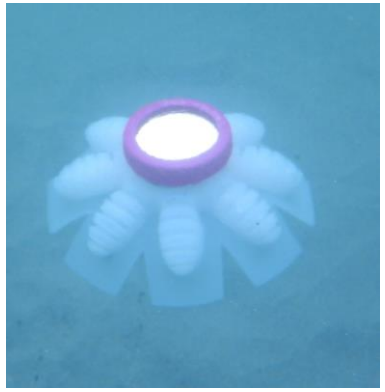
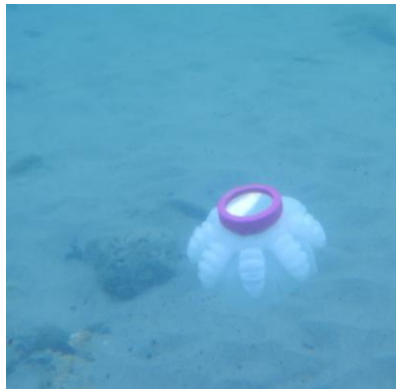
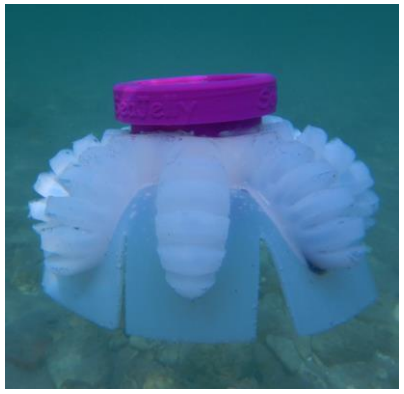






SeaJelly

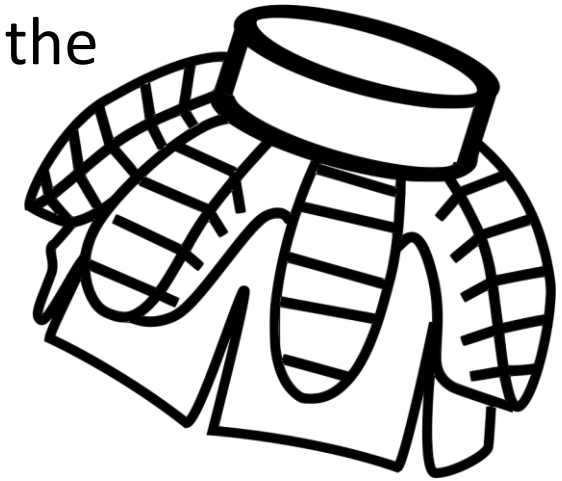
Project Guide



Overview

Build your own free-swimming biomimetic platform with help from the Navy's first soft robotics STEM initiative. The SeaJelly project includes open-source CAD and PCB designs, Arduino code, build instructions, and reference materials. It is a great way to learn new concepts, gain hands-on experience, and encourage creativity. After your first build, try modifying parts of the design or exploring different SeaJelly applications – such as citizen science! JenniFish, a master's thesis research prototype, was used as the foundation for SeaJelly. Have fun naming your own SeaJelly variant! Feel free to send comments, ask questions, and contribute new ideas by email to stem.nswccd@navy.mil.



Build Instructions

Tools, Materials, Fabrication

Tools

Item	Notes
Vacuum pump and degassing chamber	fittings, pump oil
Measuring cup	10-50mL marks, 500mL+ capacity, fit inside degassing chamber
Mixing stick	
Soldering iron	
Multimeter	
Flush cutters	
Wire strippers	for small wire (26-30AWG)
Needle nose pliers	
Safety glasses	
Scissors	
Marker	
Computer	
Micro-B to Type A USB cable	for programming Teensy
3D printer	

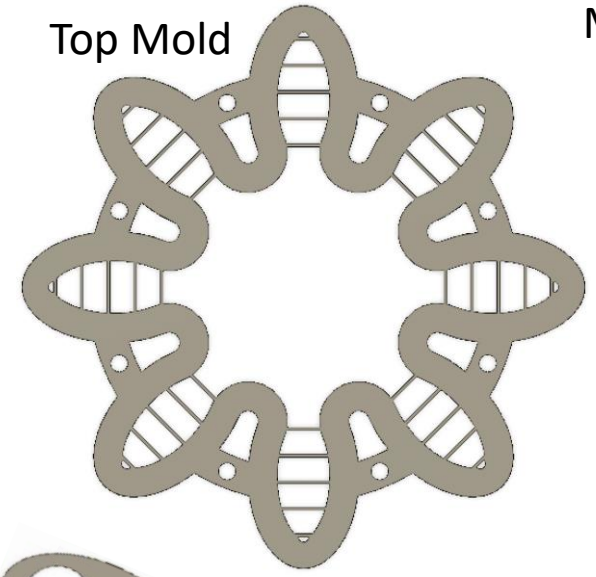
Materials

ITEM	Location URL	QTY
2-3/8" ID, 2-1/2" OD, 1/16" width O-rings	25pk Buna-N O-rings 036 70A	1
1/8" thick plexiglas sheet (not needed if using Pololu laser)	12"x12"x1/8" plexiglas sheet	1
17 x 66.5mm 1/8" plexiglass disks (not needed if DIY cutting)	Laser Quote	1
Ecoflex 00-30 (Trial Unit, net wt 2lbs)	ecoflex-00-30	1
SuperSeal & Ease Release 205	SuperSeal, Ease Release combo	1
PETG printer filament	Transparent White, 1.75mm, 1kg	1
Solder		
Disposable gloves		
Nuts, Bolts, Washers	#8-32, 1inch length	8
Scrap fabric (cotton, polyester)	minimum 3.5” x 6.5”	2
Paper towels		
Marine Epoxy	Loctite 2-hr	
Electrical Tape		

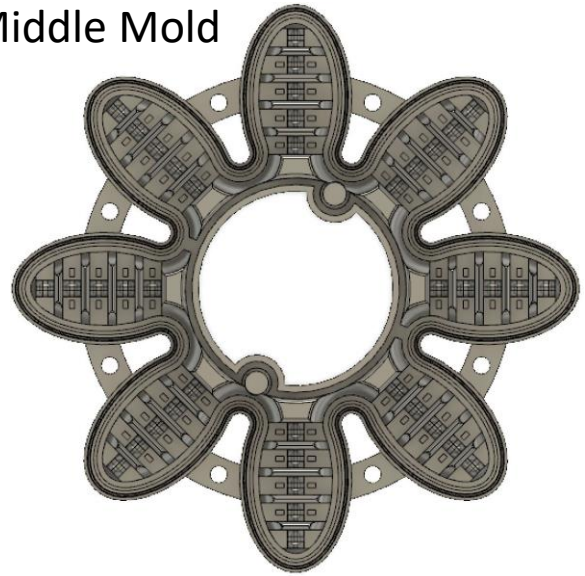
ITEM	Location URL	QTY
Teensy 4.0	teensy 4.0	1
Common Anode RGB LED	product/159	1
IR remote	product/389	1
Pressure sensor breakout board 6pk	product/1212	1
Mini Submersible Water Pump DC 3-6V 120L/H	Vipe-Micro-Submersible-Motor-Water	2
Silicone wire	26AWG	1
9V li-ion recharable batteries and charger	2 EBL batteries and 2 spot charger	1
MicroSD card	SanDisk 16GB class 10 microSD card	1
Pressure-Sensitive Conductive Sheet (Velostat)	adafruit-industries-llc/1361/1528-2211-ND/	1
Copper Foil Tape with Conductive Adhesive	3M 1/8" x 18yd Copper Tape	1
IR receiver	Vishay 20TSOP38438	1
n-channel MOSFETs	TN0702N3-G	2
Temperature sensor	MCP9701A-E/TO-ND	1
Pressure sensor	MPXHZ6250AC6T1CT-ND	1
SeaGlide diode	comchip-technology/1N4001-G/641-1310-1-ND/	2
microSD card breakout board	sparkfun-openlog/1568-1368-ND/5993830	1
470 ohm resistor	CF18JT470RCT-ND	3
10k, 1/8W resistors	CF18JT10K0CT-ND	4
18k, 1/8W resistor	CF18JT18K0CT-ND	2
22k, 1/8W resistors	CF18JT22K0CT-ND	2
5V regulator	Pololu D24V6F5	1
Op amp buffer	296-12671-5-ND	1
9V battery clip	36-235-ND	1
0.1 pitch 32 position male headers	3M9457-32-ND	1
0.1 pitch 32 position female headers	929974E-01-32-ND	1
SeaJelly PCB 3pk	Upload Eagle file	1
0.1 pitch 5 position male headers (optional)	3M9457-05-ND	1
0.1 pitch 5 position female headers (optional)	929974E-01-05-ND	1
3V coin cell battery (optional)	P183-ND	1
3V coin cell battery clip (optional)	BH501-ND	1

