CP3

Coronavirus Makers PAPR Task Force

Powered Air Purifying Respirators for healthcare personnel

Work safe
Work comfortably
Work better

#EUvsVirus challenge

can you hack it?

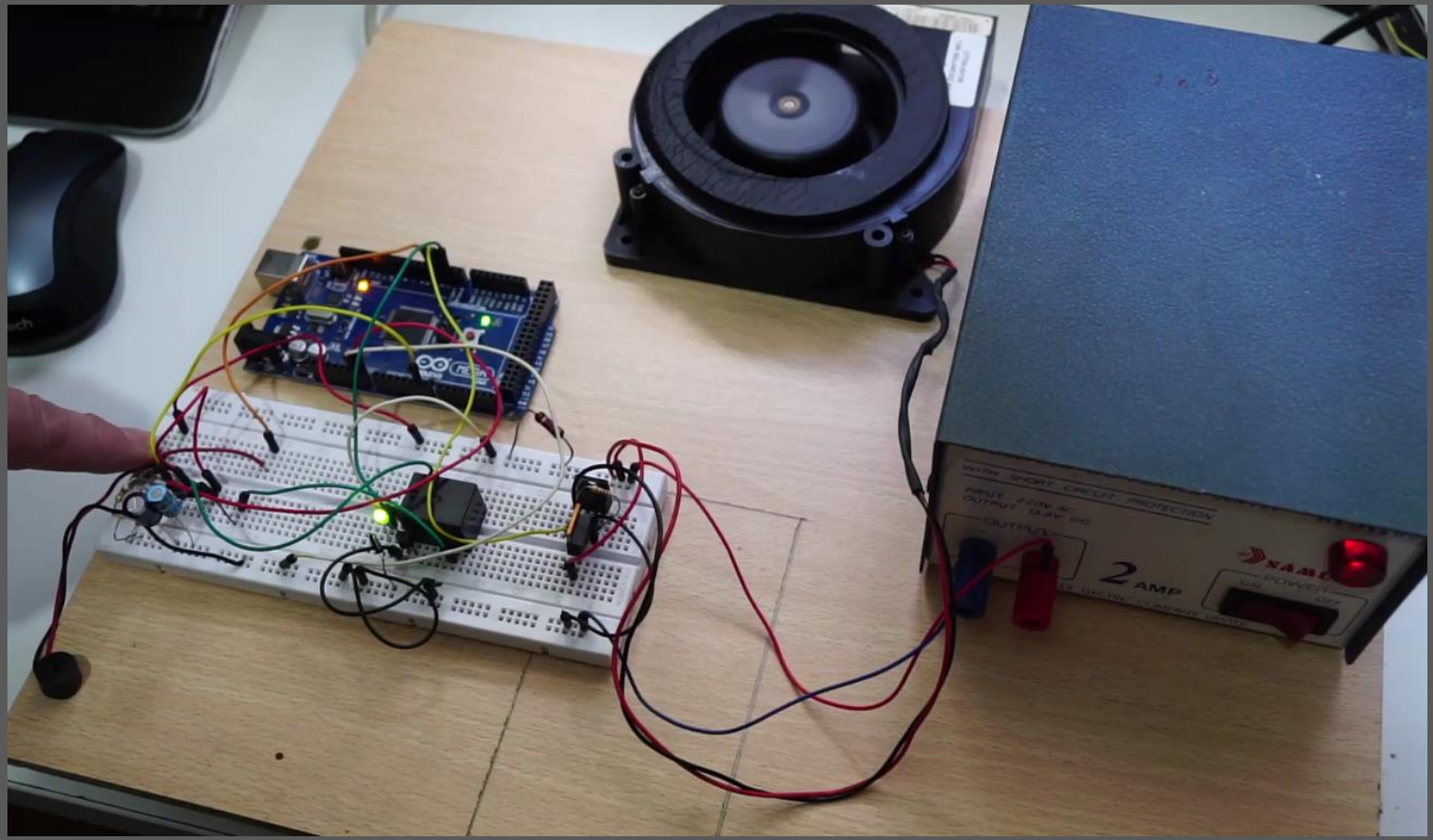
PROGRESS

Prototype

- Software function test
- P.A.P.R Arduino code
- Electromagnetic Isolation
- Caudal detection
- Fabric pattern
- Helmet support improvement
- Definitive blower fan

