

TRANSFORMATION D'UN RESPIRATEUR FONCTIONNANT EN CIRCUIT OUVERT EN UN RESPIRATEUR FONCTIONNANT EN CIRCUIT FERMÉ

Louis Vauterin
Numéro de candidat: 28393



APPROCHE DU SUJET

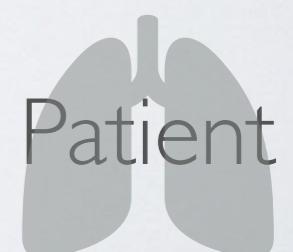
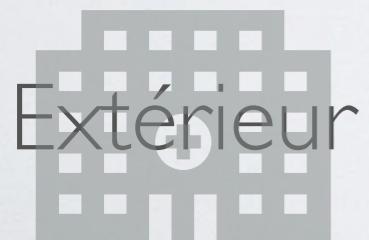
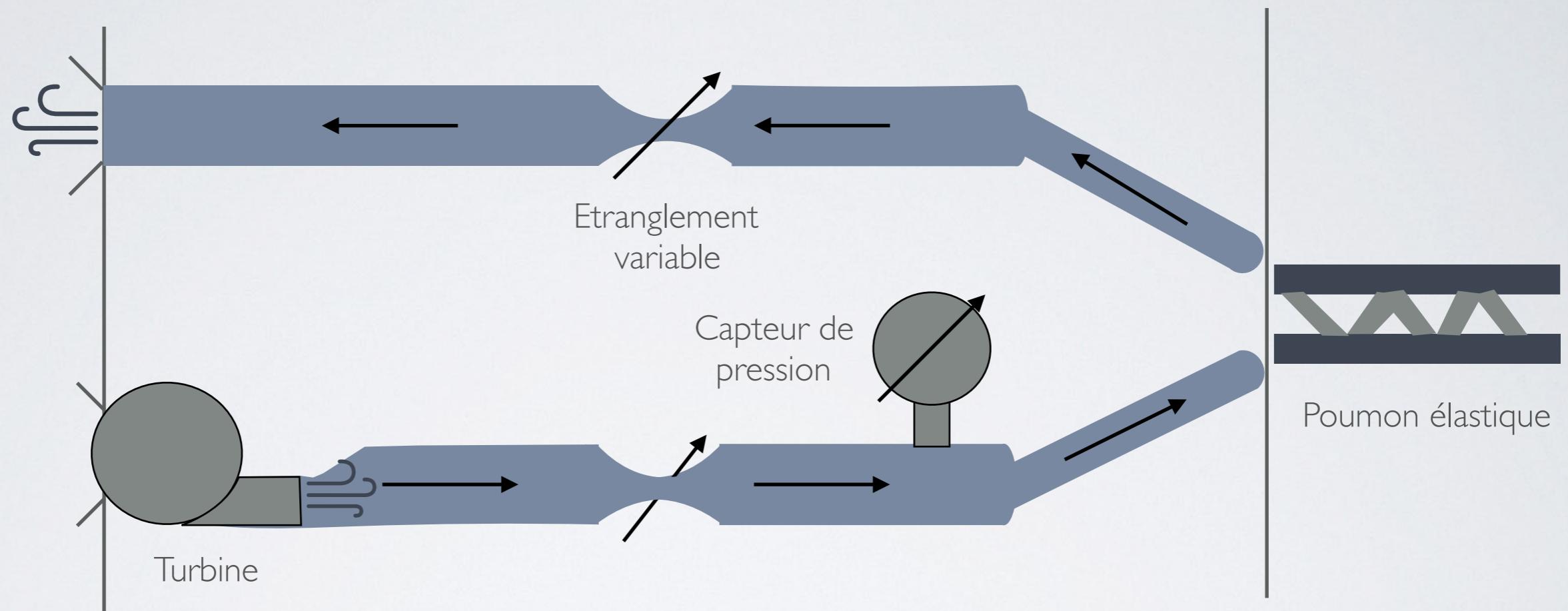
- I. Présentation d'un respirateur artificiel: le Makair.
- II. Fabrication d'une maquette fonctionnant en circuit fermé.
- III. Répondre aux attentes demandées d'un respirateur artificiel.

LE MAKAIR

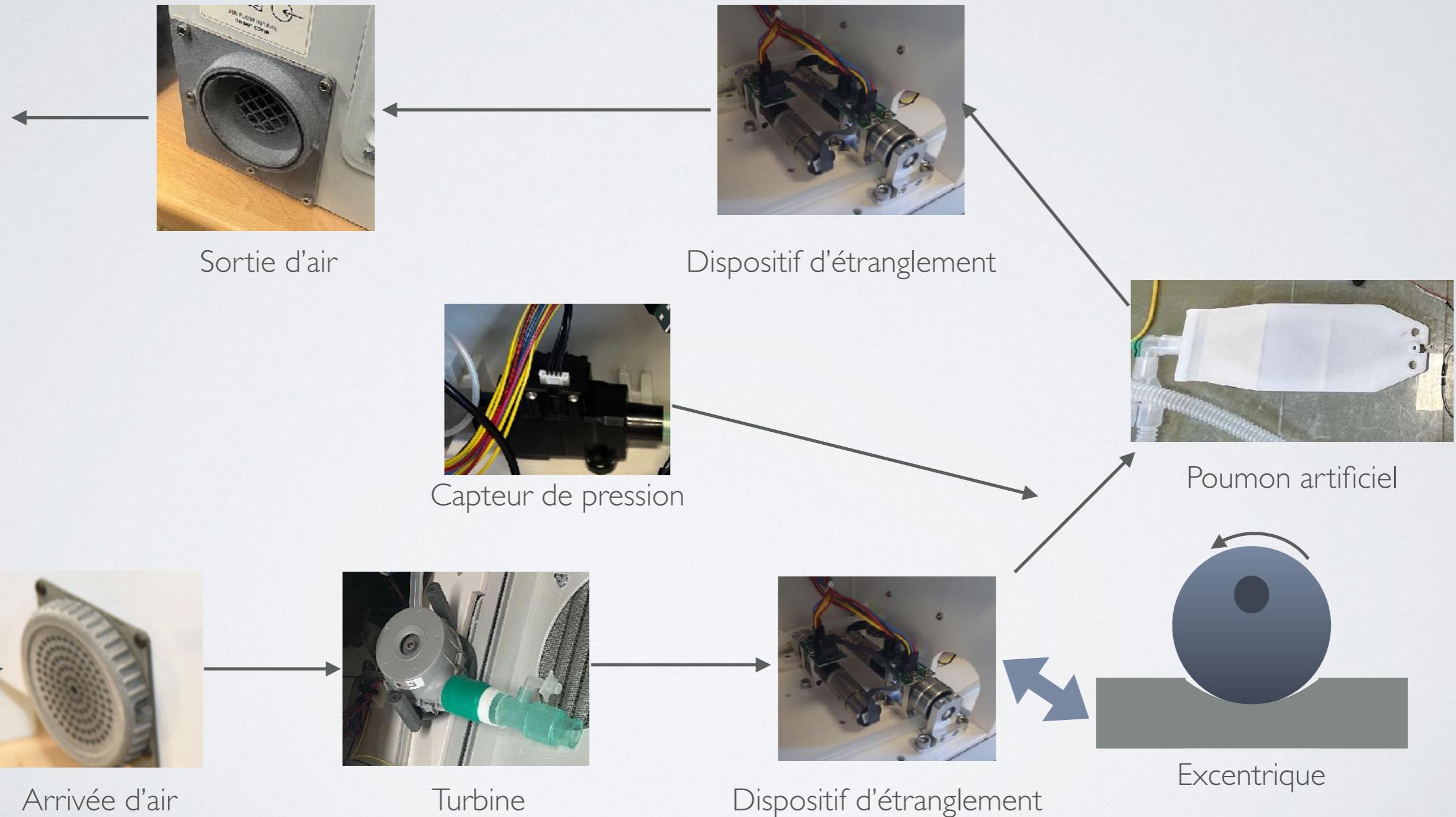


- Construit pour répondre aux besoins des hôpitaux durant la crise du Covid.
- But: respirateur peu coûteux, facile à transporter et performant.
- Nous nous en sommes procuré un en contactant l'équipe d'ingénieur.

SCHÉMA PNEUMATIQUE



LES DIFFÉRENTS COMPOSANTS



PEUT-ON FAIRE FONCTIONNER LE MAKAIR EN CIRCUIT FERMÉ ?



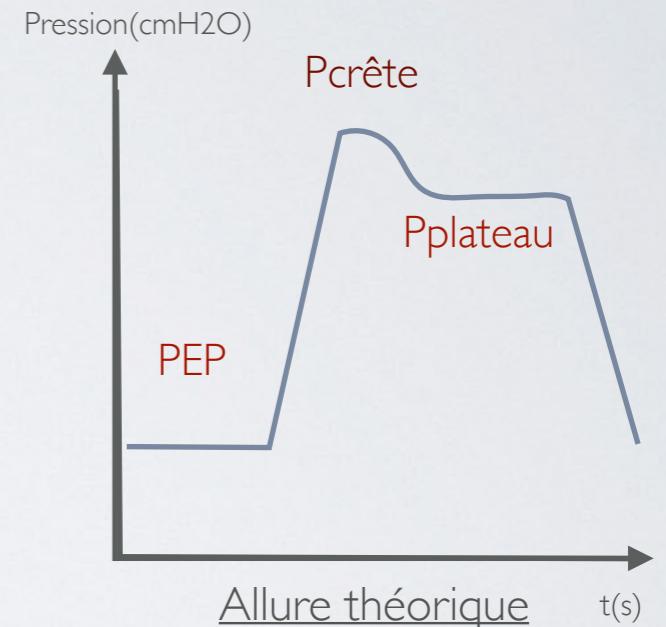
Courbe de pression visible sur l'écran du Makair

Pièces réalisées sur Solidworks pour fermer le circuit

PEUT-ON FAIRE FONCTIONNER LE RESPIRATEUR EN CIRCUIT FERMÉ ?