Fabrication: 3D Printing

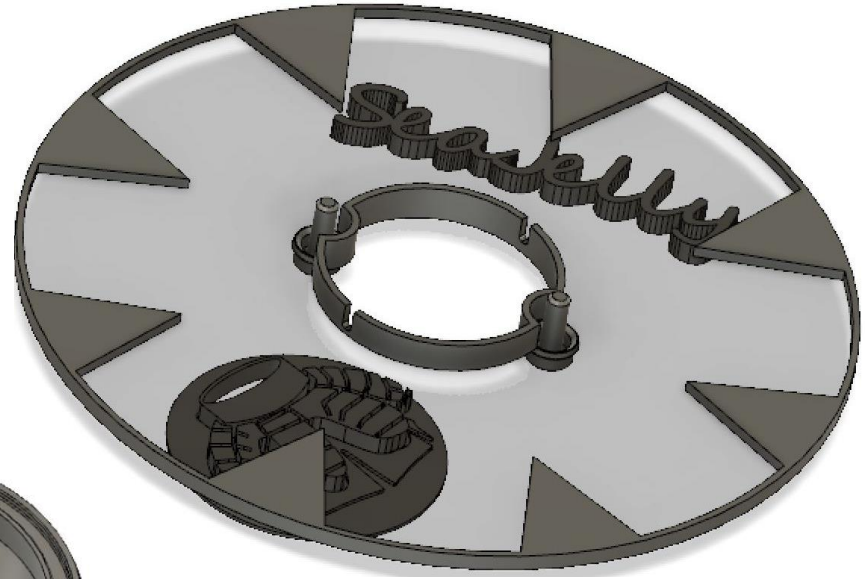
Top Mold



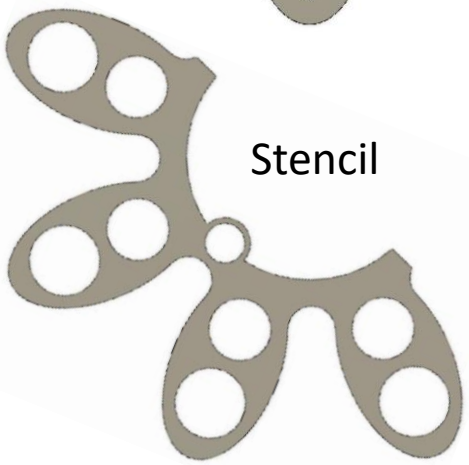
Middle Mold



Bottom, Tube Molds & Supports



Stencil



eLid

eCan

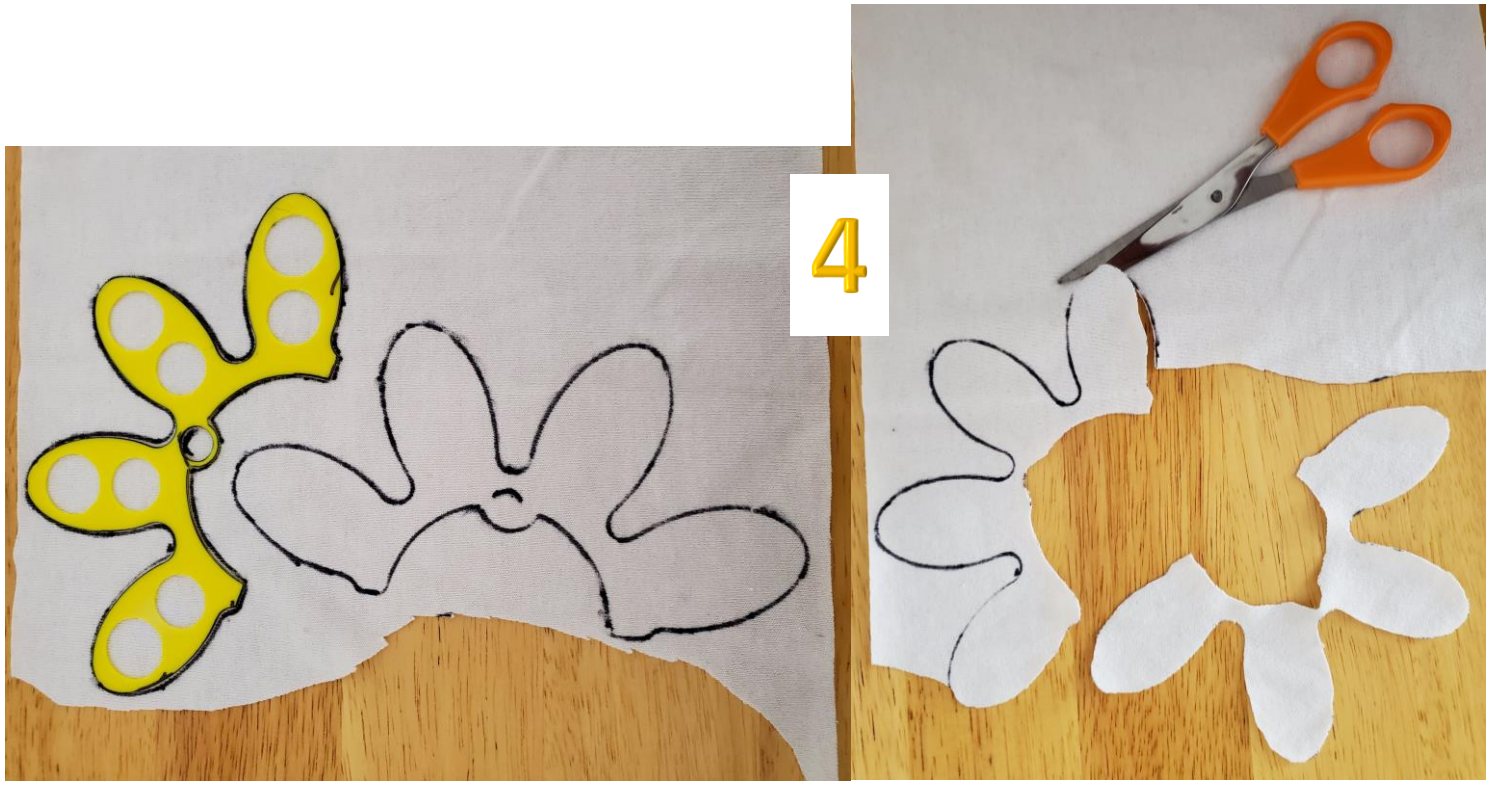
Helpful Hint:
If you are new to 3D printing, start with the small parts to fine tune settings with less material waste.

Fabrication: Mold Preparation

1) Electrical tape tube molds onto bottom mold
2) Apply SuperSeal and then Ease Release to molds per bottle instructions (leave drying time between coats)
3) Fasten together top and middle molds with #8-32 hardware

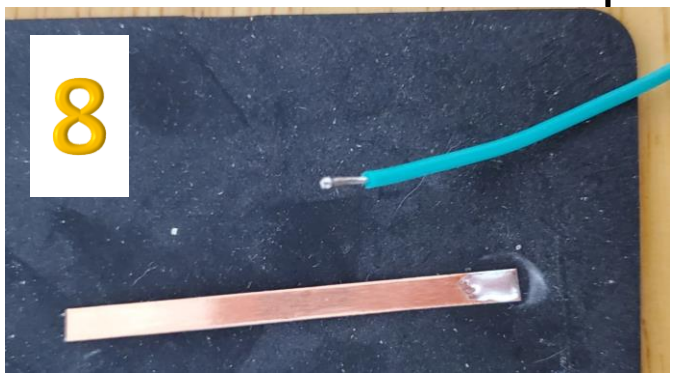
Fabrication: Reinforcement Footprint

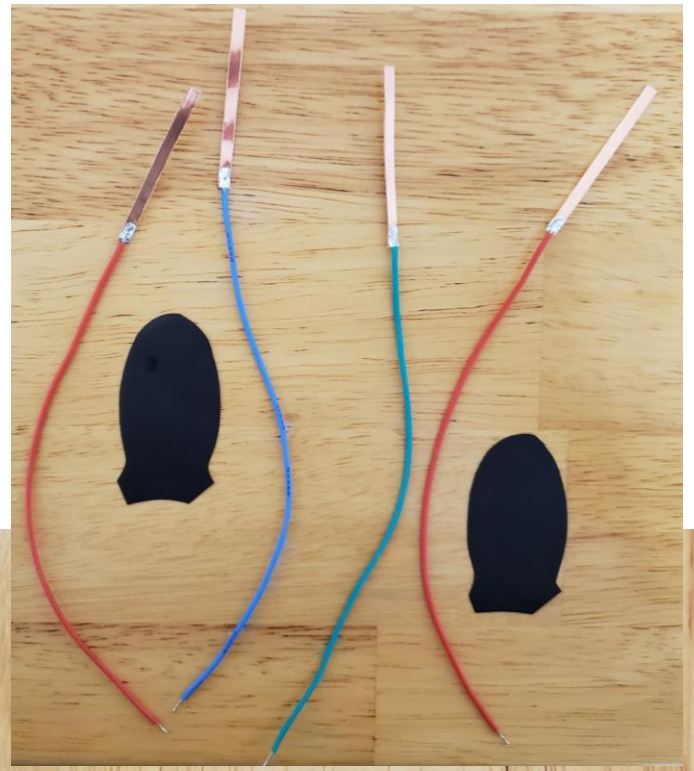
4) Trace stencil on fabric scrap and cut out
5) Trace one actuator twice onto Velostat and cut out

