** in reef-pi. A jack can have multiple pins. Every dosing pump is associated with one of the pins of a **jack**. Since we are using Raspberry Pi's hardware PWM (GPIO . 18 and 19) , we'll declare two jacks, with rpi as the driver (it can be PCA9685 IC as well). We are using L293D IC which allows us to control the direction of peristaltic pump using two input pins for each of the pump. If both these inputs are on or off, the pump is off. If one of them in on and the other is off, then the pump runs. The pumps direction depends upon which of the input pin is on. To simulate this, we'll employ a trick in reef-pi, where we'll create outlets with each of the GPIOs (total 4) controlling the direction of two peristaltic pumps. We'll create virtual equipment with these GPIO and then we can control the on/off of those equipment, which in turn will control the GPIO and there by the direction of the pumps.

Now with this planned out, lets go ahead and create all the necessary connectors , four outlets and two jacks.

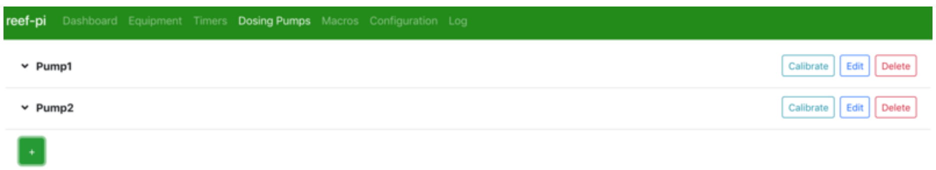


Note, the PWM pins (jacks) are 0 and 1 which represents GPIO 18 and 19. This is because linux kernel maps them as 0 and 1

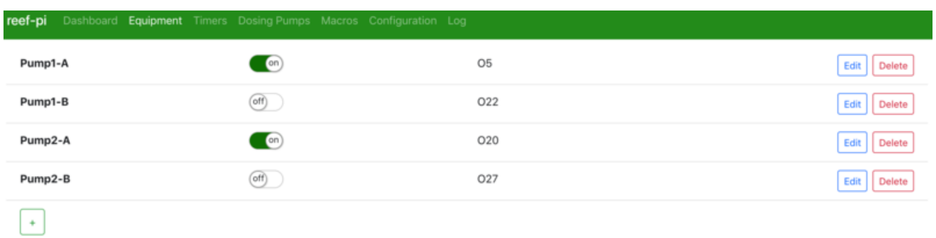
Create two dosing pumps, associate each of them with one of the jack and its pin. Keep the pumps in the disabled state. We'll calibrate them before using them.



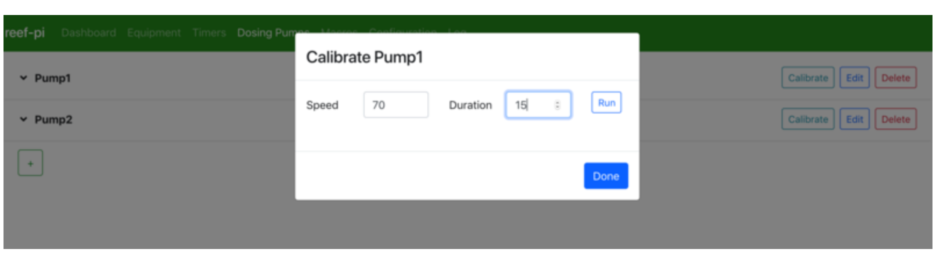
Heres the list of pumps in my **dosing pumps** tab.



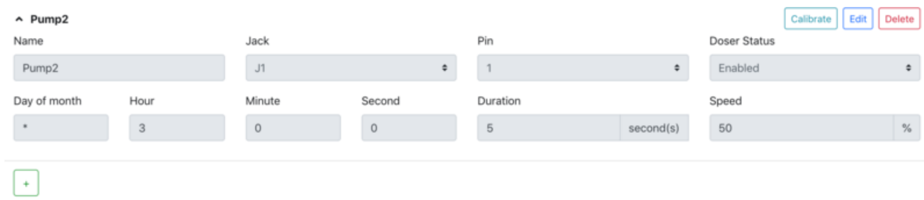
Next, switch to the equipment tab and create four virtual equipment and associate them with the outlets. As per our circuit, GPIO 5 and 22 is controlling the first pump, hence when the equipment associated with them (Pump1-A and Pump1-B) are both on or off, the pump will be off. Same goes for the other pump. In the screenshot below one of the equipment associated each pump is on while the other is off. Which means the pump will run (as long as theres a PWM output). If we make Pump1-A off and Pump1-B on, that will change the direction of PumpA.



With the pump direction set, lets go back to the dosing pumps tab and start calibrating the pump. Different dosing pumps have different capacity (volume of liquid dispensed in unit time) and should be calibrated before they are used. For calibration, individual dosing pumps in reef-pi must be in disabled state. Click on the **calibrate** button. Set a PWM or speed value, which ranges from 1-100 and duration in seconds. Make sure you attach silicon tubing to your dosing pump and use some type of liquid (like water) for testing. In this example I am using speed 70 and duration 15 seconds. This will instruct reef-pi to run Pump1 for 15 seconds at 70% speed.



Use a measuring cylinder (https://adafru.it/D3X) or similar vessel to measure the amount of liquid dispensed by the pump. Note it down. For me it was 5ml. This is the value that you'll use to decide how long you need to run your dosing pump and at what speed, depending upon your dosing regimen. I dose 2ml of two-part solution (Calcium carbonate and Alakalinity) every day. So, I went back and scheduled my dosing pump to dose every night at 3 AM for 5 seconds at speed 50. Click on the edit button, set the schedule an enable the dosing pump. Thats all, your automatic dosing system is ready for operation.



I highly recommend calibrating your dosing pumps every two months and use a dosing container with volume indicators to make it easy to check how much liquid has been dispensed by the pumps over a period of time. There are really expensive and precise dosing pumps in the market that will make this much safer, but I rarely found they are necessary unless you are automating titration (we'll get back to this later). For most reef keeping purpose its the amount of chemical over a time period that matters, and we can always make higher dilution and dispense them more frequently at a lower volume, thus reducing our error margin. In most cases, you can use your pH sensor as a feedback mechanism to check the effect of chemicals on the tank.

Thank you for reading through the 7th guide in reef-pi, I sincerely hope this series will help you automating (or get started) some of the chores involved in reef keeping, a hard but rewarding hobby.