Cahier des charges

Critères	Exigences	Validation
Temps de montée	< 1s	OUI
Temps de descente a la PEP	< 3s	NON trop long
PEP	Réglable et toujours >0cmH2O	NON
P_plateau	Réglable entre 20 et 25cmH2O	OUI
Fonctionnement pour différents cycles	Cycle normal et détresse respiratoire	NON
Pmax	<30cmH2O	OUI



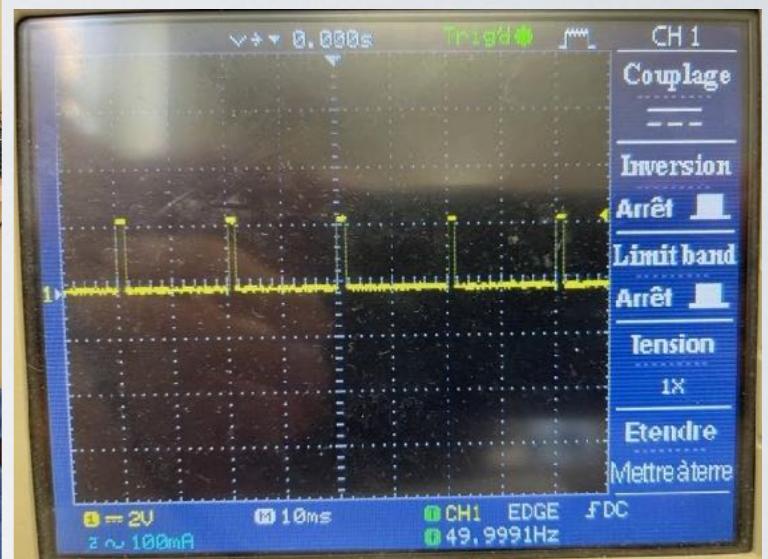
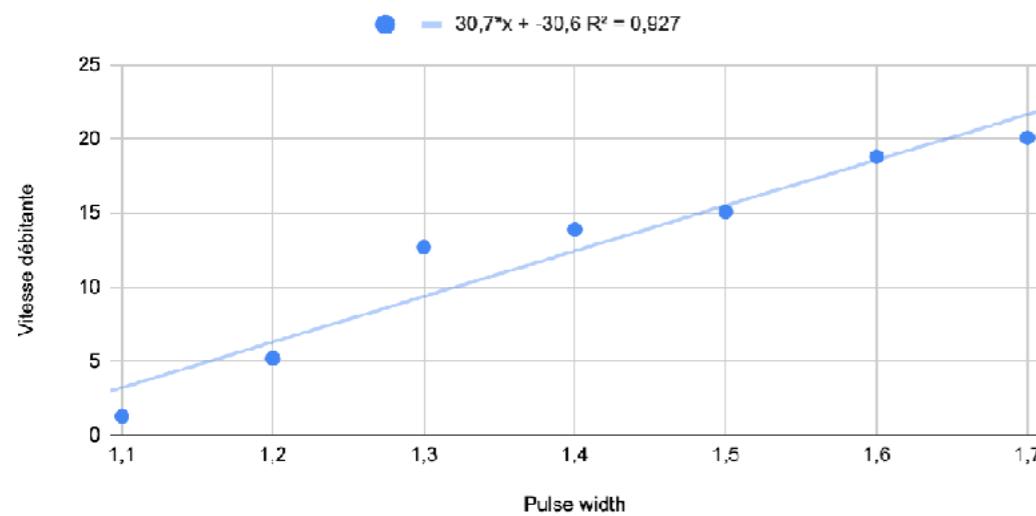
Courbe de pression du Makair en circuit fermé

Poumons qui ne se vident pas assez vite = risque de barotraumatismes dangereux

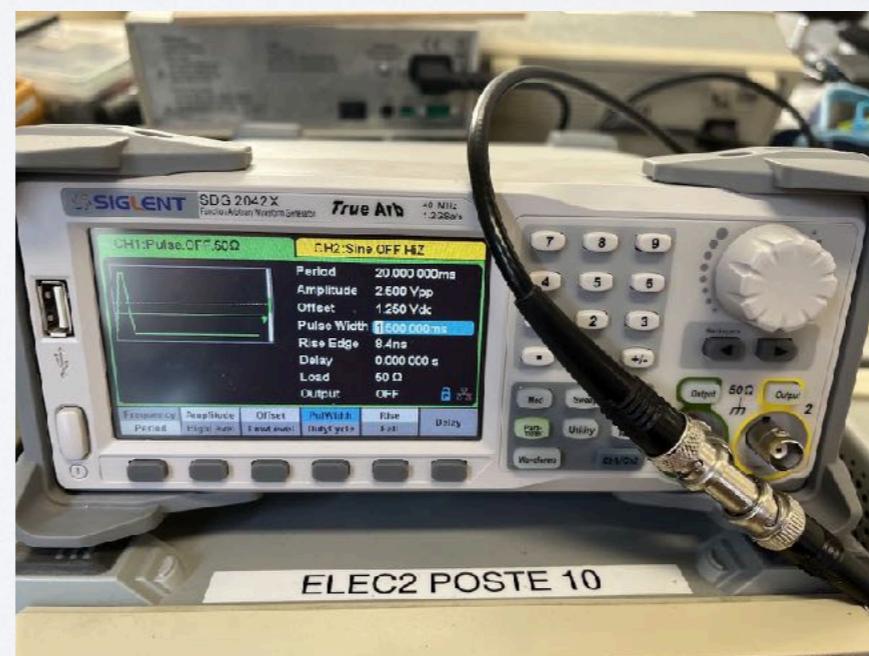
COMMENT CONSTRUIRE UN RESPIRATEUR ARTIFICIEL?

A) Envoyer de l'air

Vitesse débitante en fonction du pulse width



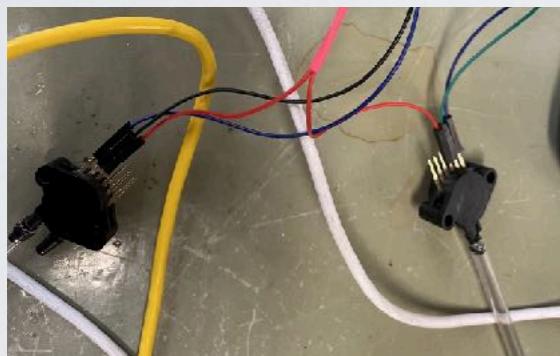
Retrouver le signal envoyé à la turbine avec des sondes branchées sur l'appareil et une carte d'acquisition.



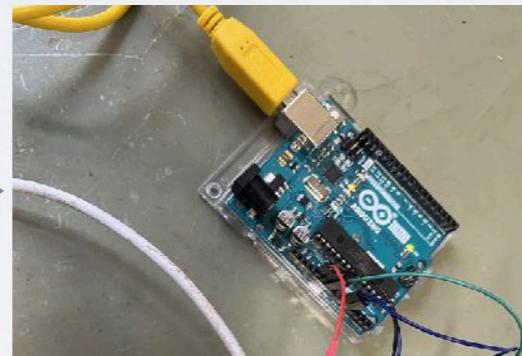
Signal reproduit envoyé au à la turbine.

COMMENT CONSTRUIRE UN RESPIRATEUR ARTIFICIEL?

B) Mesurer la pression



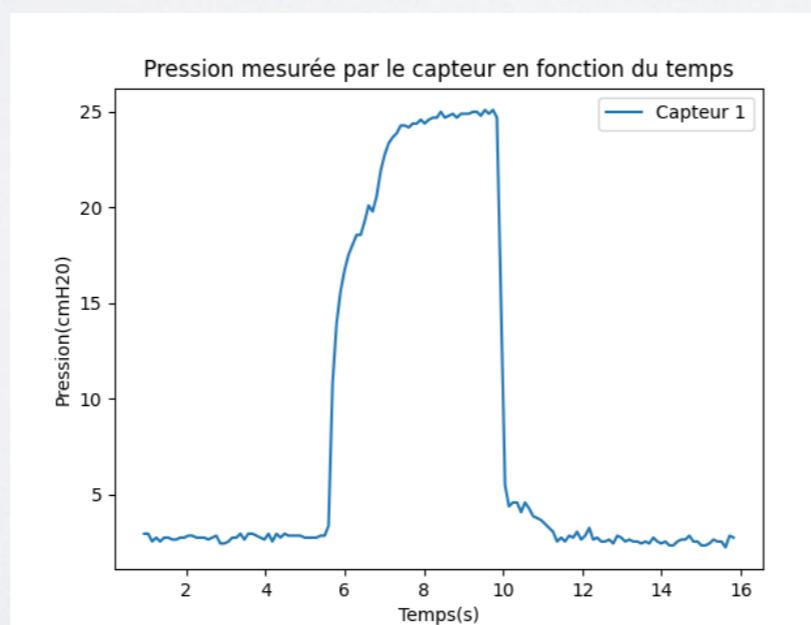
Capteurs de pression différentiels



Carte Arduino



Programme python.

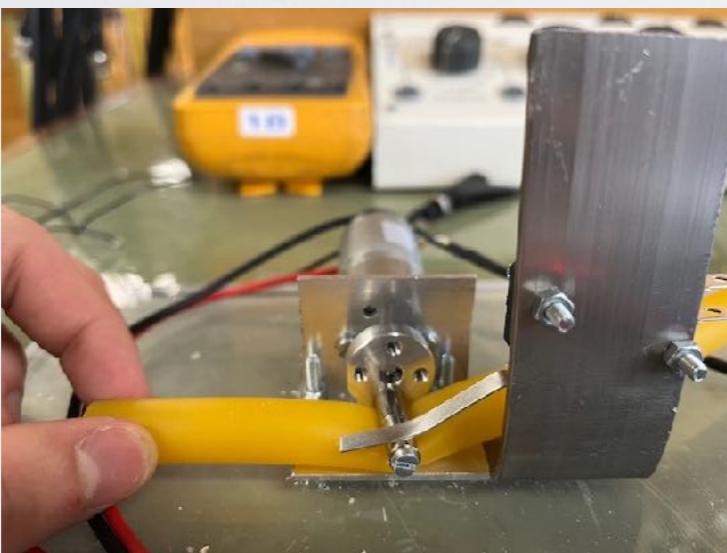


Exemple de courbe renvoyée
après une expérience.

COMMENT CONSTRUIRE UN RESPIRATEUR ARTIFICIEL?

C)Contrôler la pression

1ère version



2ème version

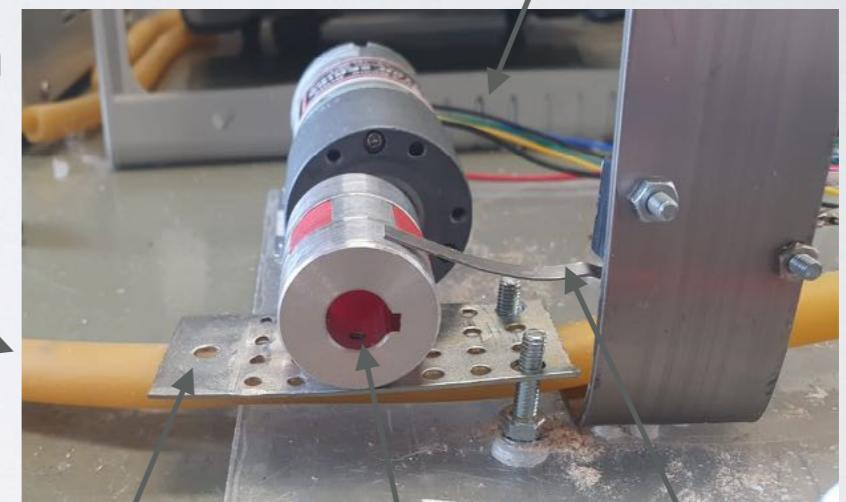
Arrivée d'air

Plaque d'appui

• Le deuxième moteur a un meilleur rapport de réduction et un couple plus élevé.