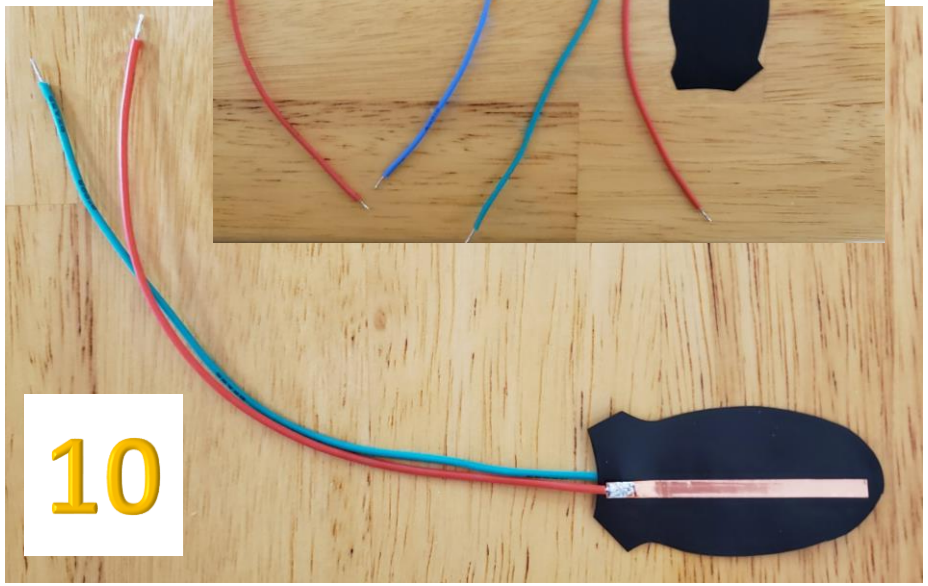


Fabrication: DIY Flex Sensors

6) Cut four 6” pieces of wire
7) Cut four 1.75” strips of copper foil tape
8) Strip wires, tin wire tips and end of copper
9) Solder wire onto copper
10) Remove tape backing and stick to each side of Velostat (try to offset wires slightly, now can cut Velostat footprint smaller if desired)





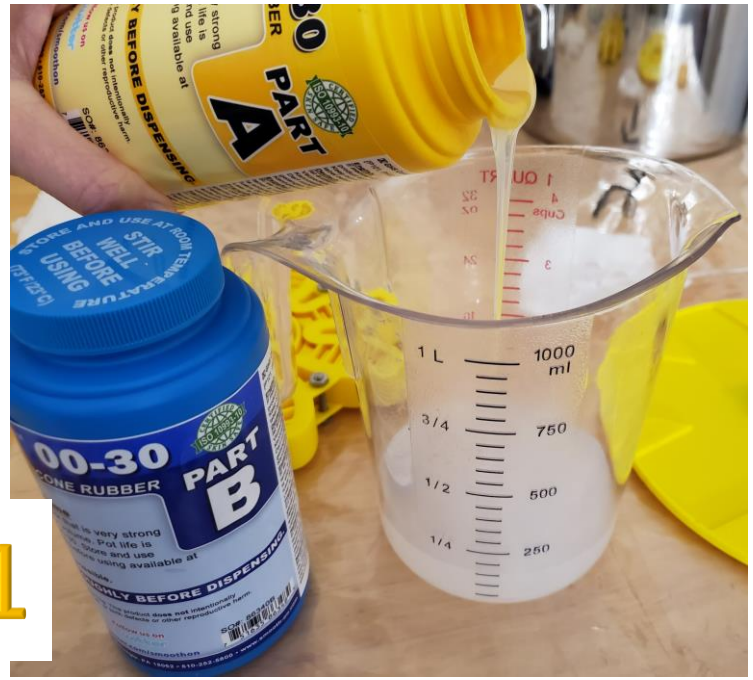


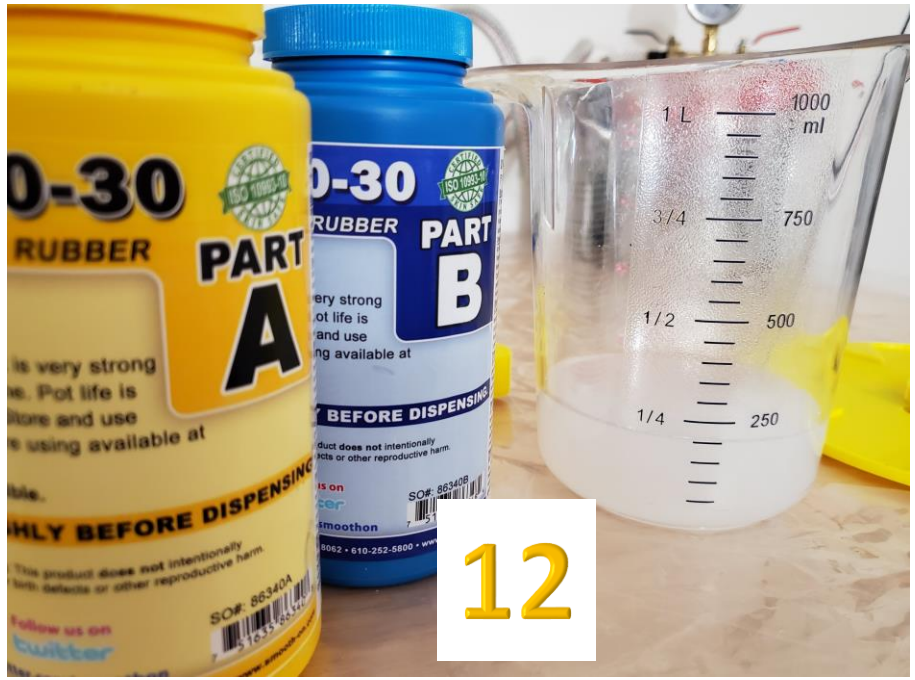
Fabrication: Pour & Mix Silicone

11) Pour 100mL Part B and then 100 mL Part A

12) Mix thoroughly

11





12

Fabrication: De-gas Silicone

13) Remove air bubbles created during mixing with vacuum pump and degassing chamber

13

Fabrication: Skirt

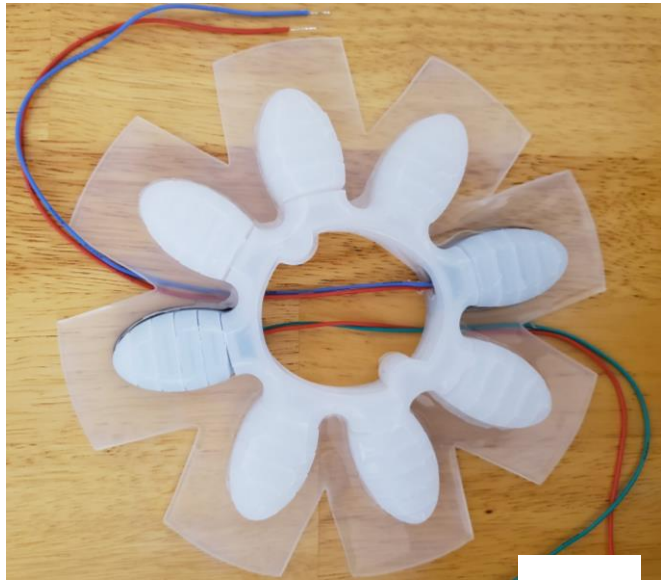
14) Pour thin layer of silicone into bottom mold, fill tube molds
15) Place bottom reinforcement
16) Pour thin layer of silicone on flex sensor reinforcement
17) Gently place flex sensors in mold, routing wires through slots so sensors lay flat
18) Pour a thin layer of silicone over entire footprint

Fabrication: Channels

19) Slowly pour silicone into two-part mold until filled
20) Let cure undisturbed

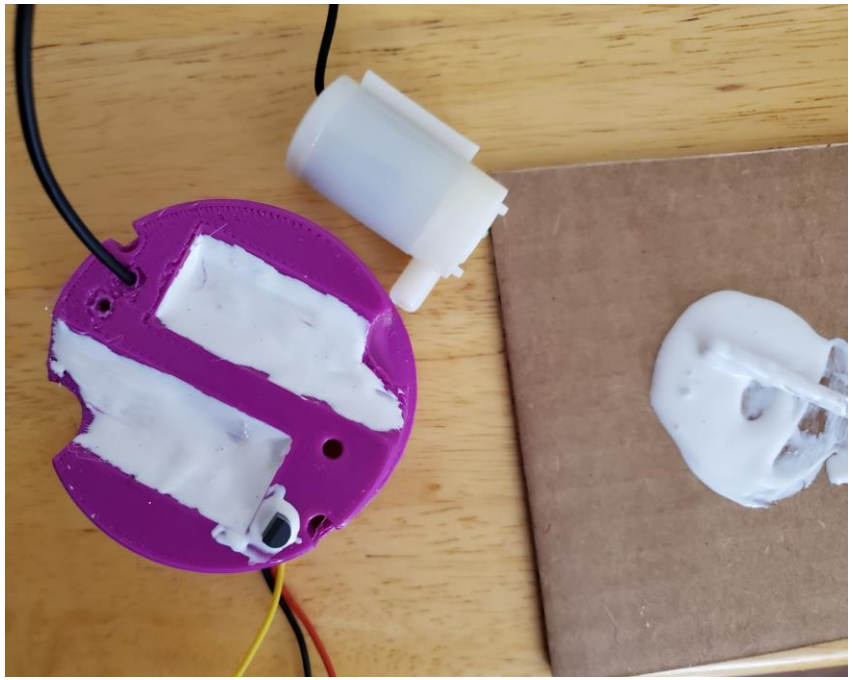
Fabrication: Actuator Assembly

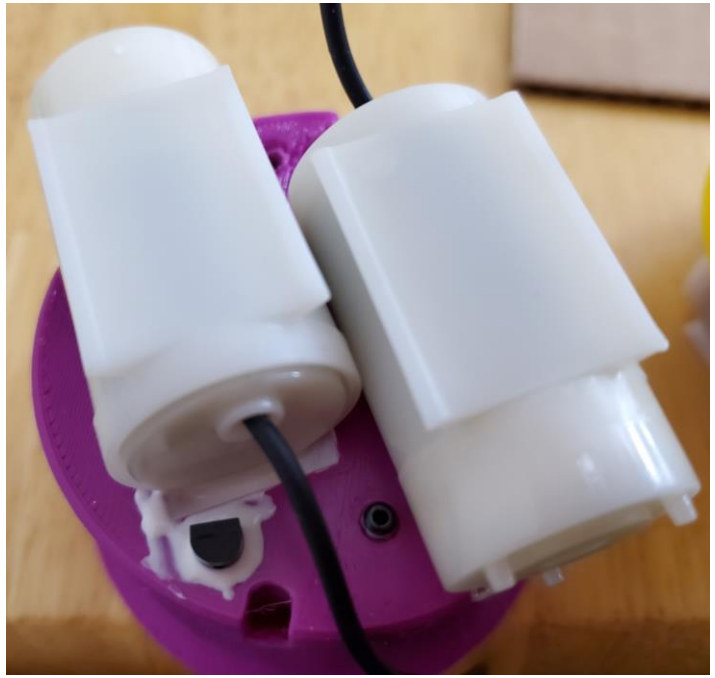
21) Once silicone is cured, remove hardware from two-part mold
22) Slowly pull apart top, middle molds
23) Mix half the silicone amount from step 11 and follow steps 12-13
24) Pour a thin layer of silicone over entire bottom mold cast
25) Place de-molded channels on bottom mold, aligning with footprint
26) Let cure before de-molding actuators from bottom mold



