

## **Introduction**

For the MATE ROV competition in 2023 we need to build a device that can sink to the bottom of the pool, then rise to the top of the pool and transmit data repeatedly following these specifications. In order to earn the full points for the challenge we need to use a buoyancy engine. A buoyancy engine is a device that is able to change its density in order to sink and float in water.

We made a float last year using a similar mechanism to what I am making this year, but the enclosure for the mechanism was not well optimized for being small and lightweight. This year I am trying to improve on that.

## **My Approach to the Challenge**

In order to complete the above task, I am using a large animal feeding syringe with a lead screw and rubber stopper attached to a lead screw. I chose this method because it reuses some of the work from last years attempt, and it is as simple as possible.

## **Hardware Progress**

I have figured out the general concept and have modeled an enclosure for the motor. I tested for fit by 3D printing a test enclosure, and it worked. I have to 3D print the enclosure sideways because I want to increase the strength of the plastic in the direction that it is being stressed. I still need to make the top of the syringe that will hold the lead screw and keep the enclosure water-tight, but that will come later in the project.

## **Electronics Progress**

I have tested the particular motor using last years design and have determined that I can power all the electronics using 3 alkaline batteries (we are not allowed to use rechargeable batteries for safety reasons). I am using a small motor driver connected to a Raspberry Pi Pico to control the power coming directly from the batteries and put them into the motor. I am under-volting the Raspberry Pi Pico so I don't need as many batteries.

## **Software Progress**

Controlling the motors is very simple and does not require much processing power. I want to make sure that the code for the float is easily maintainable long past I have graduated, so I am writing it in MicroPython. I chose MicroPython because it is much simpler to teach new developers Python than C, hopefully MicroPython will be supported long into the future, but even if it is not the code should be relatively easily ported to a different implementation of Python on micro controllers.

PDF Version