License

Software function testing



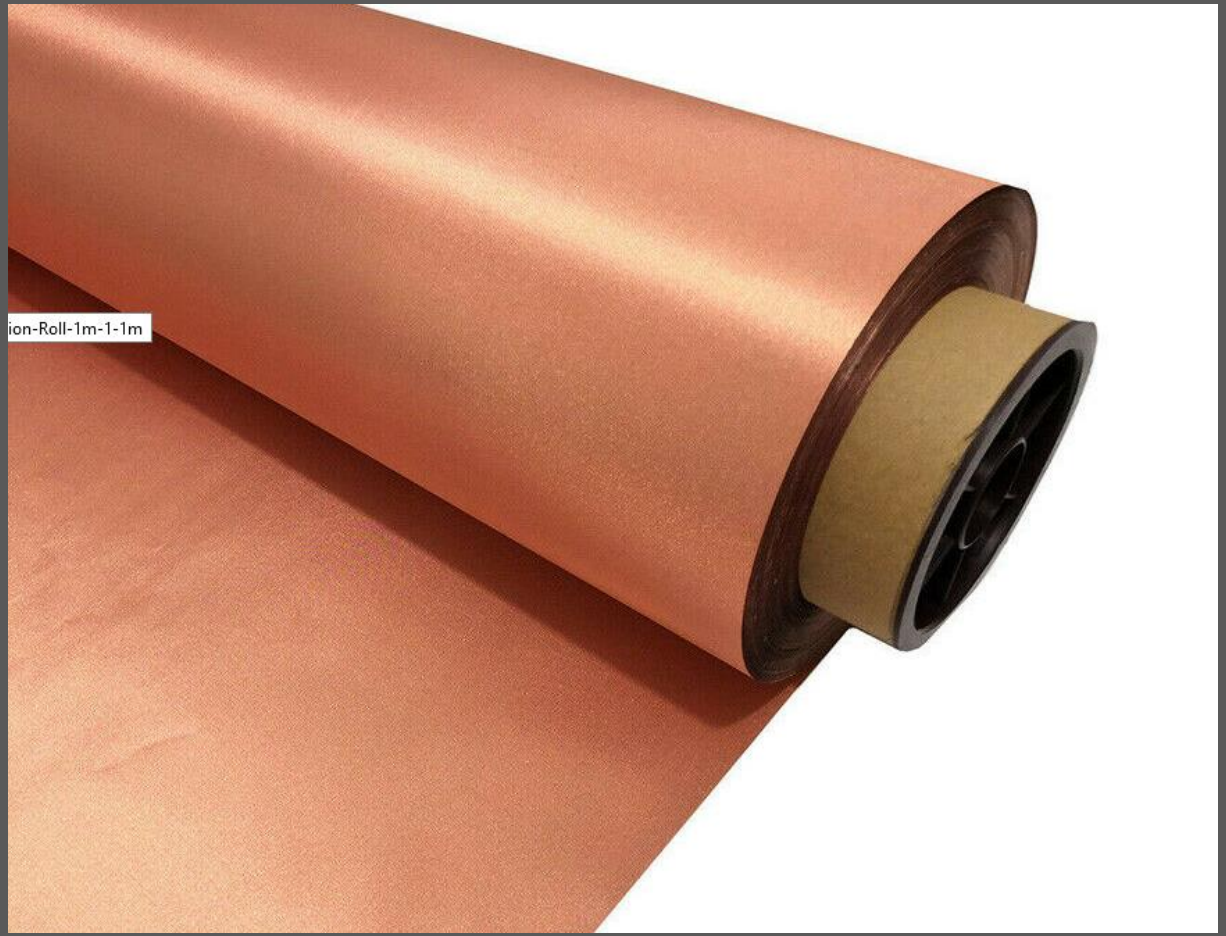
PAPR Software coding v1 (alpha version)

```
//Output Pin layout
const int motor = 3; // PWM pin (Arduino Nano)
const int bat20 = 4;
const int bat40 = 5;
const int bat60 = 6;
const int bat80 = 7;
const int bat100 = 8;
const int onled = 12;
const int runled = 10;
const int alarmled = 11;
const int buzzer = 9; // PWM pin (Arduino Nano)


//Input Pin layout

const int battery = A5;
const int conmutA = A0;
const int conmutB = A1;
const int conmutC = A2;
const int conmutD = A3;
const int conmutE = A4;
const int flowsensor = A6;


//Constant variables (do not change)
const float PWDIVR1 = 100000; // R1 from the Power Divider from the battery (100 kOhms)
const float PWDIVR2 = 10000; // R2 from the Power Divider from the battery (10 kOhms)
const int minflow = 100;
const int medflow = 200;
const int highflow = 300;
const int maxflow = 400;


//Calibration variables (modify with your hardware datasheet)
const float batvolt100 = 20.0; // Voltage of the full-charge battery (Volts)
const float batvolt0 = 18.0; // Voltage of the empty-charge battery (Volts)
const int admflowbias = 20;// admisible real flow % missmatch from the theoric flow
const float lowbatlim = 20.0; // % low battery limit before battery alarm
const float vcc = 12.0; // Voltage supply of the system after the battery step-down
const float flowpervolt = 100.0;// air flow (l/min) per each volt of supply
const int flowconstant = 100; //CONSTANTE PROVISIONAL A DETERMINAR POR EL SENSOR DE CAUDAL DE AIRE

//Variables
float batvolt = 0.0;
float batlvl = 100;
int teoricflow = 0;
int outpwm = 0;
float outVolt = 0.0;
int OutputPulseCounter1 = 500;
int OutputPulseCounter2 = 0;
int realflow = 0;


void setup() {

  Serial.begin(9600);
  pinMode(motor, OUTPUT);
  pinMode(bat20, OUTPUT);
  pinMode(bat40, OUTPUT);
  pinMode(bat60, OUTPUT);
  pinMode(bat80, OUTPUT);
  pinMode(bat100, OUTPUT);
  pinMode(onled, OUTPUT);
  pinMode(runled, OUTPUT);
  pinMode(alarmled, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(conmutA,INPUT);
  pinMode(conmutB,INPUT);
  pinMode(conmutC,INPUT);
  pinMode(conmutD,INPUT);
  pinMode(conmutE,INPUT);
```

```
void loop() {

//PHASE 1: INPUT READING

  batvolt = ((analogRead(battery) * 5.0) / 1024.0)/(PWDIVR2/(PWDIVR1+PWDIVR2)); // Reads and Calculates the battery real voltage from de voltage divider
  batlvl = map(batvolt, batvolt0, batvolt100,0,100); // Determines the current level of charge of the battery (%charge)

  realflow = flowconstant * (analogRead(flowsensor)); //FALTA DETERMINAR EL TIPO DE SENSOR

  //Determines the desired air flow from the commuter inputs.
  if (digitalRead(conmutB)==HIGH){
                                  teoricflow=minflow;
                                  }
                                  else if (digitalRead(conmutC)==HIGH){
                                  teoricflow=medflow;
                                  }
                                  else if (digitalRead(conmutD)==HIGH){
                                  teoricflow=highflow;
                                  }
                                  else if (digitalRead(conmutE)==HIGH){
                                  teoricflow=highflow;