Fabrication: Pump Attachment

27) Solder breakout board and wires to pressure sensor (see SeaGlide instructions)
28) Solder wires to temperature sensor
29) Mix epoxy to secure, waterproof sensors, pumps (continuity check sensors)



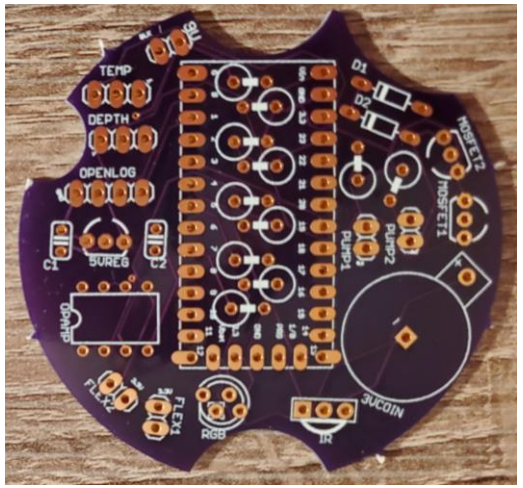


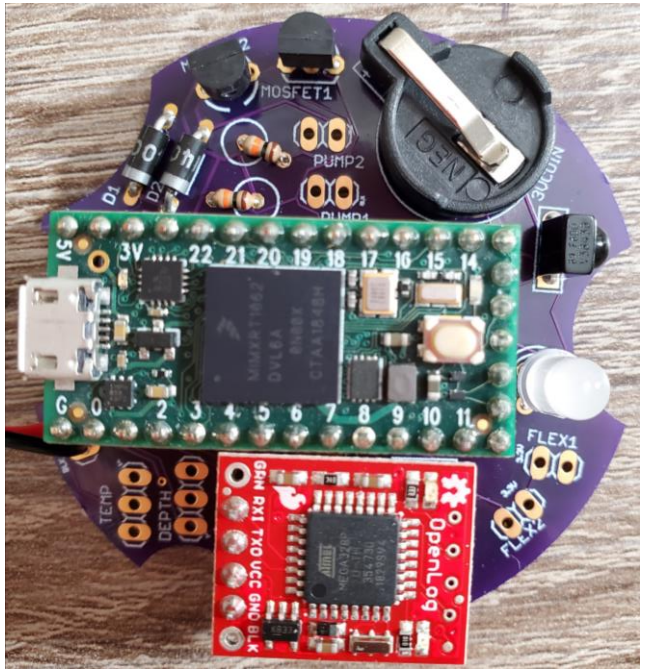
Fabrication: Body Assembly

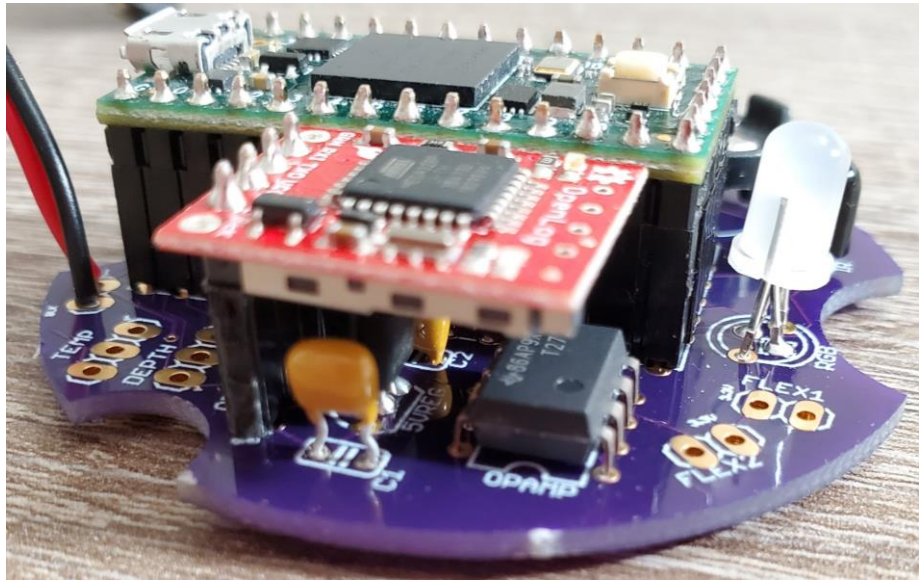
30) Push pump wires through can holes
31) Slide actuators around can, push flex sensor and pump wires through can holes, push tubes over pump outlets
32) Mix epoxy
33) Waterproof flex sensor, pump penetrations
34) Use epoxy to secure actuators onto can
35) Let dry undisturbed for 24-hr

Fabrication: PCB Population

36) Solder resistors, diodes
37) Solder coin cell holder, op amp
38) Solder OpenLog, Teensy headers
39) Solder mosfets, regulator board, IR receiver, RGB LED, 9V clip (recommend crimping wires prior)
40) Solder pump, sensor wires