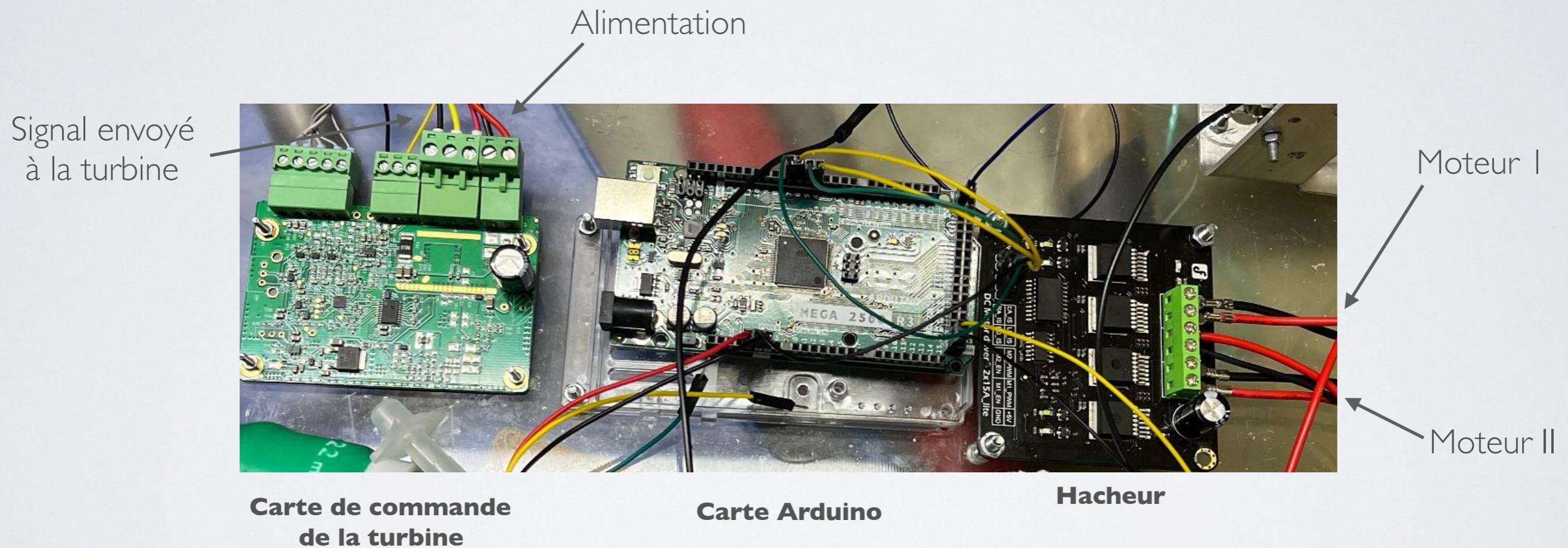
- Moteur trop rapide
- Couple trop faible
- Frottements avec le tuyau

• Avec deux de ces dispositifs on peut régler la pression en écrasant le tuyau.

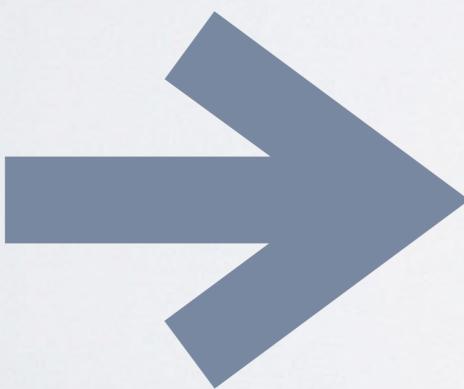
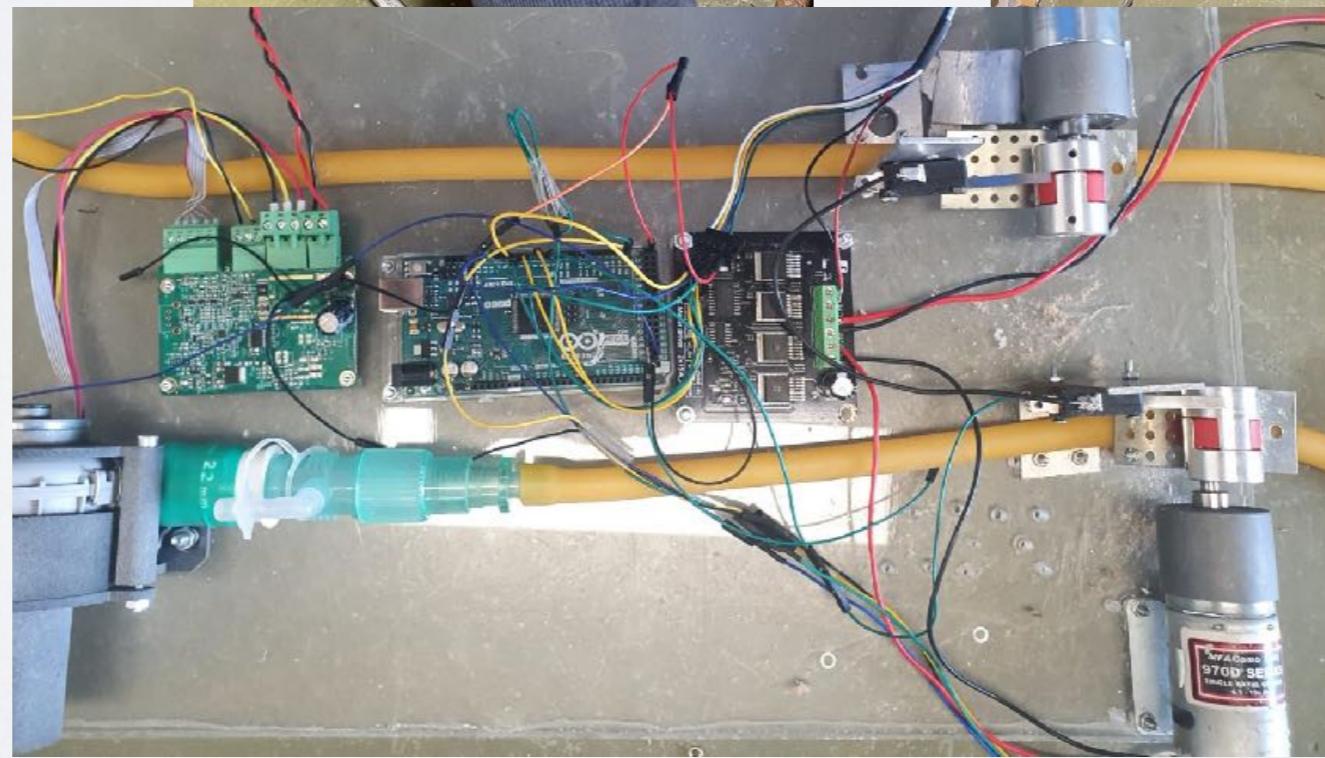
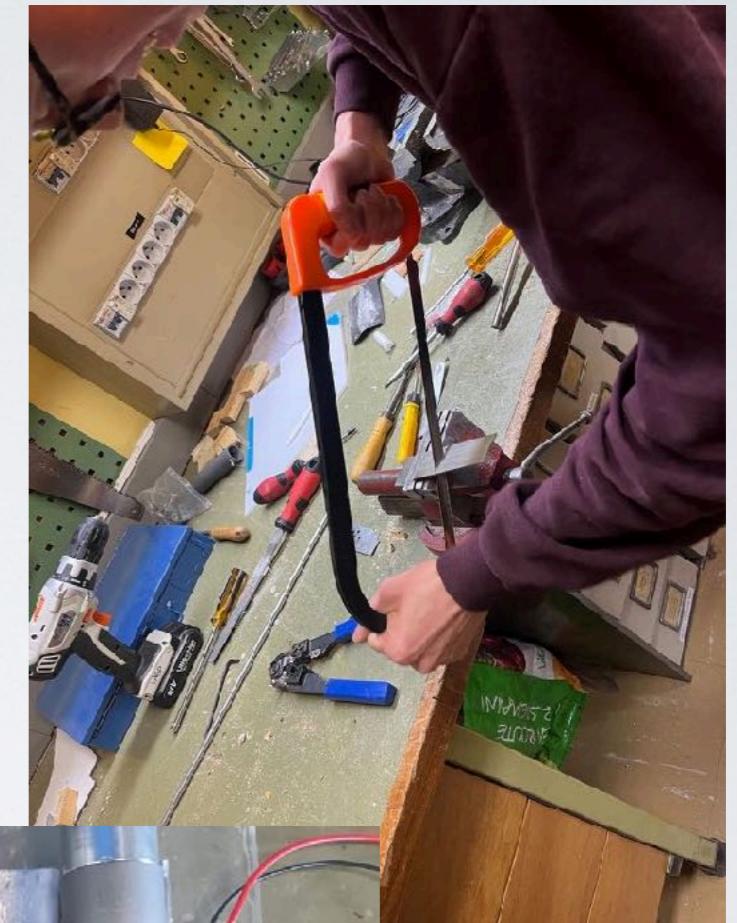
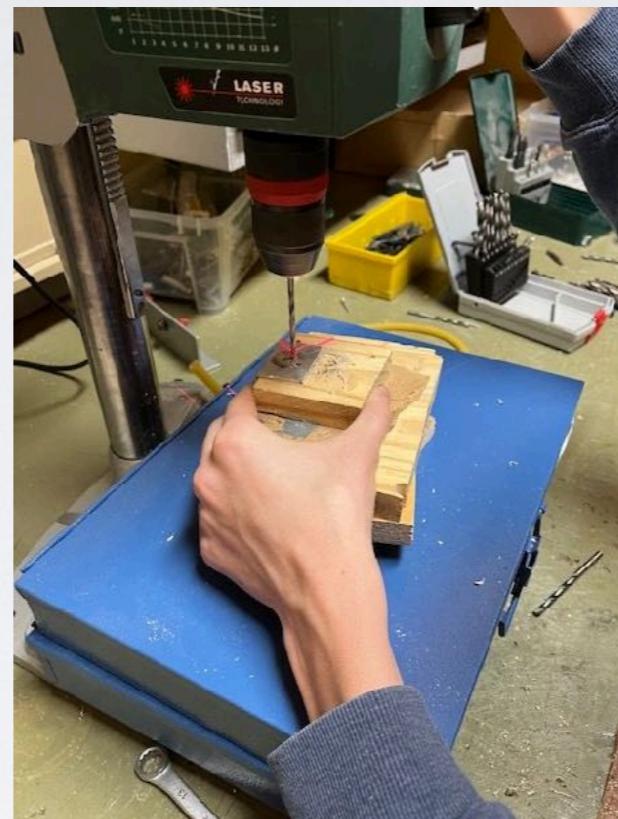
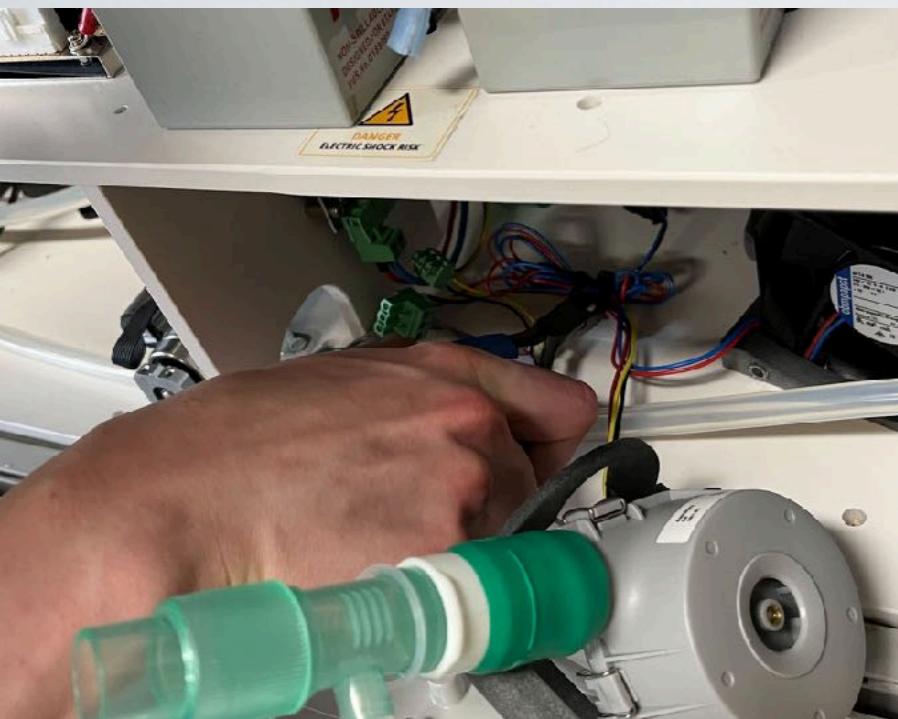


COMMENT CONSTRUIRE UN RESPIRATEUR ARTIFICIEL?