                                  }
                                  else {
                                  teoricflow=0;
                                  }

//PHASE 2: AIRFLOW VALIDATION

  if ((teoricflow + (admflowbias*teoricflow/100)) > realflow) {
                                  digitalWrite(runled, LOW);
                                  digitalWrite(alarmled, HIGH);
                                  tone(buzzer,150);
  }

//PHASE 3: AIRFLOW SETUP  // leer voltaje del sensor de flujo de aire (por determinar)

  outVolt = teoricflow/flowpervolt; //calculates the teorical voltage for the selected air flow
  outpwm = map(outVolt, 0, vcc, 0, 255); //calculates the PWM value for the selected air flow
  analogWrite (motor,outpwm);  //activates the PWM output to the MOSFET
  digitalWrite(runled,HIGH); //on Green led "RUN"
  digitalWrite(onled,LOW); //off Red led "ON"
```

```
//PHASE 4: BATTERY INTERFACE UPDATE

 if (batlvl>=80){
                                  digitalWrite(bat20, HIGH);
                                  digitalWrite(bat40, HIGH);
                                  digitalWrite(bat60, HIGH);
                                  digitalWrite(bat80, HIGH);
                                  digitalWrite(bat100, HIGH);
 }

 else if (batlvl<80){
                                  digitalWrite(bat20, HIGH);
                                  digitalWrite(bat40, HIGH);
                                  digitalWrite(bat60, HIGH);
                                  digitalWrite(bat80, HIGH);
                                  digitalWrite(bat100, LOW);
 }

 else if (batlvl<60){
                                  digitalWrite(bat20, HIGH);
                                  digitalWrite(bat40, HIGH);
                                  digitalWrite(bat60, HIGH);
                                  digitalWrite(bat80, LOW);
                                  digitalWrite(bat100, LOW);
 }

  else if (batlvl<40){
                                  digitalWrite(bat20, HIGH);
                                  digitalWrite(bat40, HIGH);
                                  digitalWrite(bat60, LOW);
                                  digitalWrite(bat80, LOW);
                                  digitalWrite(bat100, LOW);
 }
  else if (batlvl<20){
                                  digitalWrite(bat20, HIGH);
                                  digitalWrite(bat40, LOW);
                                  digitalWrite(bat60, LOW);
                                  digitalWrite(bat80, LOW);
                                  digitalWrite(bat100, LOW);
 }

 if (batlvl<lowbatlim){  //If batterylevel is below the limit, an intermitent alarm blinking-tone starts

  if (OutputPulseCounter1 > 0) {

                                  digitalWrite(alarmled, HIGH);
                                  tone (buzzer, 150);
                                  OutputPulseCounter1--;
                                  OutputPulseCounter2 = 500;
  }

   else if (OutputPulseCounter2){

                                  digitalWrite(alarmled, LOW);
                                  noTone(buzzer);
                                  OutputPulseCounter2--;
                                  OutputPulseCounter1 = 500;

 }
 }
 }
```