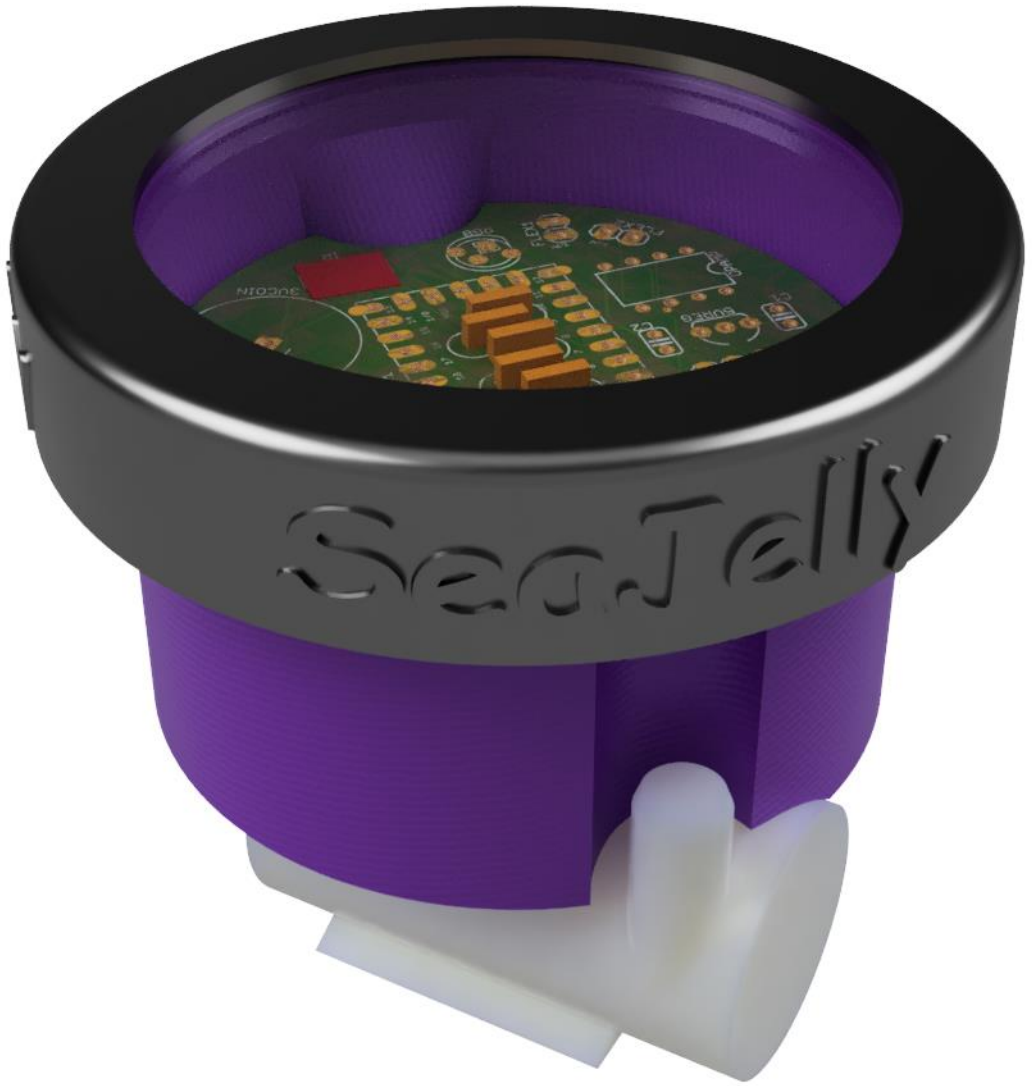




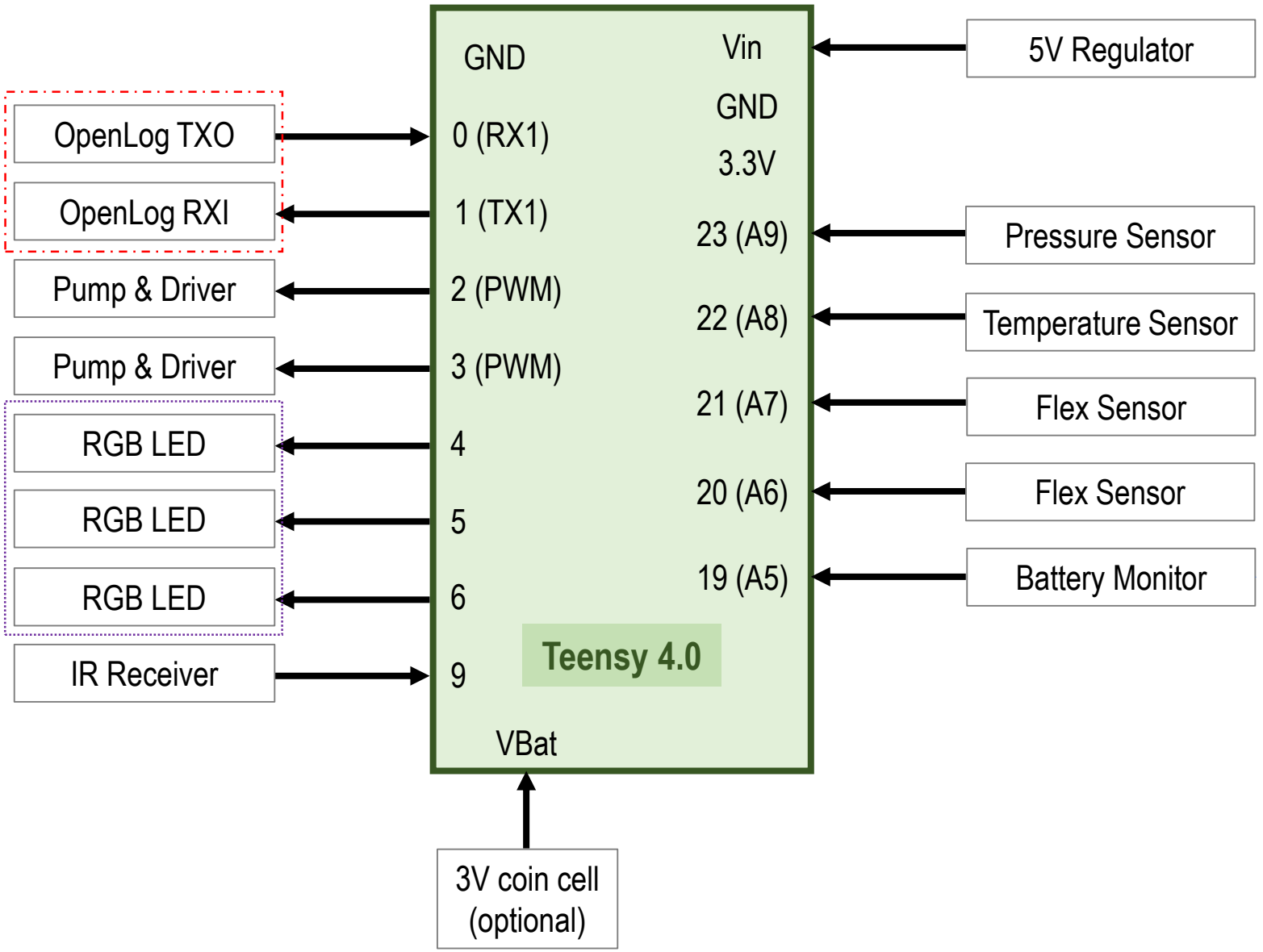


Electronics Background

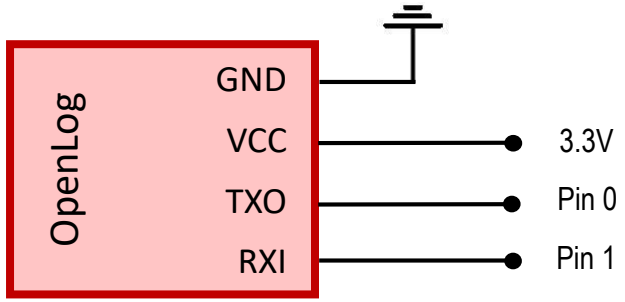
Diagrams, Schematics, Eagle files

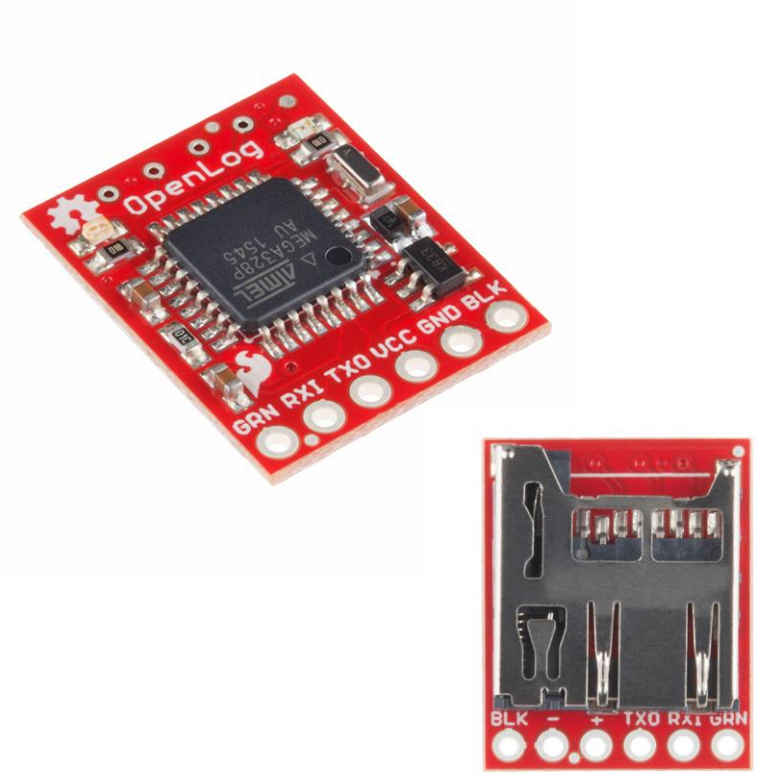


System Block Diagram



Subsystem Schematic – OpenLog

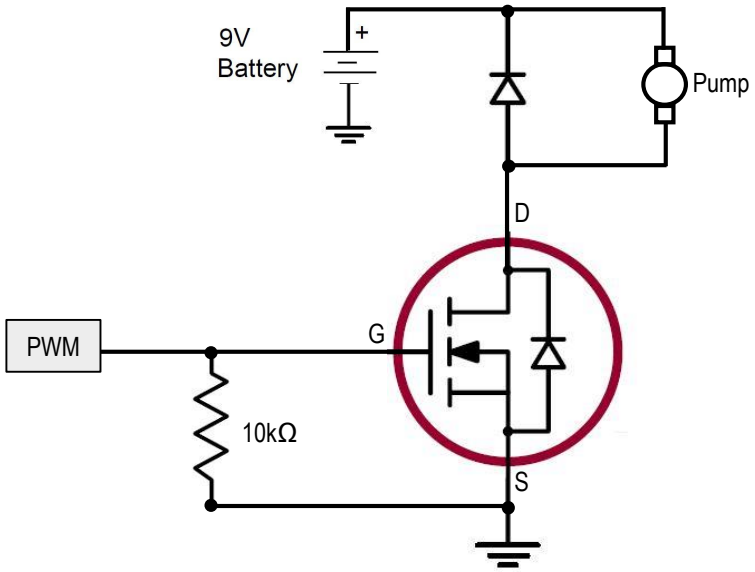


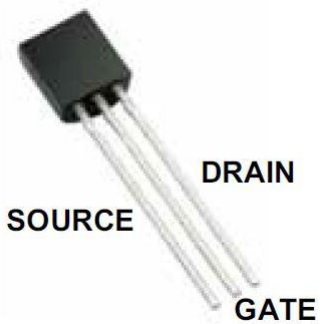


Parts

- Sparkfun OpenLog
- SD card
- Header pins

Subsystem Schematic – Pump & Driver

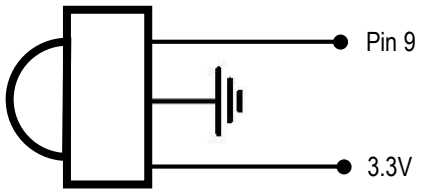


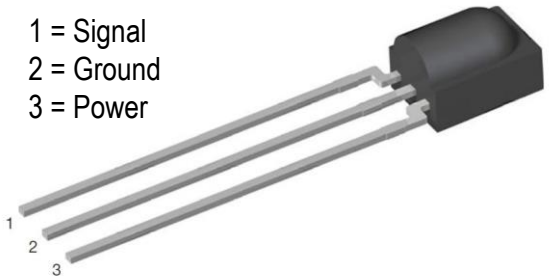


Parts

- 2 x Mini submersible pumps
- 2 x MOSFETs (TN0702)
- 2 x 10kΩ resistors
- 2 x SeaGlide diodes (1N4001)