D)Système de commande

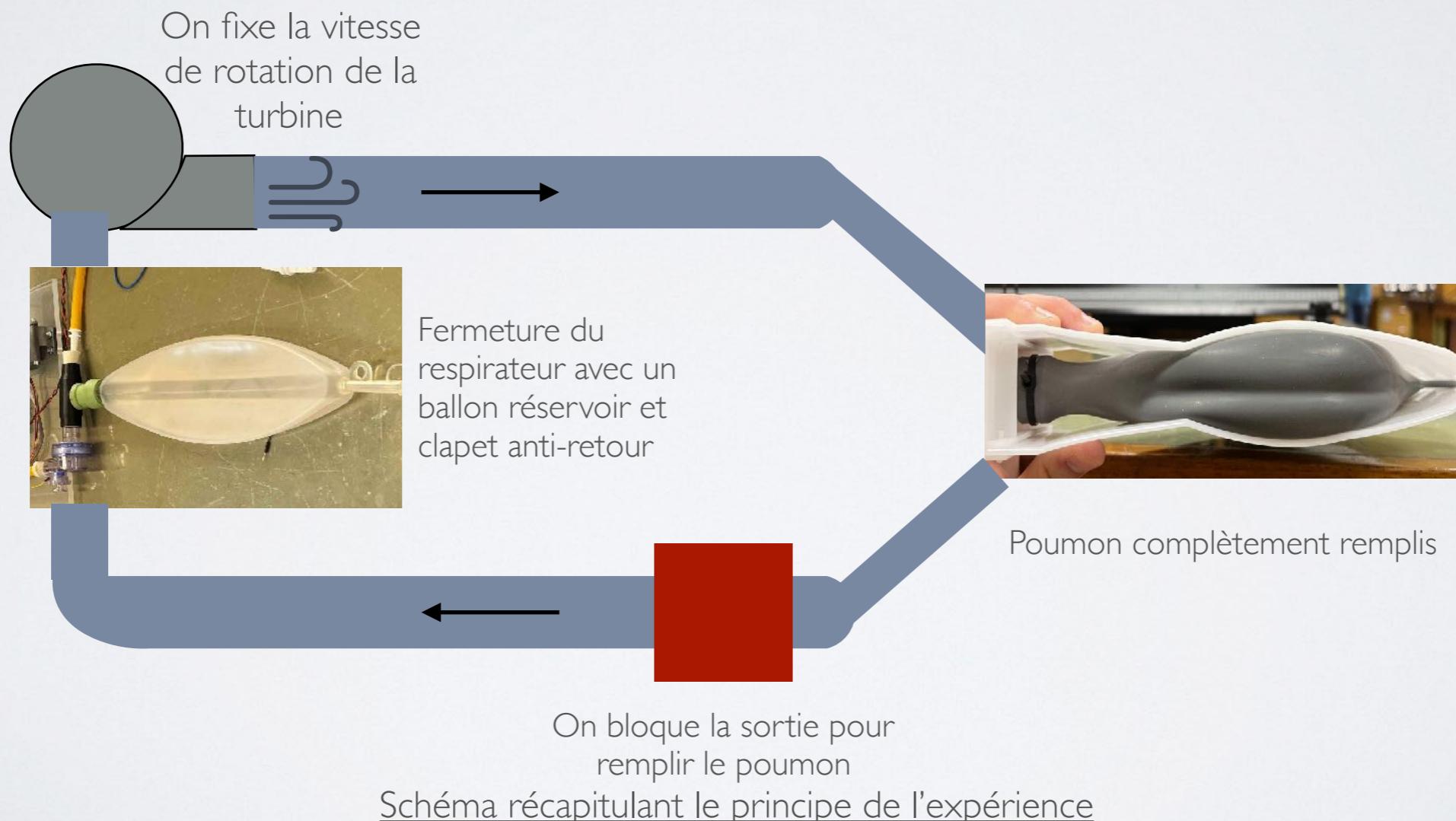


ASSEMBLAGE

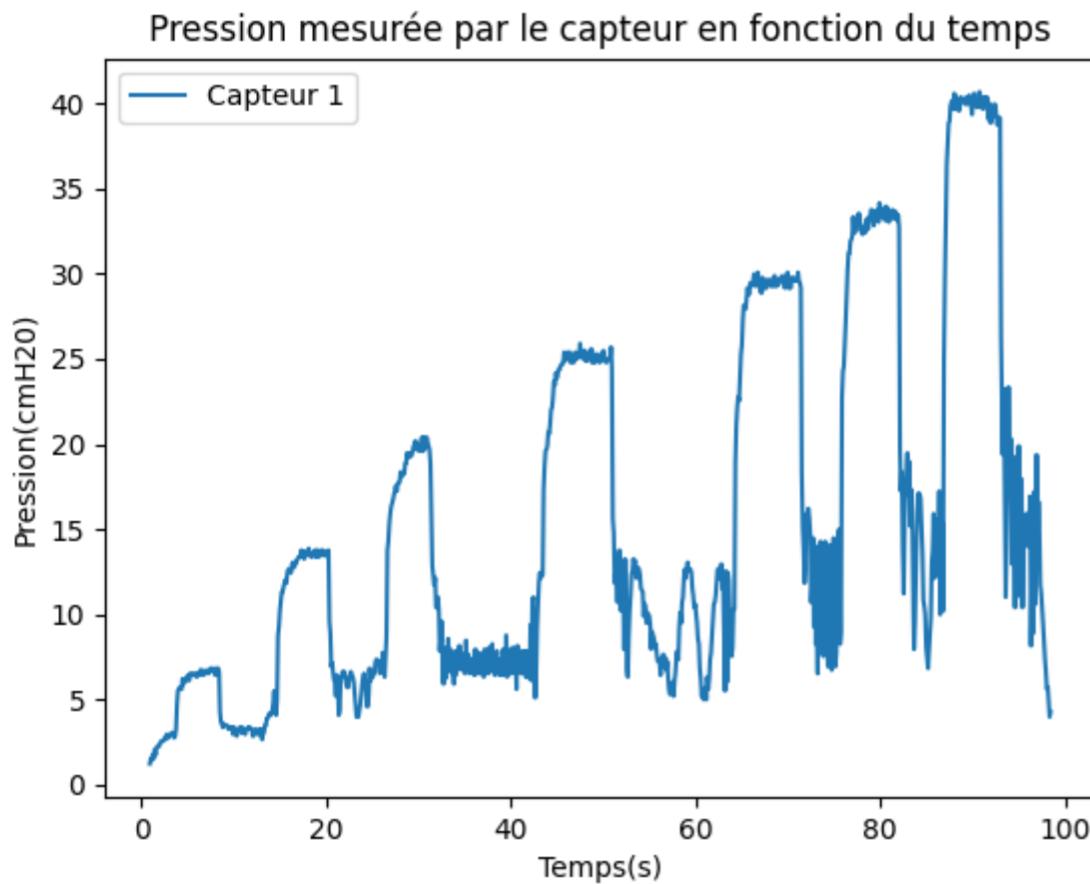


RÉPONDRE AUX DIFFÉRENTES EXIGENCES EN CIRCUIT FERMÉ

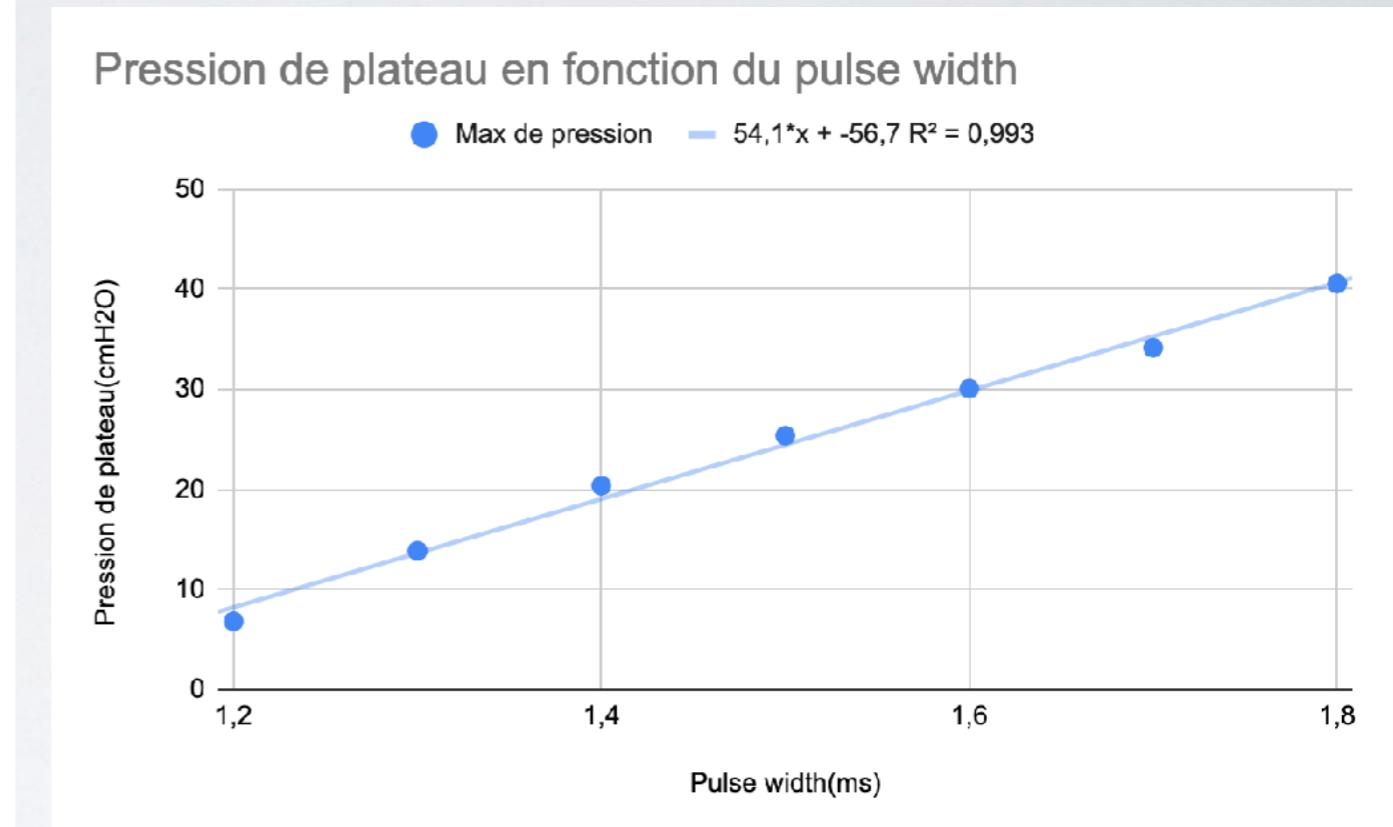
A) Respecter une consigne de pression de plateau



RÉSULTATS EXPÉRIENCE I



Courbe obtenue après avoir remplis à bloc puis vidé le poumon plusieurs fois pour des vitesses de rotation de la turbine variantes.