Electromagnetic Isolation



EMG Test

Pure Copper Fabric EMF
RFID Signal Blocking
Radiation Protection Roll

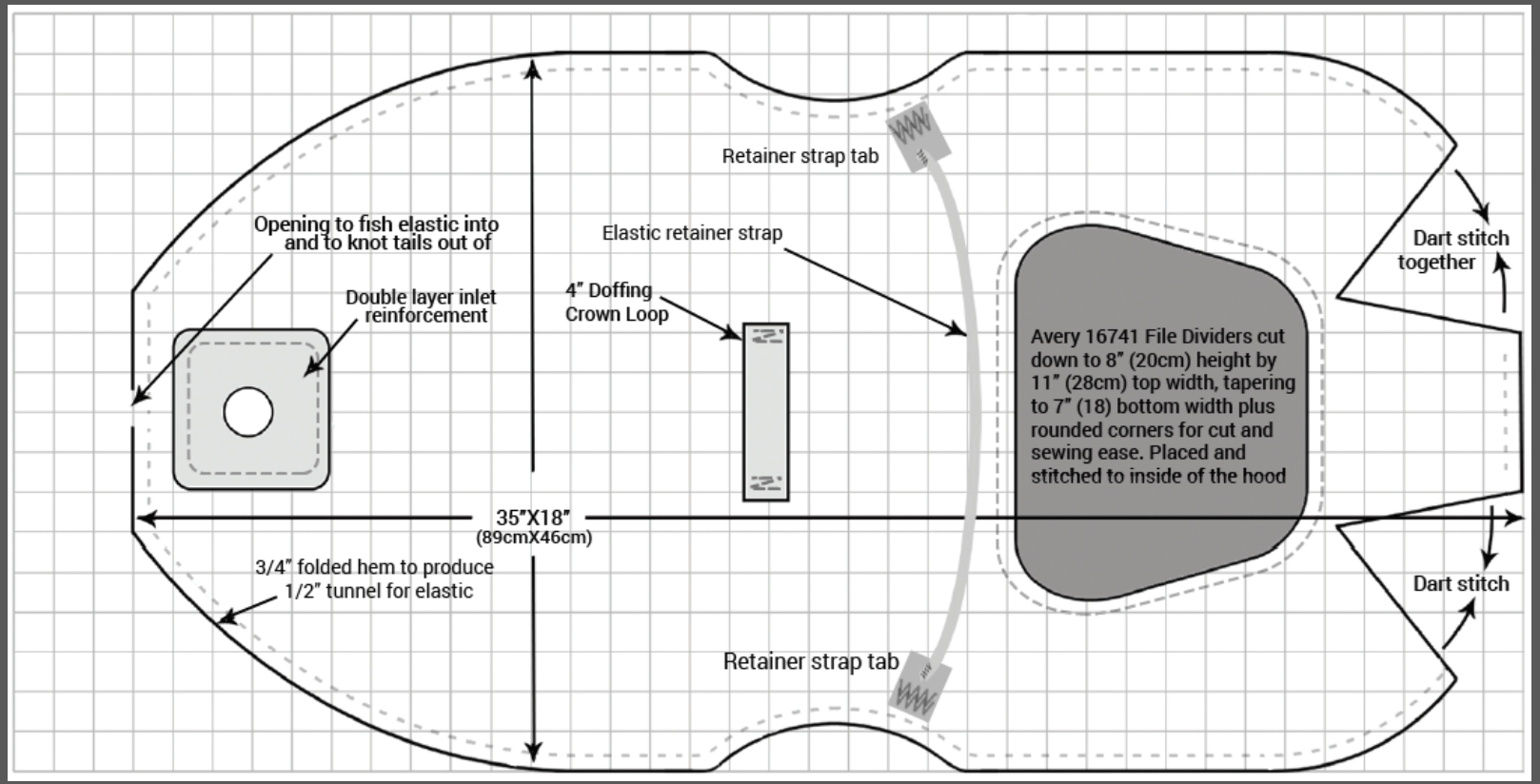
Caudal detection

MPX5010DP
Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