Subsystem Schematic – IR Receiver

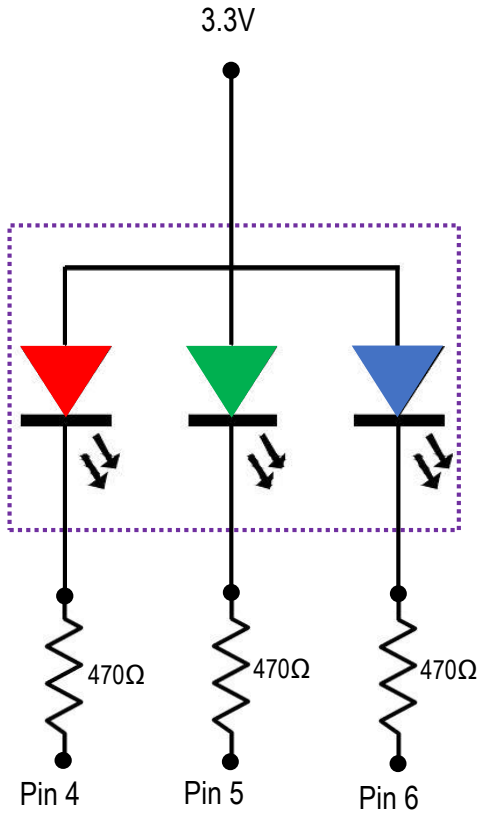


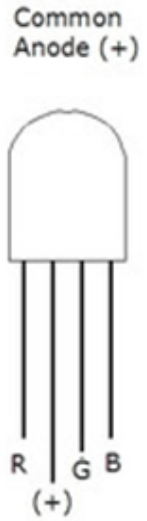


<u>Parts</u>

- IR receiver (TSOP38438)
- IR remote (Adafruit 389)

Subsystem Schematic – RGB LED (Common Anode)

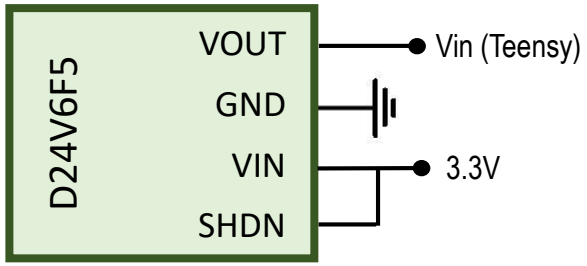




Parts

- RGB LED (Adafruit 159)
- 3 x 470Ω resistors

Subsystem Schematic – 5V Regulator

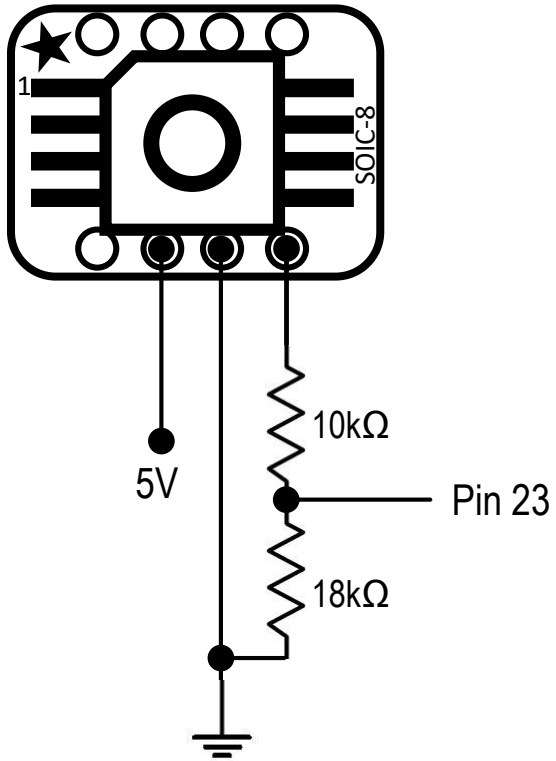


(bottom) (top)

Parts

- 5V regulator (Pololu D24V6F5)
- Header pins

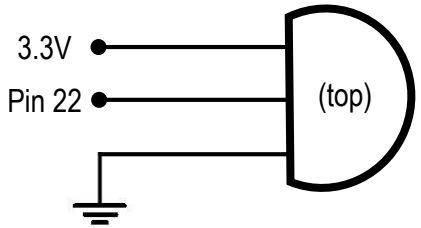
Subsystem Schematic – Pressure Sensor

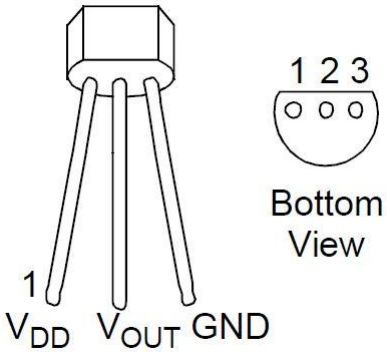


Parts

- Pressure sensor (MPXHZ6250A)
- Breakout board (Adafruit 1212)
- 10kΩ resistor
- 18kΩ resistor

Subsystem Schematic – Temperature Sensor

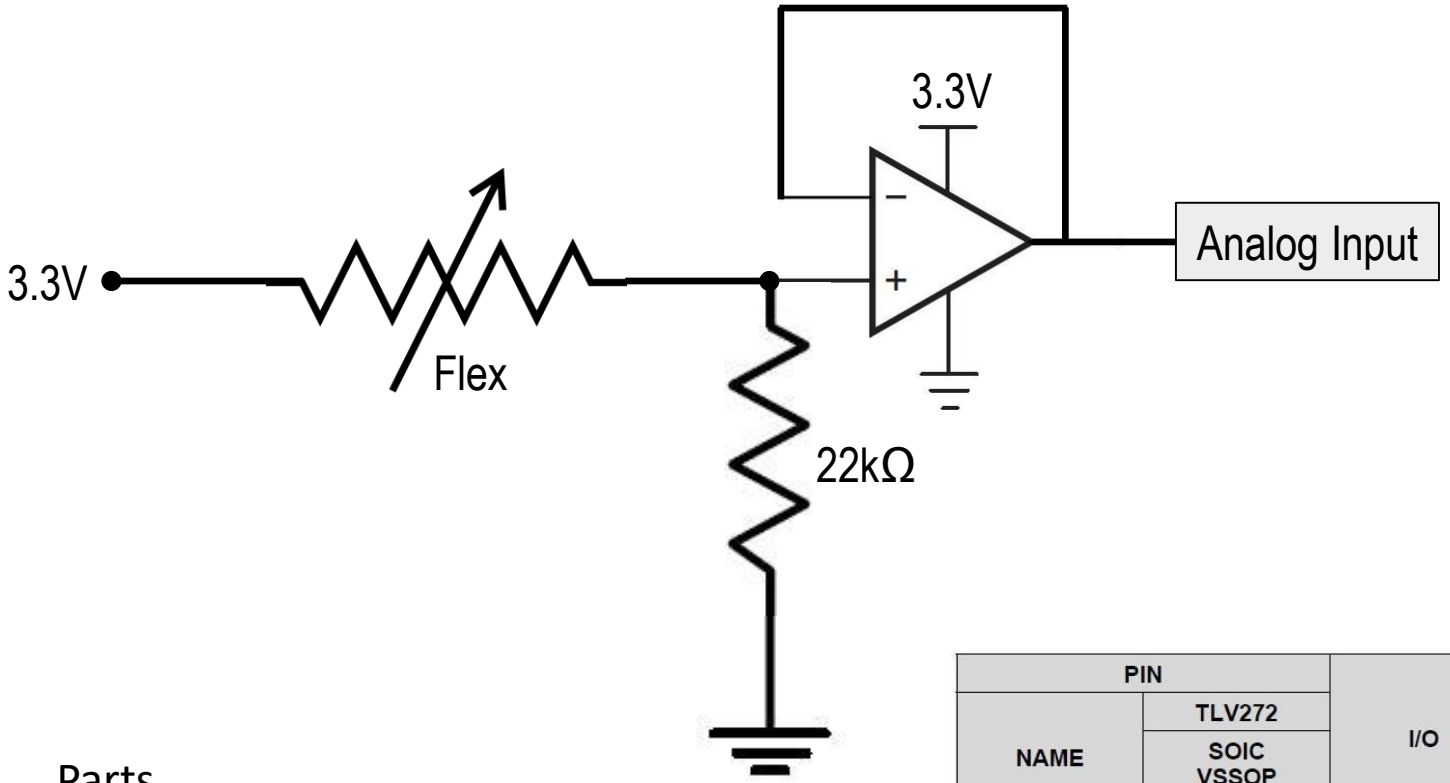


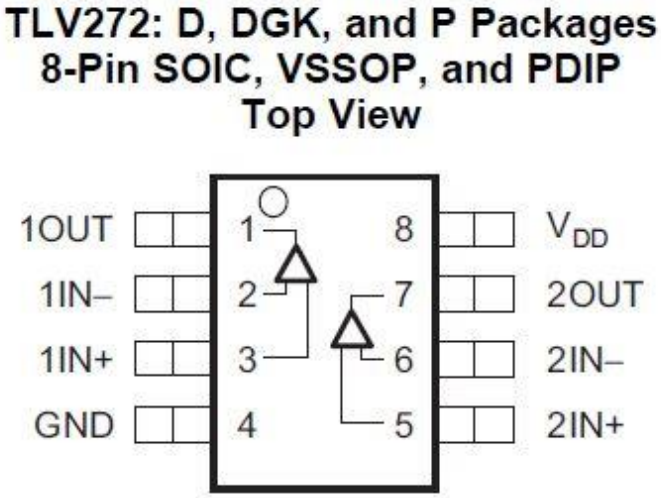


<u>Parts</u>

- Temperature sensor (MCP9701A)