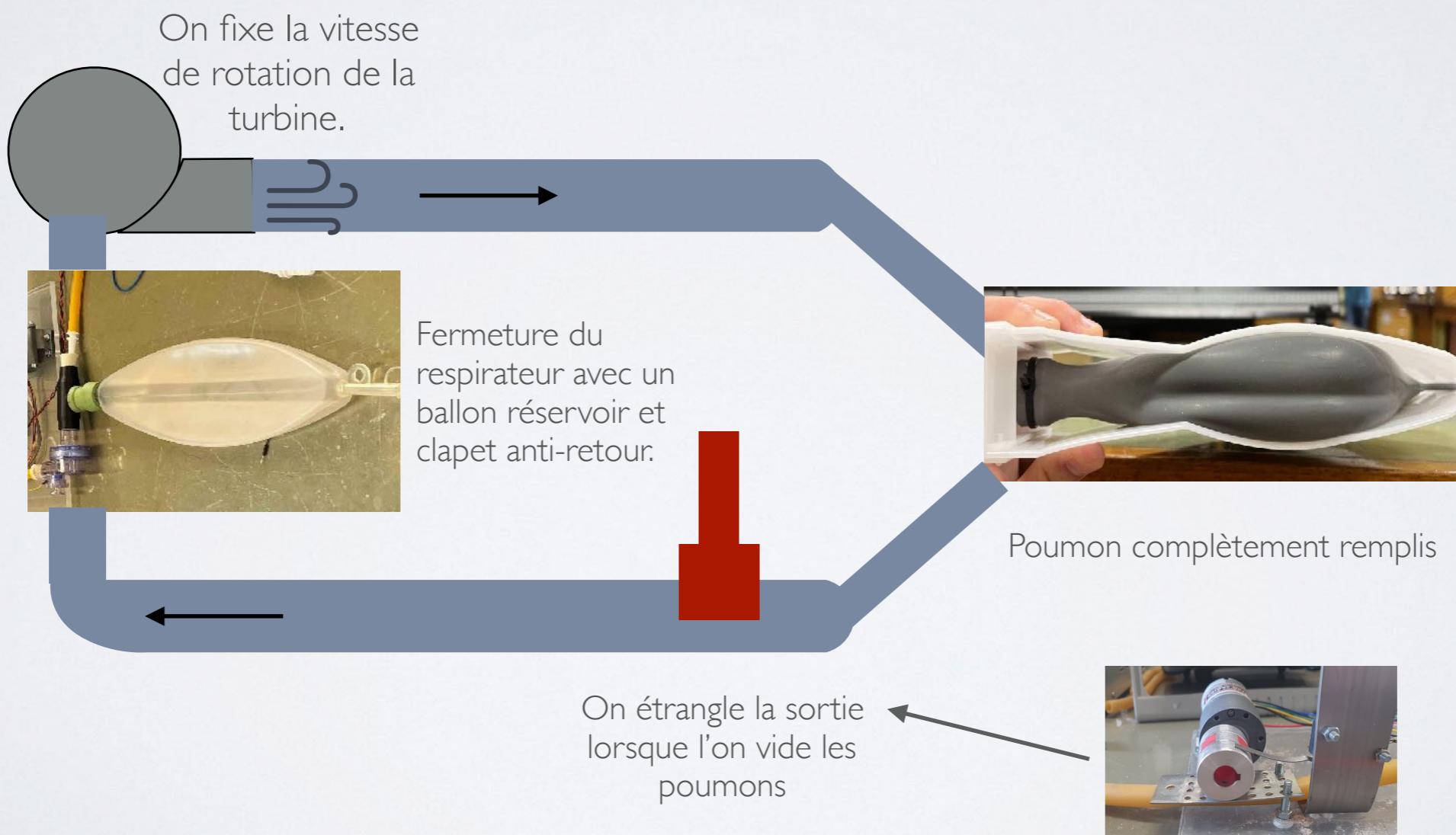


On a ainsi une relation linéaire entre la vitesse de rotation de la turbine et la pression de plateau.

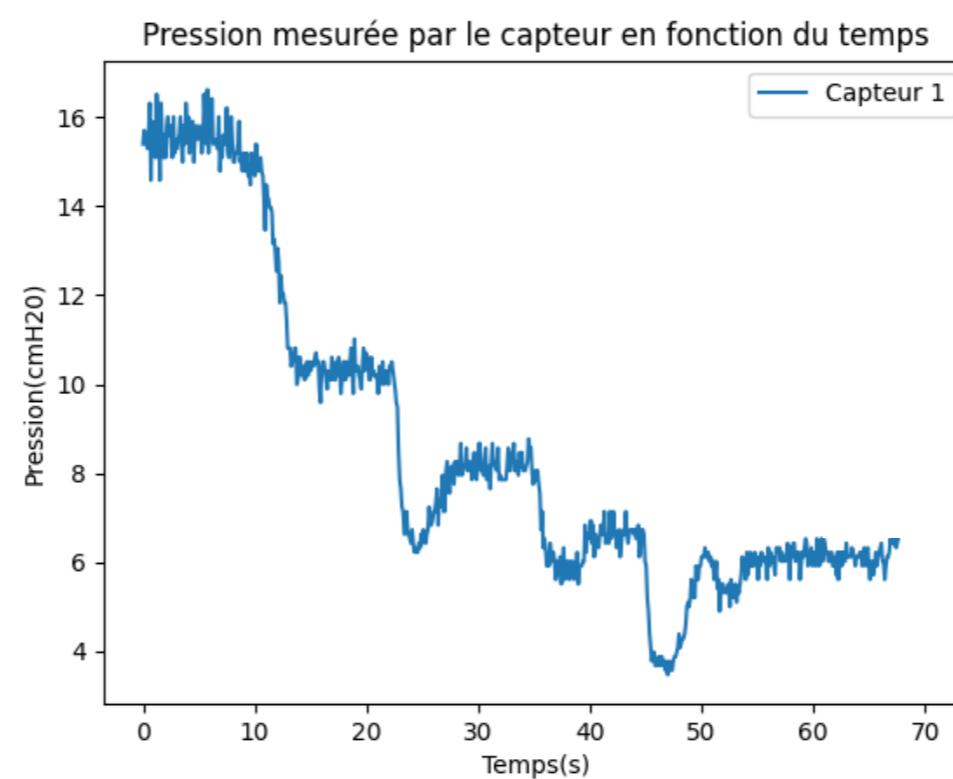
On est donc capable de régler la pression de plateau dans les poumons avec une bonne précision.

RÉPONDRE AUX DIFFÉRENTES EXIGENCES EN CIRCUIT FERMÉ

B) Régler la PEP en étranglant la sortie



RÉSULTATS EXPÉRIENCE 2

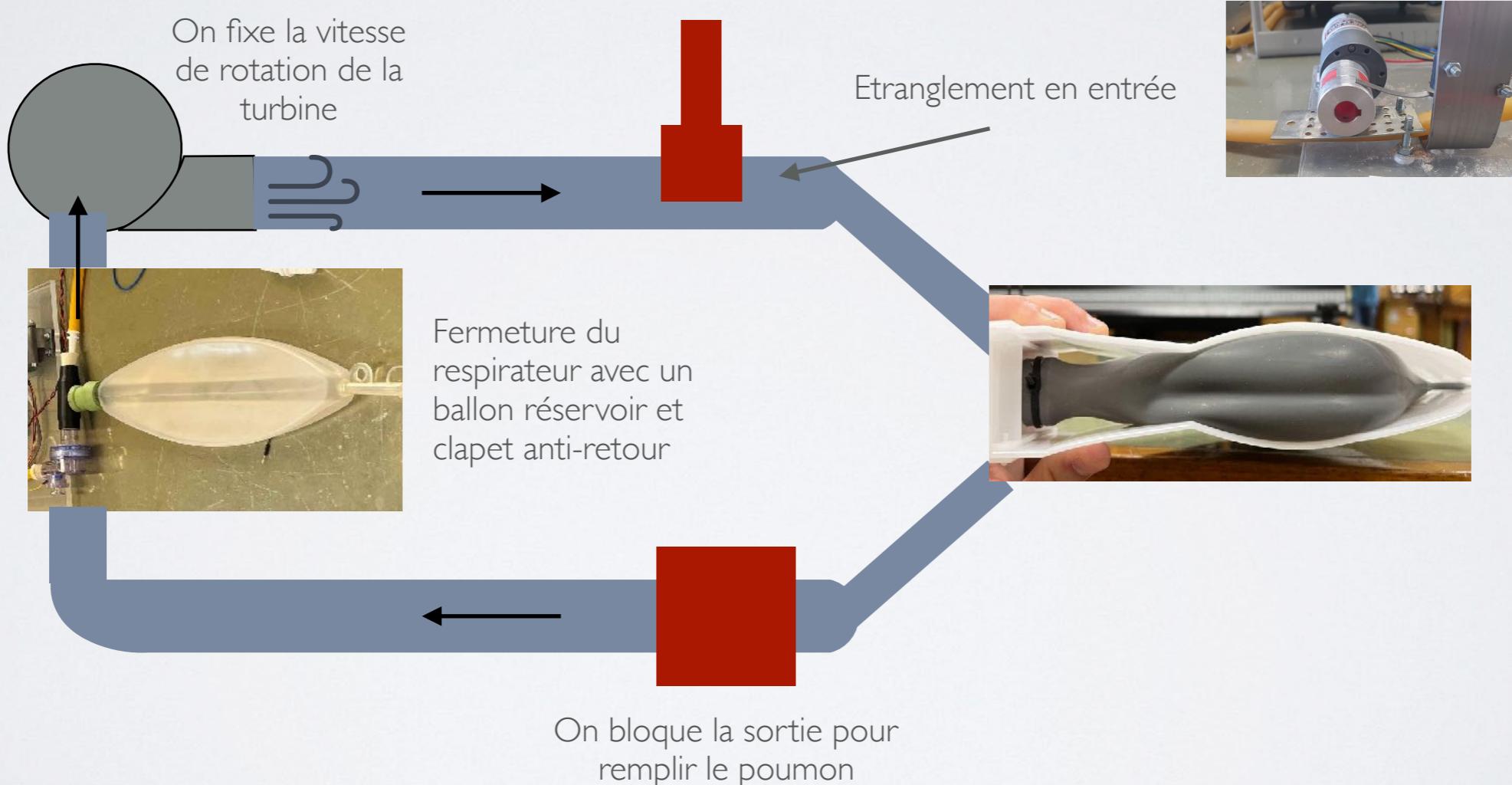


Courbe montrant les variations de la PEP en faisant varier l'étranglement en sortie et en vidant le poumon petit à petit