Minimum Caudal: 120 L/min
Maximum Pressure on overhang: 5 mbar

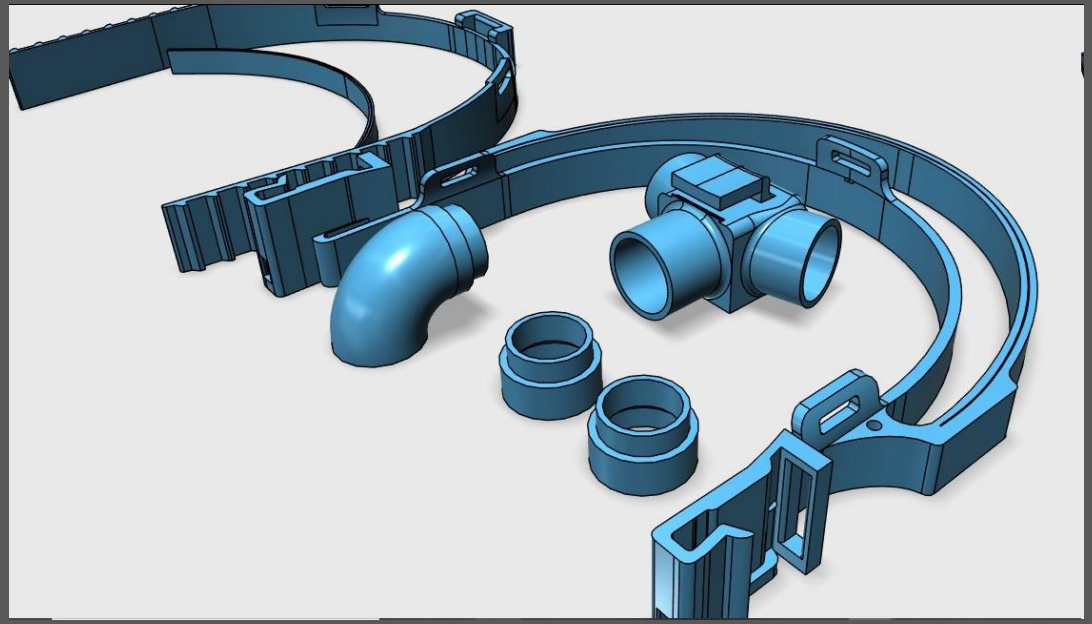
Additionally
Max weight equipment: 5 kg
Max weight on head: 1.5 kg (included on max weight equipment)