Subsystem Schematic – Flex Sensor





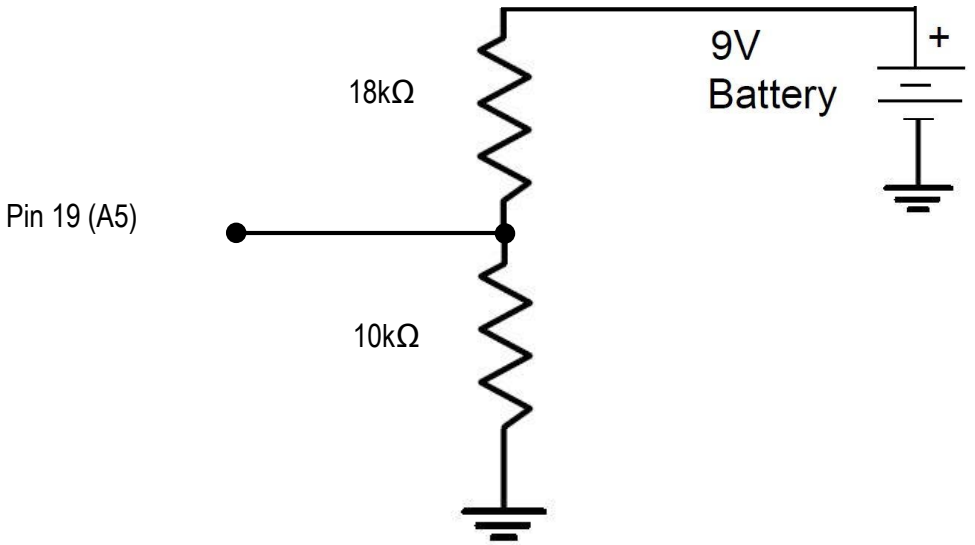
Pin Functions

PIN		I/O	DESCRIPTION
NAME	TLV272 SOIC VSSOP PDIP		
GND	4	—	Negative (lowest) supply or ground (for single-supply operation)
1IN–	2	I	Inverting input, channel 1
1IN+	3	I	Noninverting input, channel 1
2IN–	6	I	Inverting input, channel 2
2IN+	5	I	Noninverting input, channel 2
1OUT	1	O	Output, channel 1
2OUT	7	O	Output, channel 2
V_{DD}	8	—	Positive (highest) supply

Parts

- Velostat
- Copper foil tape
- Wire
- 2 x 22kΩ resistors
- Op amp buffer chip (TLV272)

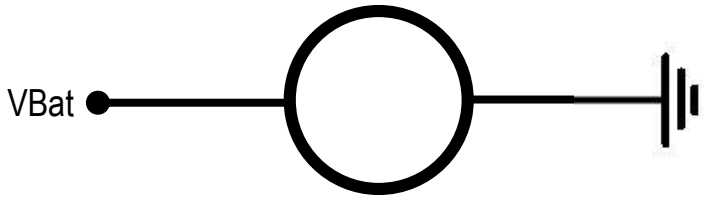
Subsystem Schematic – Battery Monitor



Parts

- 9V battery
- 9V battery harness
- 18kΩ resistor
- 10kΩ resistor

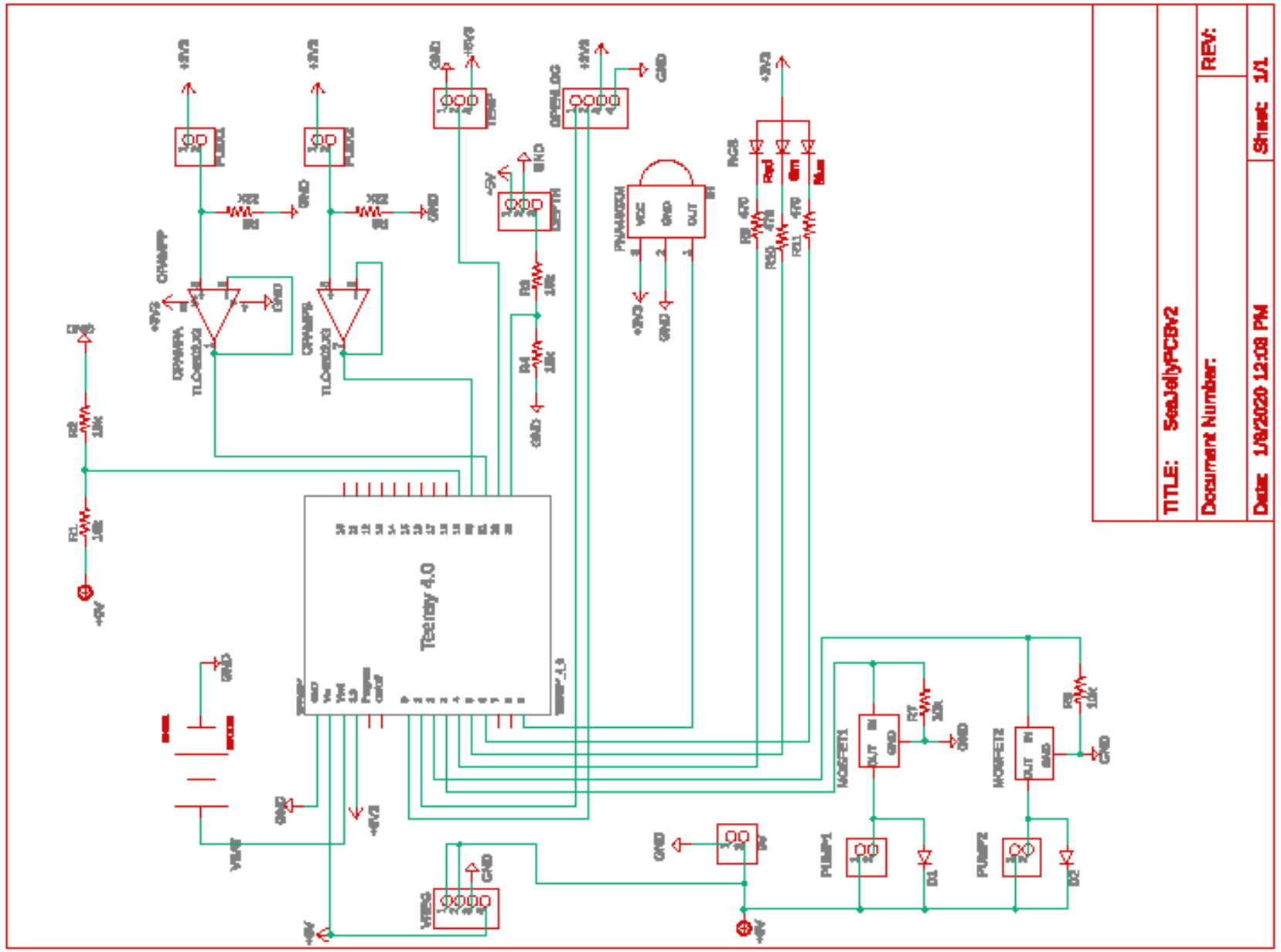
Subsystem Schematic – Real Time Clock Battery



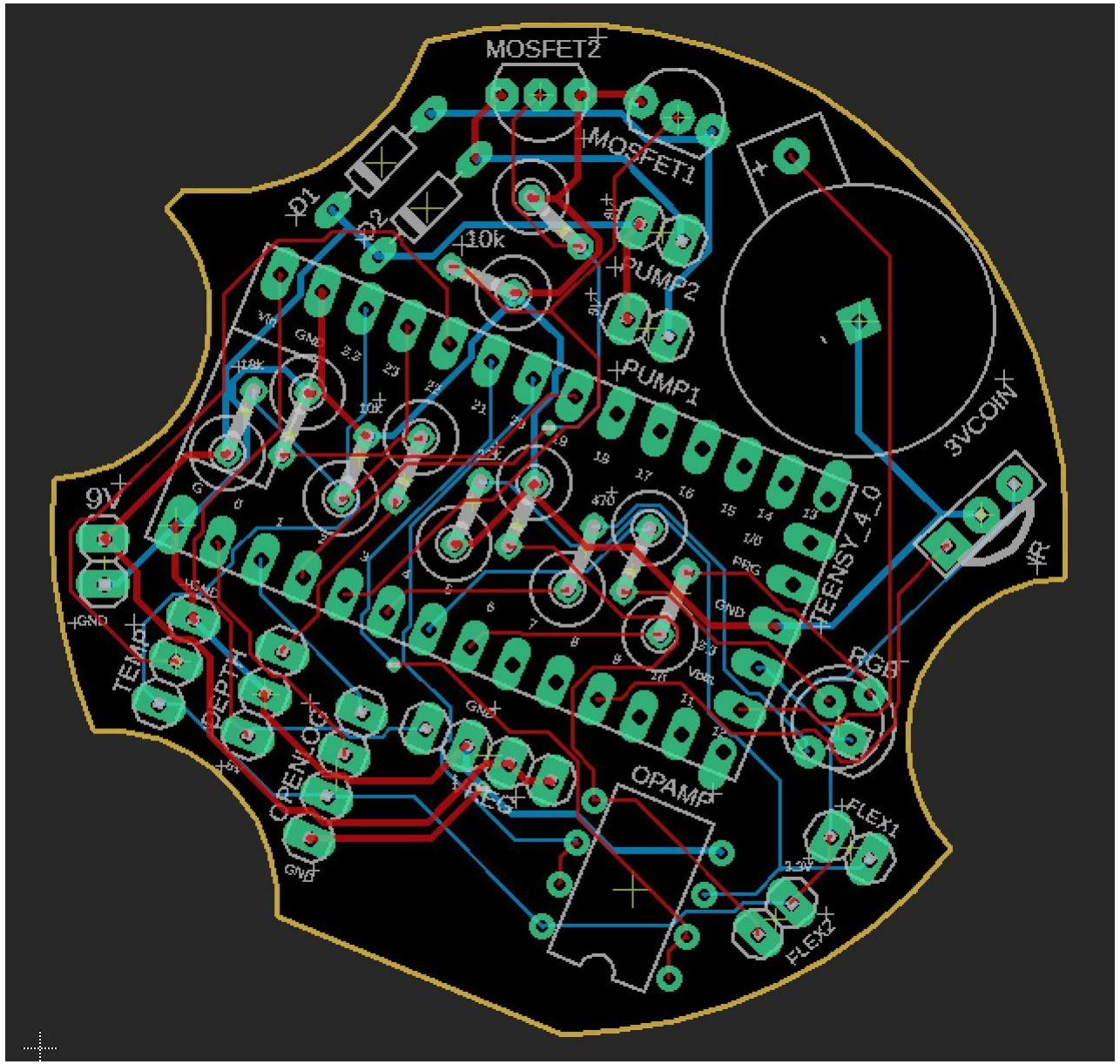
Parts

- 3V coin cell battery (BR1225)
- Coin cell battery holder (BH501)