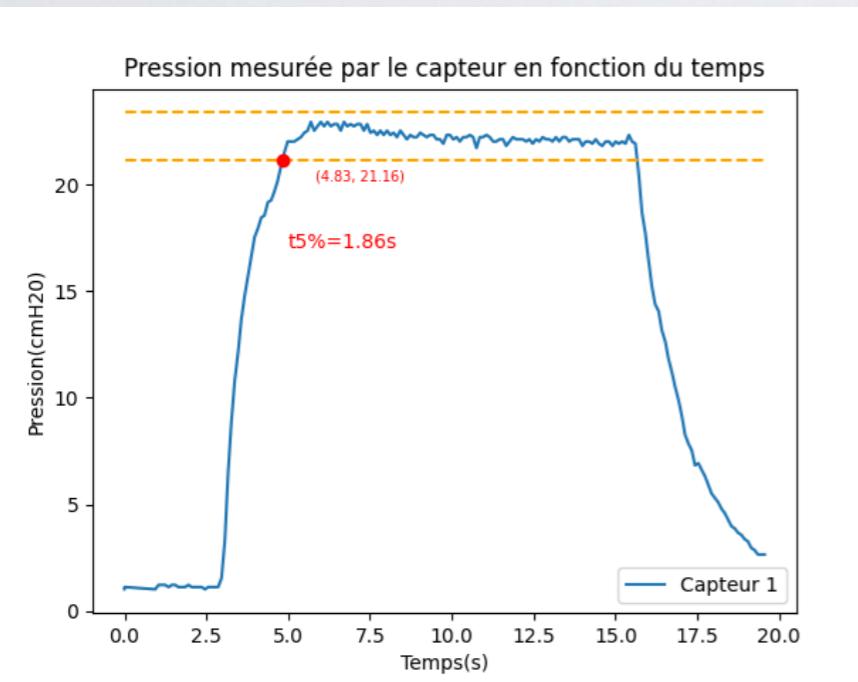
On est donc capable de faire varier la PEP et de s'assurer qu'elle est toujours >0.

RÉPONDRE AUX DIFFÉRENTES EXIGENCES EN CIRCUIT FERMÉ

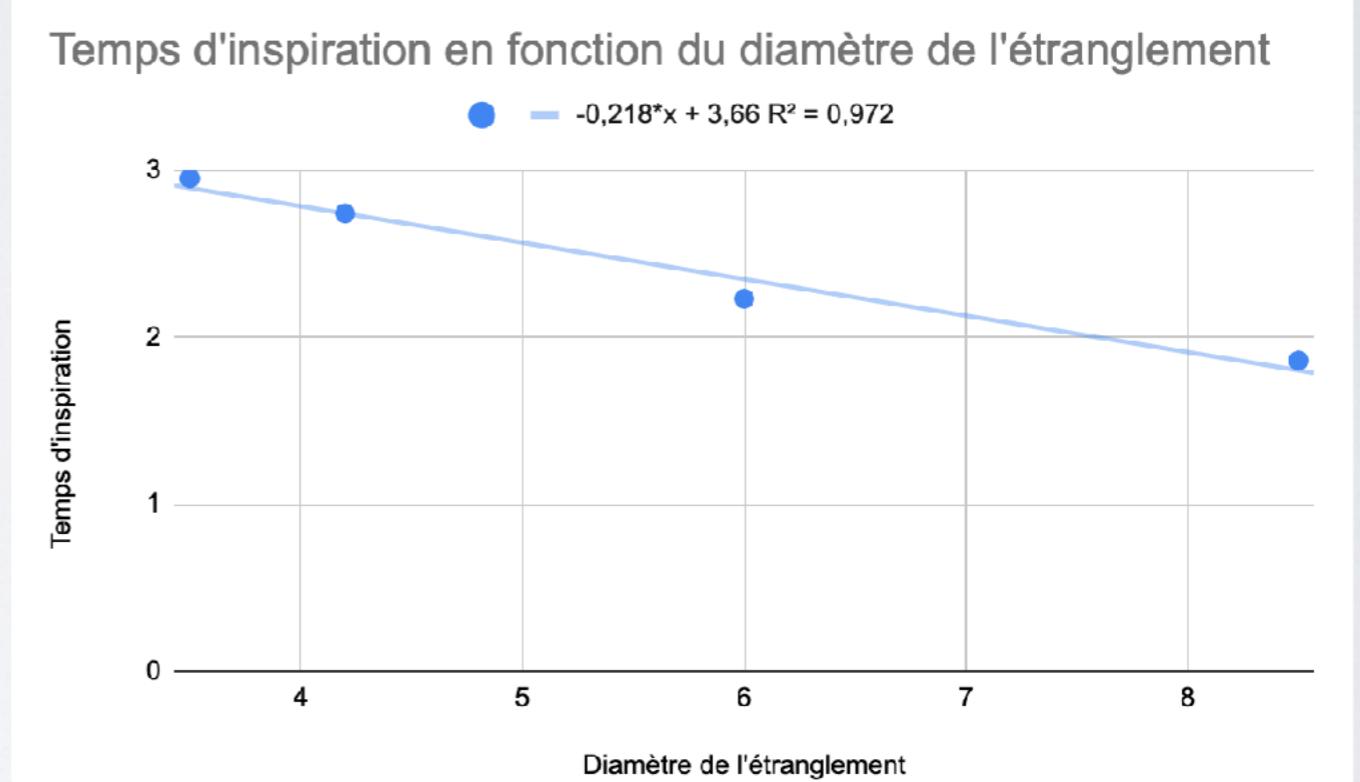
C) Régler le temps d'inspiration en étranglant l'entrée



RÉSULTATS EXPÉRIENCE 3



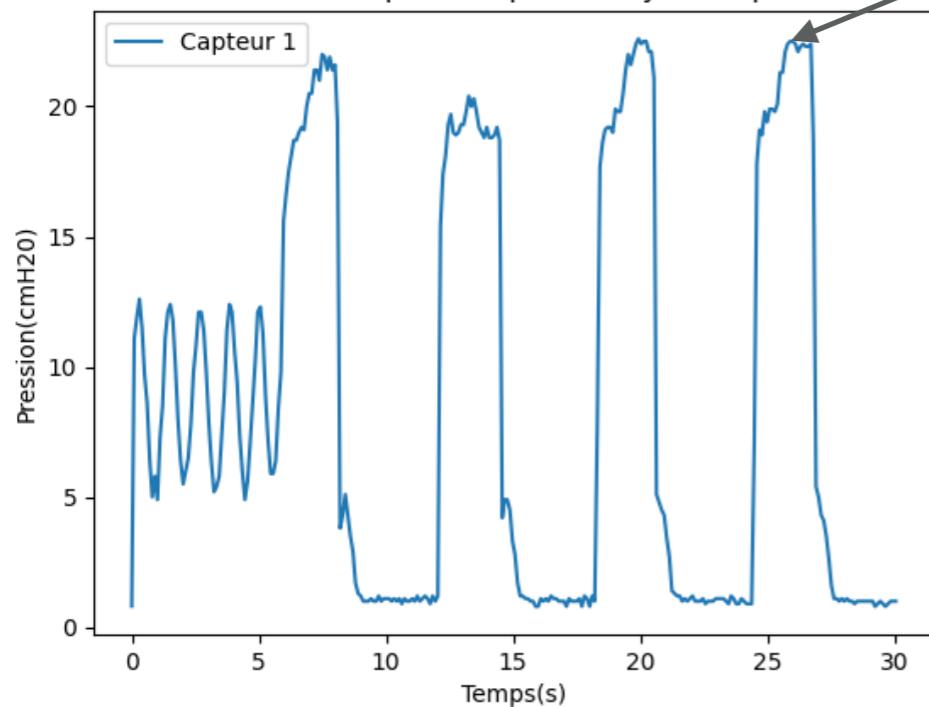
Exemple de courbe obtenue pour un étranglement de 8.4 mm en entrée



On est donc capable de faire varier le temps de montée de la pression dans les poumons.

FONCTIONNEMENT EN CIRCUIT FERMÉ POUR DEUX CYCLES RESPIRATOIRES

Pression mesurée dans le poumon pour un cycle respiratoire "normal".

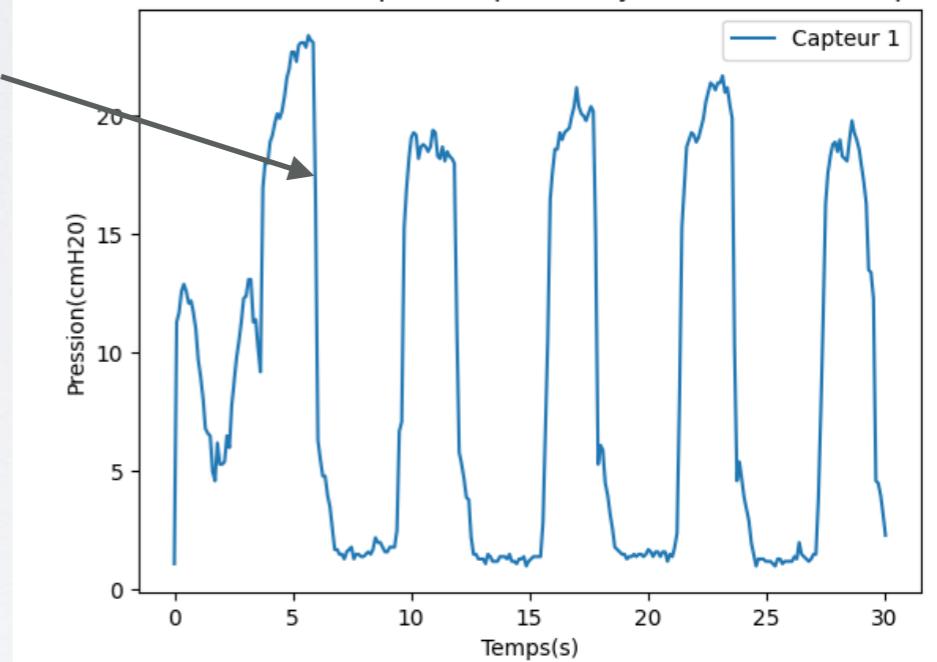


- Inspiration 1.6s
- Expiration 3.2s

Pmax<30cmH20

Tdescente<3s

Pression mesurée dans le poumon pour un cycle de "détresse respiratoire".