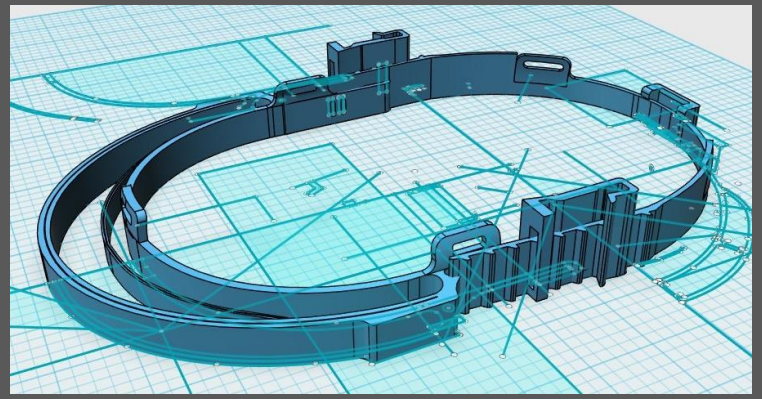
Fabric Pattern



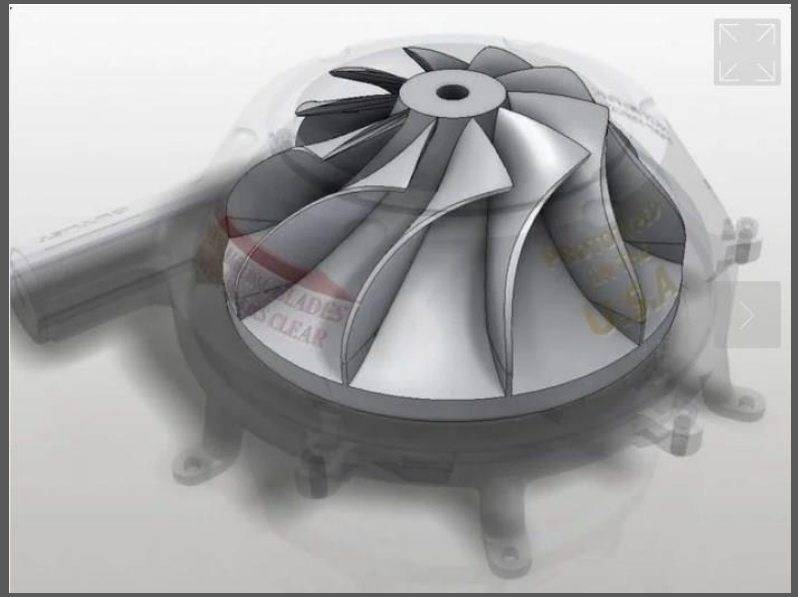
Design from https://open-mask.org/

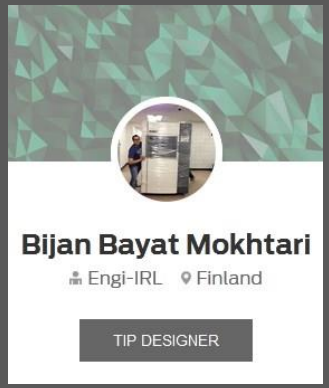
Helmet support improvement





Definitive 3D printed blower fan





After researching and tests we finally use this CC design from Bijan Bayat Mokhtari

LICENSE

Hardware

The **CERN Open Hardware Licence** (OHL or CERN OHL) is a license used in open-source hardware projects

Software

The **GNU General Public License V3** (GNU GPL or GPL) is a series of widely used free software licenses that guarantee end users the freedom to run, study, share, and modify the software

PROBLEMS

- Fabric pattern design
  - No idea about fabric design, open-mask.org as an option
- Licenses
  - Open hardware and software licenses proposed
- Electronics
  - Lack of flow sensor integration
- MVP
  - Can’t join different parts of the assembly due to is a virtual hackathon

PLANS

- Fabric pattern design
  - Need a fabric designer to help with pattern, or manage with open-mask by CC License

- Licenses
  - Check Open-source + commercial

- Video for deadline