Eagle Schematic – SeaJelly PCBv2



Eagle Board – SeaJelly PCBv2



Code Explanation

Arduino, Teensyduino

```
#include <IRremote.h>

#define PLAY_PAUSE 0xFD807F
#define STOP_MODE 0xFD609F
#define BUTTON_1 0xFD08F7
#define BUTTON_2 0xFD8877
#define BUTTON_3 0xFD48B7
#define BUTTON_4 0xFD28D7
#define BUTTON_5 0xFDA857
#define BUTTON_6 0xFD6897
#define BUTTON_7 0xFD18E7
#define BUTTON_8 0xFD9867
#define BUTTON_9 0xFD58A7
#define BUTTON_HOLD 0xFFFFFFFF
```

Define IR codes associated with each button on remote

Declare variables and specify data types

```
const float PUMP_OP_VOLTAGE = 6.0; // Ideal pump operating voltage
const unsigned long LOOP_PERIOD = 50; // 20 samples per second

const byte pmp2Pn = 2; // pump 2
const byte pmp1Pn = 3; // pump 1
const byte rPn = 4; // red leg - pin 5 on v1 PCB
const byte gPn = 5; // green leg - pin 6 on v1 PCB
const byte bPn = 6; // blue leg - pin 7 on v1 PCB
// Pin 7 on T4 is used by IR library, interferes with PWM output
const byte rcvPn = 9; // IR receiver - pin 4 on v1 PCB
const byte led = 13; // built-in Teensy 4.0 LED
const byte batPn = 19; //A4, battery monitoring voltage divider
const byte flx2Pn = 20; //A6, flex sensor 2
const byte flx1Pn = 21; // A7, flex sensor 1
const byte tmpPn = 22; // A8, temperature sensor
const byte dpthPn = 23; // A9, pressure sensor

unsigned long pCase = PLAY_PAUSE;
unsigned long sTime, loopTime, actTime;
float batVolt, flex1, flex2, temp, depth;
float pGain = 0;
byte stage;
byte vRecoveryCount = 5; // allow 5 loops for battery to recover
int pump1, pump2;
int p1Write, p2Write;
```

```
IRrecv irrecv(rcvPn);
decode_results results;

void setup() {
  Serial1.begin(9600); // for OpenLog
  pinMode(rPn, OUTPUT);
  pinMode(gPn, OUTPUT);
  pinMode(bPn, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(pmp1Pn, OUTPUT);
  pinMode(pmp2Pn, OUTPUT);
  irrecv.enableIRIn(); // start IR receiver
  pCase = PLAY_PAUSE;
}
```

Initiate IR, serial, and, I/O libraries

List SeaJelly programs as case statements, and use IR remote to select program

```
// SeaJelly Programs
  switch (pCase)
  { // NO DELAY IN SWITCH CASES
    case PLAY_PAUSE: // Idle
    case STOP_MODE:
      color(175,255,175); // purple
      pump1 = 0;
      pump2 = 0;
      stage = 0;
      break;
    case BUTTON_1: // 3sec Swim Cyle (1sec contract, 2sec relax)
      swim(100, 0, 100, 0, 1000, 2000);
      break;
    case BUTTON_2: // empty
      color(0,255,255); // red
      break;
    case BUTTON_3: // empty
      color(0,0,255); // yellow
      break;
    case BUTTON_4: // empty
      color(0,63,52); // pink
      break;
```

```
/// Check battery and set pump PWM output
batVolt = analogRead(batPn) * (9.24 / 1023.0);
  // max 9.24V can be read by 18kOhm & 10kOhm voltage divider with 0-3.3V analog input
if (pump1 == 0 && pump2 == 0) { //if "relaxed"
  vRecoveryCount--;
  if (vRecoveryCount == 0) {  //relaxed for 5 loops and recovered
    if (batVolt > PUMP_OP_VOLTAGE) {
      //if battery voltage is above 6v when resting (pumps are off)
      pGain = (PUMP_OP_VOLTAGE / 100.0) * (255.0 / batVolt);
      // sets pwm scale for 0-6V initial at motor for the next flex cycle
    }
    else { // Low battery voltage
      digitalWrite(led, 1); // turn on built-in led
      pGain = 255.0;
    }
  }
}
else vRecoveryCount = 5;  //reset recovery counter

if (pump1 <= 100 && pump2 <= 100) //if pump values make sense
{
  p1Write = round(pump1 * pGain);
  analogWrite(pmp1Pn, p1Write);
  // PWM (255 = battery voltage)
  p2Write = round(pump2 * pGain);
  analogWrite(pmp2Pn, p2Write);
}
else {
  analogWrite(pmp1Pn, 0);
  analogWrite(pmp2Pn, 0);
}
```

Use battery feedback to consistently output 6V to pumps

```
/// Collect sensor data
flex1 = analogRead(flx1Pn);
flex2 = analogRead(flx2Pn);
temp = analogRead(tmpPn);
depth = analogRead(dpthPn);

/// Record data to OpenLog
Serial1.print(pCase, HEX);
Serial1.print(", ");
Serial1.print(pump1);
Serial1.print(", ");
Serial1.print(pump2);
Serial1.print(", ");
Serial1.print(flex1);
Serial1.print(", ");
Serial1.print(flex2);
Serial1.print(", ");
Serial1.print(temp);
Serial1.print(", ");
Serial1.print(depth);
Serial1.print(", ");
Serial1.println(batVolt);
```

Collect and record sensor data

Maintain a constant loop period

```
/// LOOP MANAGER
loopTime = millis() - sTime;
if (loopTime < LOOP_PERIOD) delay(LOOP_PERIOD - loopTime);
else {
  // report error
  Serial1.print("Over loop error  ");
  Serial1.println(loopTime);
```

```
int swim(int c1, int r1, int c2, int r2, int cT, int rT)
/* c1 [% pump 1 power contract(0-100)], r1 [% pump 1 power relax(0-100)]
 *  c2 [% pump 2 power contract(0-100)], r2 [% pump 2 power relax(0-100)]
 *  cT [msec contract time], rt [msec relax Time]
 *  output = % pump power
*/
{
  switch (stage)
  {
    case 0: //intitialize
      actTime = millis();
      stage++;

    case 1: // contract
      color(255,0,255); // green
      pump1 = c1;
      pump2 = c2;
      if (int(millis() - actTime) > cT)
      {
        stage++;
      }
      break;
    case 2: //relax
      color(255,255,0); // blue
      pump1 = r1;
      pump2 = r2;
      if (int(millis() - actTime) > (cT + rT))
      {
        stage = 0;
      }
      break;
  }
}
```

Function for timed swimming

Function for changing RGB LED color

```
void color(int r, int g, int b)
// for common anode RGB LED
{
      analogWrite(rPn, r);
      analogWrite(gPn, g);
      analogWrite(bPn, b);
}
```

Beyond Starter Code

- Implement feedback control with depth sensor
- Implement feedback control with flex sensors
- Time stamp data logging with real-time clock

References

JenniFish Publications

- https://fau.digital.flvc.org/islandora/object/fau%3A33675/datastream/OBJ/view/Self-Contained_Soft_Robotic_Jellyfish_with_Water-Filled_Bending_Actuators_and_Positional_Feedback_Control.pdf
- https://iopscience.iop.org/article/10.1088/1748-3190/aadcb3

Related STEM

- http://seaglide.org/
- https://www.navsea.navy.mil/Home/Warfare-Centers/NSWC-Carderock/STEM/

Software Links

- https://www.autodesk.com/products/eagle/free-download
- https://www.autodesk.com/products/fusion-360/free-trial
- https://www.arduino.cc/en/Main/Software
- https://www.pjrc.com/teensy/teensyduino.html

Tutorials

- https://www.dropbox.com/sh/uxjsedfrzmr9kq3/AABnqPHgvnL1GjgJhl7JfRfKa/Arduino%20lessons?dl=0&subfolder_nav_tracking=1
- https://learn.adafruit.com/ktowns-ultimate-creating-parts-in-eagle-tutorial/understanding-parts-in-eagle
- https://learn.sparkfun.com/tutorials/using-eagle-schematic
- https://www.youtube.com/channel/UCjPWeOcXkMIOA_bd4LbivGg
- https://www.autodesk.com/products/eagle/learning-center
- https://f360ap.autodesk.com/courses/getting-started-in-fusion-360
- https://www.youtube.com/user/AutodeskFusion360