- Inspiration 1.2s
- Expiration 2.4s

Réglages:

- Pplateau 20cmH20
- PEP >0 cmH20

CONCLUSION

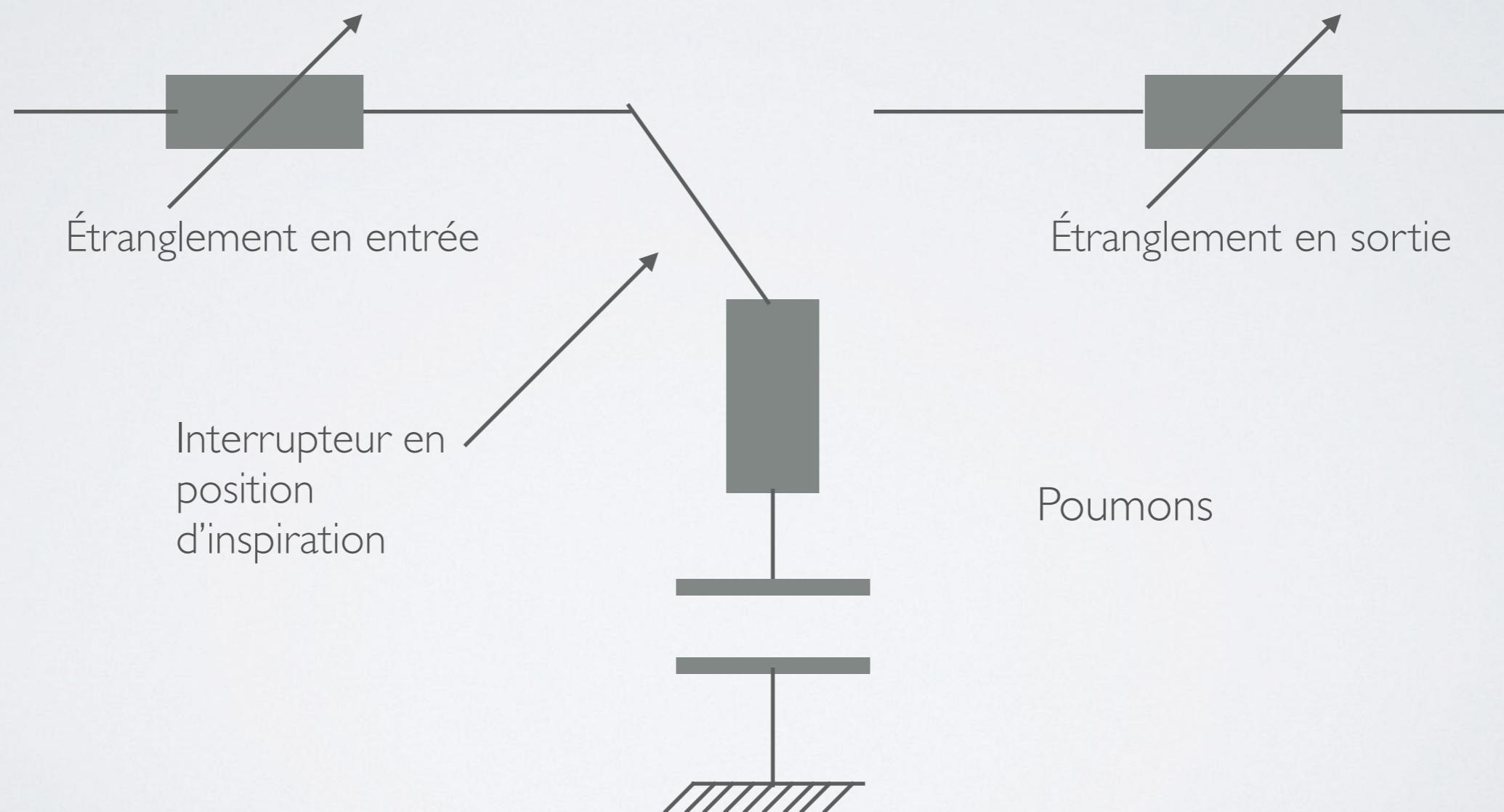
Validation du cahier des charges

Critères	Exigences	Validation
Temps de montée	< 1s	OUI
Temps de descente a la PEP	< 3s	OUI
PEP	Réglable et toujours >0cmH2O	OUI jamais nulle
P_plateau	Réglable entre 20 et 25cmH2O	Manque de précision
Fonctionnement pour différents cycles	Cycle normal et détresse respiratoire	OUI
Pmax	<30cmH2O	OUI

- On règle le problème du temps de descente a la PEP.
- Un asservissement en pression permettrait d'améliorer la precision pour arriver à Pplateau.
- On a du mal a être précis sur l'étranglement que l'on impose.

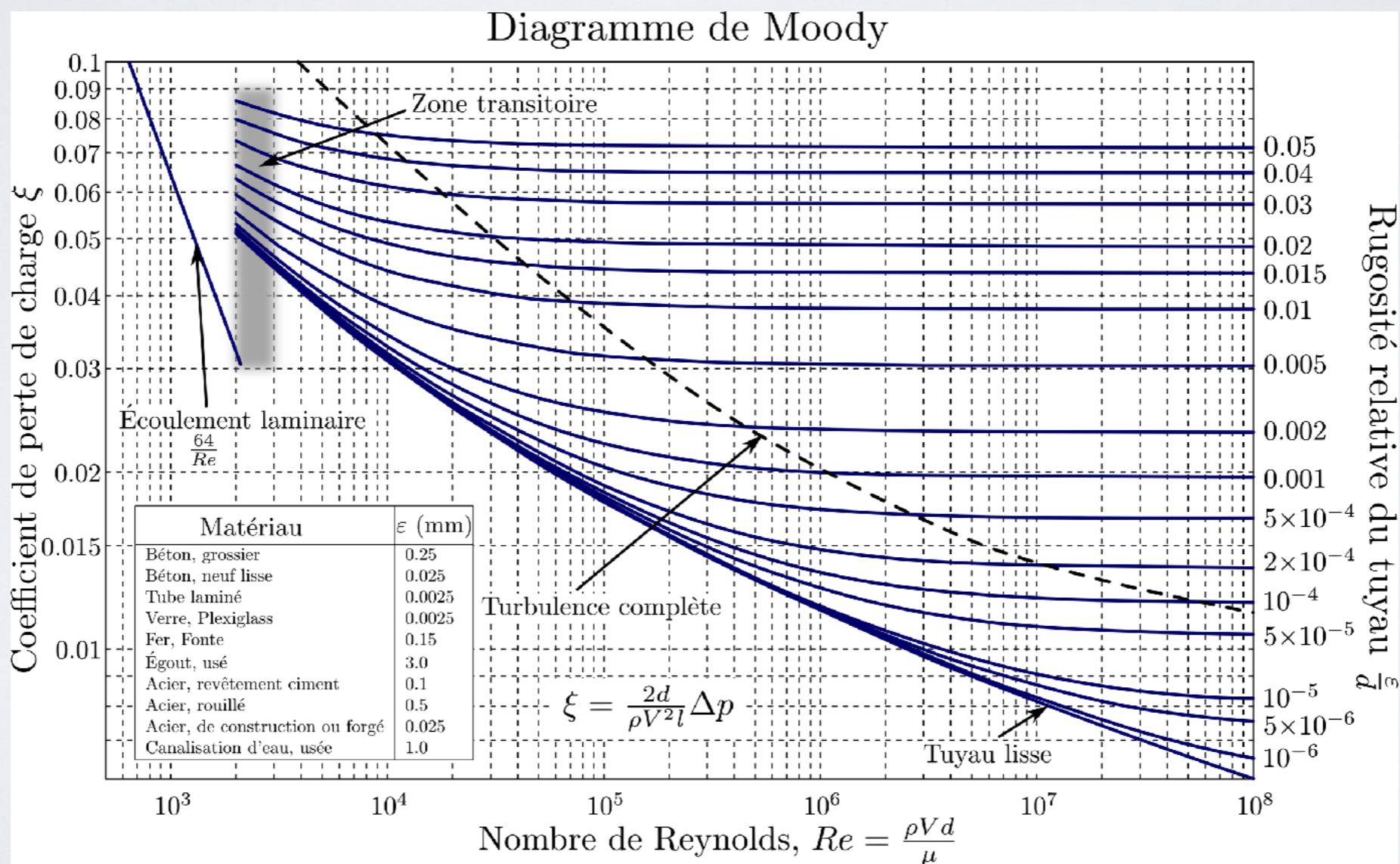
ANNEXE

A)Modèle théorique



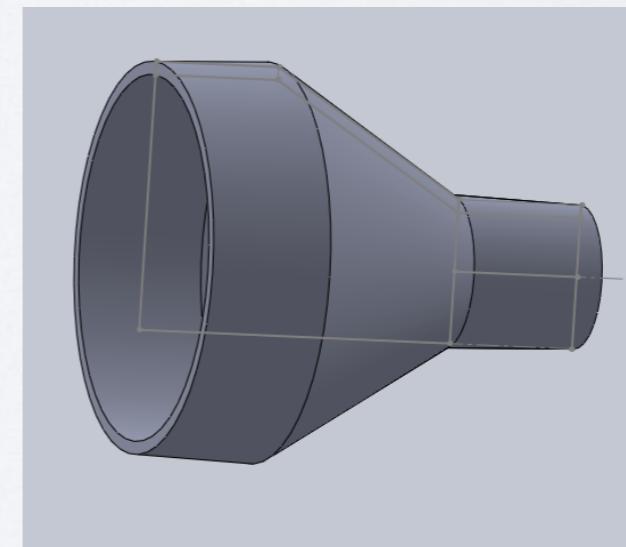
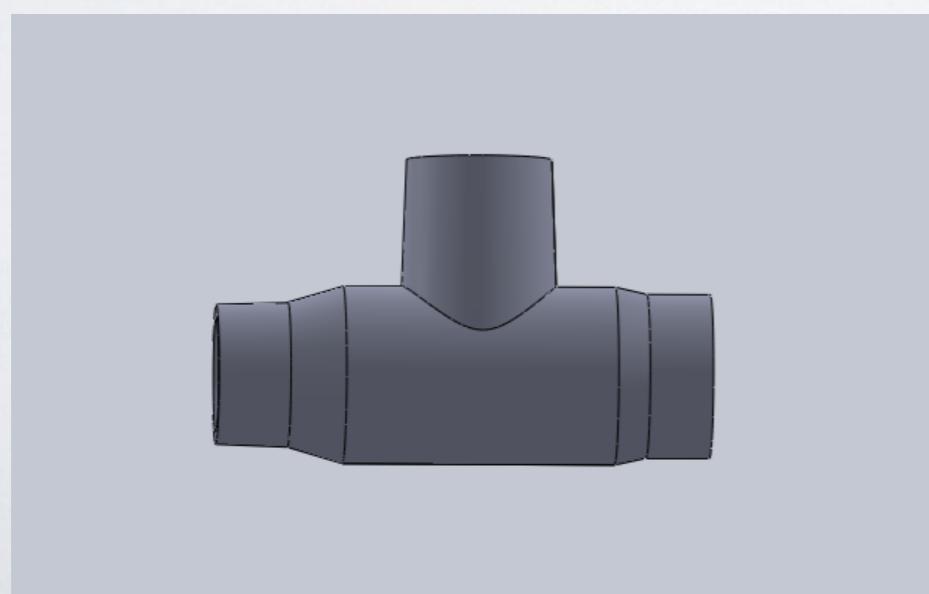
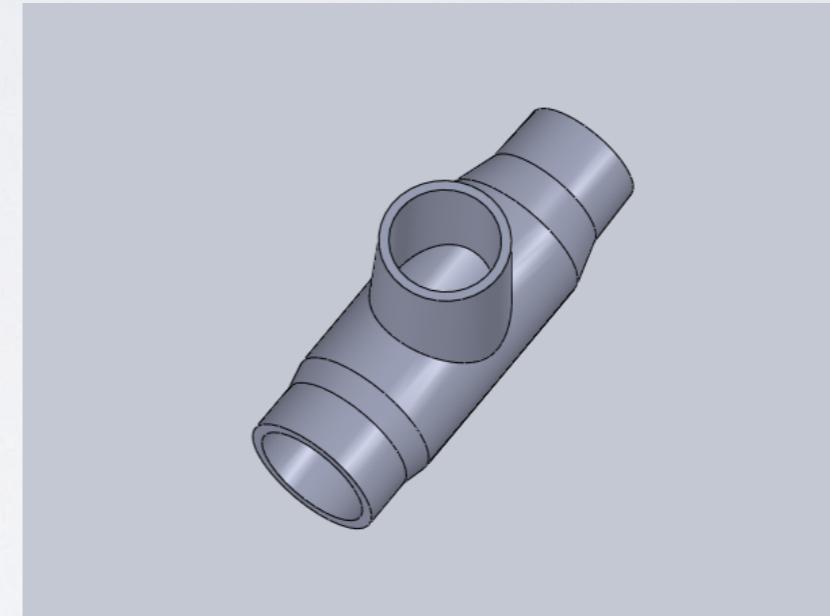
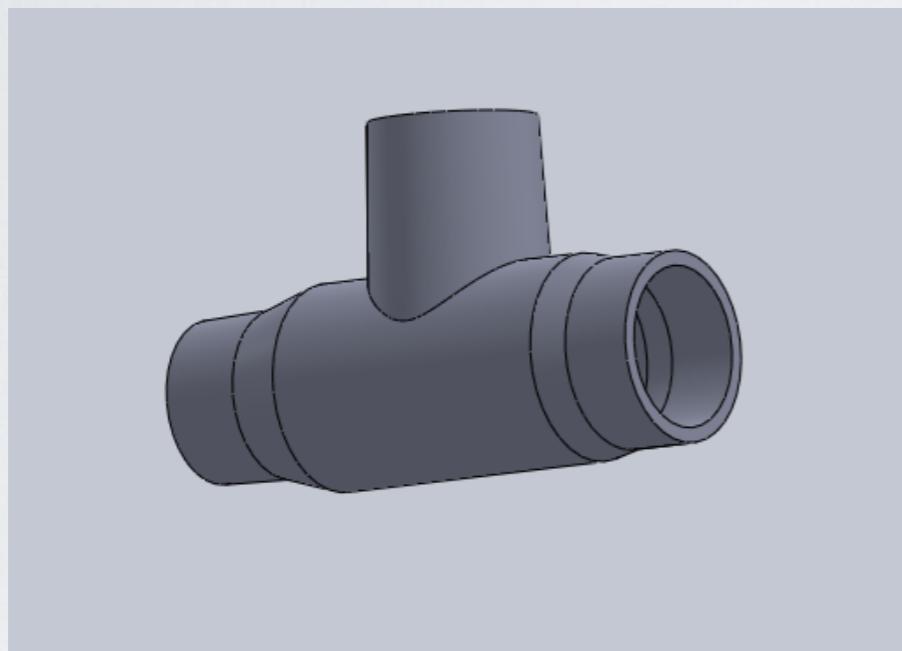
ANNEXE

A) Modèle théorique



ANNEXE

B)Solidworks pour la fermeture du circuit



ANNEXE

C)Python

```
import serial
import time
import csv
import matplotlib.pyplot as plt

#1 RELEVER VALEURS DE PRESSION ET SAUVEGARDER DANS UN FICHIER

nom_fichier = str(time.asctime(time.localtime()))
ser = serial.Serial('/dev/cu.usbmodem143101')
ser.flushInput()
x= []
y1,y2,y3=[],[],[]
debut = time.time()
while True:
    try:
        with open("/Users/louisvauterin/PycharmProjects/TIPE/Valeurs2capteurs/"+nom_fichier,'a') as f:
            sortie_arduino = str(ser.readline())
            pression1 = float(sortie_arduino.split(' ')[1])*1000/98.0665
            pression2 = float(sortie_arduino.split(' ')[3][:4])*1000/98.0665
            print(pression1, pression2)
            y1.append(pression1)
            y2.append(pression2)
            x.append(time.time()-debut)
            writer = csv.writer(f, delimiter=",")
            writer.writerow([time.time(), pression1, pression2])
    except:
        print('FIN')
        break

#2 FAIRE GRAPHIQUE ET SAUVEGARDER
#CAPTEUR 1 = sortie A0 de l'arduino
#CAPTEUR 2 = sortie A1 de l'arduino

plt.plot(x,y2,label='Capteur 1')
plt.plot(x,y2,label='Capteur 1')
plt.title("Pression mesurée par le capteur en fonction du temps")
plt.ylabel("Pression(cmH20)")
plt.xlabel("Temps(s)")
plt.legend()
plt.savefig("/Users/louisvauterin/PycharmProjects/TIPE/Graphiques2capteurs/"+nom_fichier)
plt.show()
```

ANNEXE

D)Arduino

```
// CAPTEUR PRESSION
const float ADC_mV = 4.8828125;      // conversion Arduino ADC à mV
const float SensorOffset = 200.0;       // en mV
const float sensitivity = 4.413;        // en mV/mmH20 (valeurs de la fiche technique)
const float mmh20_kpa = 0.00981;

// HACHEUR

int M1pwm = 4;          //M1 pwm Vitesse Controle
int M1dir = 5;           //M1 Direction Controle
int M2pwm = 6;           //M2 pwm Vitesse Controle
int M2dir = 7;           //M2 direction Controle

//Valeurs codeur
int codeurM1A = 20;
int codeurM1B = 21;
int codeurM2A = 19;
int codeurM2B = 18;

volatile int posM1 = 0;
volatile int posM2 = 0;
//moteur 1 tour =341.2 impulsions par tour

//Valeurs interrupteur Top zero
int topzeroM1 = 2;
int topzeroM2 = 3;

int flag_posM1=0;
int flag_posM2=0;

//Valeurs caracteristiques cycles
unsigned long t1 =1200UL;
unsigned long t2 = 2400UL;
const int Plateau = 20;
const int Crete = 30;
const int PEP = 5;
```

```
//Fonctions

//Moteur 1 gestion du codeur 1
void incrementation1()
{
    if (digitalRead(codeurM1B)==HIGH)
    {
        posM1++;
    }
    else
    {
        posM1--;
    }
}

//Moteur 2 codeur
void incrementation2()
{
    if (digitalRead(codeurM2B)==HIGH)
    {
        posM2++;
    }
    else
    {
        posM2--;
    }
}

void isr_topzeroM1() // si je suis ici c'est top zero M1
{
    flag_posM1=HIGH;
}

void isr_topzeroM2() // si je suis ici c'est top zero M2
{
    flag_posM2=HIGH;
}
```

ANNEXE

```

void prise_pression()
{
    float valeurcapteur1 = ((analogRead(A13) * ADC_mV - SensorOffset) / sensitivity * mmh20_kpa);
    float valeurcapteur2 = ((analogRead(A12) * ADC_mV - SensorOffset) / sensitivity * mmh20_kpa);
    Serial.print("Pression1: ");
    Serial.print(valeurcapteur1);
    Serial.print(" Pression2: ");
    Serial.println(valeurcapteur2);
}

void stop_moteur(int moteur_v,int moteur_d)           //Stop Moteur
{
    digitalWrite(moteur_v,0); //vitesse du moteur a 0
    digitalWrite(moteur_d,LOW); //position
}

void avancer_moteur(int moteur_v,int moteur_d,int vitesse) //Avancer le moteur
{
    analogWrite (moteur_v,vitesse);
    digitalWrite(moteur_d,HIGH);
}

void aller_top_zeroM1(void)
{
    flag_posM1=LOW;
    while(flag_posM1==LOW)
    {
        avancer_moteur(M1pwm,M1dir,150);
        prise_pression();
    }

    stop_moteur(M1pwm, M1dir);
    posM1=0;
}

```

```

void setup(void)
{
    //codeur
    pinMode(codeurM1A,INPUT);
    pinMode(codeurM1B,INPUT);
    pinMode(codeurM2A,INPUT);
    pinMode(codeurM2B,INPUT);

    //interrupteur top zero
    pinMode(topzeroM1,INPUT_PULLUP);
    pinMode(topzeroM2,INPUT_PULLUP);

    pinMode(M1pwm, OUTPUT);
    pinMode(M1dir, OUTPUT);
    pinMode(M2pwm, OUTPUT);
    pinMode(M2dir, OUTPUT);

    Serial.begin(9600);

    attachInterrupt(digitalPinToInterrupt(codeurM1A),incrementation1,RISING);
    attachInterrupt(digitalPinToInterrupt(codeurM2A),incrementation2,RISING);
    attachInterrupt(digitalPinToInterrupt(topzeroM1),isr_topzeroM1,RISING);
    attachInterrupt(digitalPinToInterrupt(topzeroM2),isr_topzeroM2,RISING);
}

```

D)Arduino

ANNEXE

```
void loop(void)
{
// on utilise l'inertie du moteur pour ce qui le place juste avant le tube apres un top zero
//Donc avec une petite impulsion on écrase complètement le tube(valeur de 6 trouvée avec des essais...)
//Au final on n'utilise pas les codeurs mais on en aurait besoin pour une plus grande précision

//Phase 1 remplissage du poumon = inspiration
for (int i=0;i<=6;i++)
{
    avancer_moteur(M1pwm,M1dir,150);
}
stop_moteur(M1pwm, M1dir);

unsigned long debut = millis();

//Continuer de faire des prises de pression pendant le temps "d'attente"
while (millis()-debut < t1)
{
    prise_pression();
}

//Phase 2 sortie d'air = expiration
aller_top_zeroM1();
for (int i=0;i<=6;i++)
{
    avancer_moteur(M2pwm,M2dir,150);
}

while (millis()-debut < t2)
{
    prise_pression();
}

unsigned long debut = millis();

//début de la phase 1
aller_top_zeroM2();
}
```