



# Estudi i millora d'un respirador pulmonar artificial

Document:

Annexos

Autor/Autora:

Amorós Pino, Manel

Director/Directora - Codirector/Codirectora:

Pujol Vazquez, Gisela / Acho Zuppa, Leonardo

Titulació:

Grau en enginyeria electrònica, industrial i  
automàtica

Convocatòria:

Primavera, 2022.

TREBALL DE FI D'ESTUDIS



## Índex

<b>Annex 1: Codi en Python de funcionament del respirador .....</b>	<b>1</b>
<b>Annex 2: Codi en C++ del sensor de pressió.....</b>	<b>3</b>
<b>Annex 3: Transformar un codi Python en executable en Linux:.....</b>	<b>4</b>
<b>Annex 4: Aplicació de control del respirador pulmonar .....</b>	<b>5</b>

## Annex 1: Codi en Python de funcionament del respirador

```
#!/usr/bin/env python3
import time
import RPi.GPIO as GPIO
from os import _exit
import serial
import numpy as np
import matplotlib.pyplot as plt
GPIO.setmode(GPIO.BOARD)
#I:E
GPIO.setup(7,GPIO.IN)
#FreR
GPIO.setup(3,GPIO.IN)
GPIO.setup(5,GPIO.IN)
GPIO.setup(11,GPIO.IN)
GPIO.setup(13,GPIO.IN)
#VT
GPIO.setup(15,GPIO.IN)
GPIO.setup(19,GPIO.IN)
GPIO.setup(21,GPIO.IN)
GPIO.setup(23,GPIO.IN)
k=0
h=0.15
GPIO.setup(32,GPIO.OUT)
freqPWM=50
pwm=GPIO.PWM(32,freqPWM)
pwm.start(4)
ser=serial.Serial("/dev/ttyACM0",9600)
time.sleep(2)
xs=[]
ys=[]
f=open("Datos3.txt","w")
j1=0
x1a=0.5
while True:
x=ser.readline()
x1=x.rstrip()
if x1==b'':
x1=x1a
try:
float(x1)
x1=x1
except:
x1=x1a
if float(x1)>1:
x1=x1a
x1a=x1
x2=float(x1)
#print('Presión= ',x2)
xs.append(k*h)
ys.append(x2)
f.write(str(round(k*h,3))+'\t')
f.write(str(round(x2,3))+'\n')
if(GPIO.input(7)==True):
aux1=3
if(GPIO.input(7)==False):
aux1=2
#Vt
if(GPIO.input(3)==False&GPIO.input(5)==False&GPIO.input(11)==False\
&GPIO.input(13)==False):
```

```

aux2=0.7
if (GPIO.input(3)==True&GPIO.input(5)==False&GPIO.input(11)==False\
&GPIO.input(13)==False):
aux2=0.8
if (GPIO.input(3)==True&GPIO.input(5)==True&GPIO.input(11)==False\
&GPIO.input(13)==False):
aux2=0.9
if (GPIO.input(3)==True&GPIO.input(5)==True&GPIO.input(11)==True\
&GPIO.input(13)==False):
aux2=1
#FR
if (GPIO.input(15)==False&GPIO.input(19)==False&GPIO.input(21)==False\
&GPIO.input(23)==False):
fr=5
if (GPIO.input(15)==True&GPIO.input(19)==False&GPIO.input(21)==False\
&GPIO.input(23)==False):
fr=10
if (GPIO.input(15)==True&GPIO.input(19)==True&GPIO.input(21)==False\
&GPIO.input(23)==False):
fr=15
if (GPIO.input(15)==True&GPIO.input(19)==True&GPIO.input(21)==True\
&GPIO.input(23)==False):
fr=20
if (GPIO.input(15)==True&GPIO.input(19)==True\
&GPIO.input(21)==True&GPIO.input(23)==True):
fr=5
#Servo-Motor
j1=j1+h
if j1<=(60/(aux1*fr)):
pwm.ChangeDutyCycle(8*aux2)
if j1>(60/(aux1*fr)) and j1<=(60/fr):
pwm.ChangeDutyCycle(3.0)
if j1 >(60/fr):
j1=0
time.sleep(h)
k=k+1
if k>10000*h:
pwm.ChangeDutyCycle(3.0)
pwm.stop()
GPIO.cleanup()
ser.close()
f.close()
break
pwm.ChangeDutyCycle(3.0)
plt.plot(xs,ys)
plt.show()
f.close()
-exit()

```

## Annex 2: Codi en C++ del sensor de pressió

Hi ha dos codis diferents que s'han utilitzat, un per a cada Arduino.

ARDUINO (ACM0): Port serial que llegeix el codi Python on la pressió s'envia en les unitats de voltatge:

```
void setup() {  
  
    Serial.begin(9600);  
  
}  
  
void loop() {  
  
    float val=analogRead(A1) * (5.0/1023.0);  
    delay(25);  
    Serial.println(val);  
  
}
```

ARDUINO (ACM1): El port serial es llegit pel Node-Red per representar els valors de la pressió en les unitats de cmH<sub>2</sub>O:

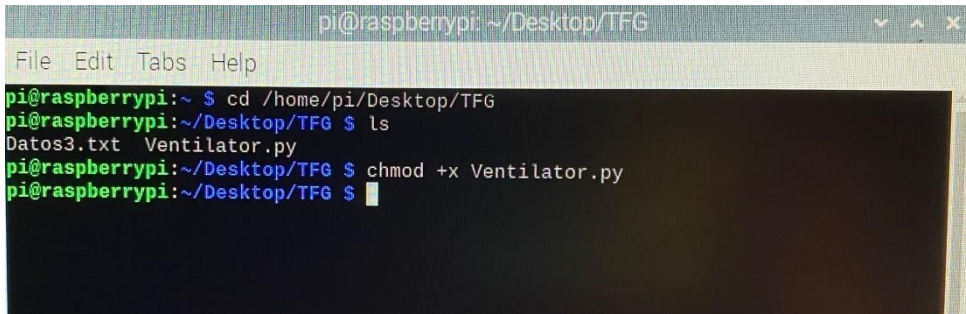
```
void setup() {  
  
    Serial.begin(9600);  
  
}  
  
void loop() {  
  
    float val=analogRead(A1) * (5.0/1023.0);  
    delay(25);  
    float pressio=21.8(val-0.4);  
  
    Serial.println(pressio);  
  
}
```

## Annex 3: Transformar un codi Python en executable en Linux:

Primer de tot s'ha d'afegir una línia en el codi Python per a poder definir la ubicació del programari i fer-lo executable en un sistema operatiu Linux:

```
#!/usr/bin/env python3
```

Una vegada afegida, hem d'accedir a la consola de la nostra Raspberry i accedir a la ubicació on es troba el codi. En el nostre cas la ubicació és: /home/pi/Desktop/PythonPrograms.



```

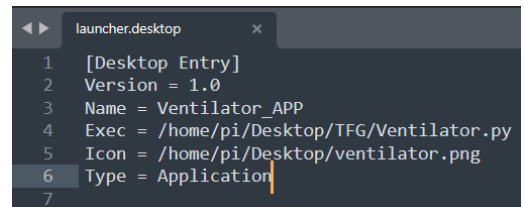
pi@raspberrypi:~/Desktop/TFG
File Edit Tabs Help
pi@raspberrypi:~ $ cd /home/pi/Desktop/TFG
pi@raspberrypi:~/Desktop/TFG $ ls
Datos3.txt Ventilator.py
pi@raspberrypi:~/Desktop/TFG $ chmod +x Ventilator.py
pi@raspberrypi:~/Desktop/TFG $
  
```

Després afegim la comanda “`chmod +x Covid_19.py`”. Ara aquest programa ja es executable, però només a través de la consola. L'objectiu és crear una icona a l'escriptori per a que sigui fàcilment accessible. Per fer-ho, ens hem d'instal·lar un editor de text, en el nostre cas vam instal·lar “Sublime Text”.

Una vegada hi tenim accés, s'han d'escriure les següents línies de codi:

```

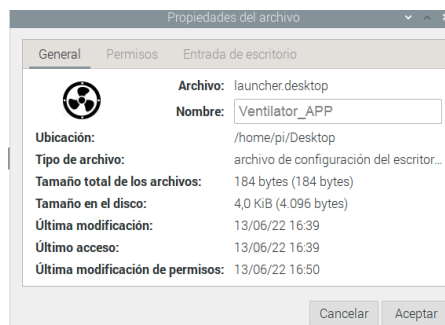
[Desktop Entry]
Version = 1.0
Name = “Nom que es vulgui posar a l'executable”
Exec = “Ubicació del codi que es vol executar”
Icon = “Ubicació de la imatge en .png que es vol posar d'icona”
Type = Application
  
```



```

launcher.desktop
1 [Desktop Entry]
2 Version = 1.0
3 Name = Ventilator_APP
4 Exec = /home/pi/Desktop/TFG/Ventilator.py
5 Icon = /home/pi/Desktop/ventilator.png
6 Type = Application
7
  
```

Finalment, es crea al escriptori una icona executable del codi Python que fa funcionar el respirador artificial. El resultat és el següent:

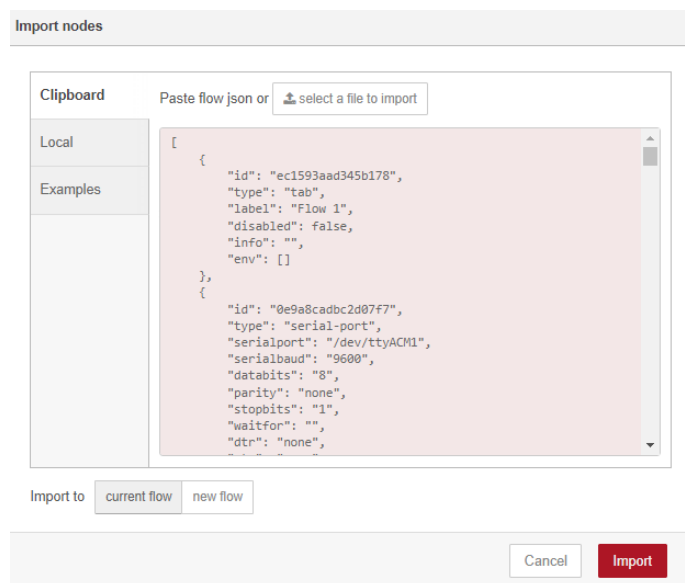


## Annex 4: Aplicació de control del respirador pulmonar

Clicant a la imatge següent us portarà a un fitxer que conté el programa de control del respirador pulmonar en format .json. Has d'accedir-hi amb el correu upc.edu per poder veure l'arxiu.



Per importar l'aplicació al Node-Red, hem d'accedir a la secció "Import" i afegir el fitxer anterior o copiant el contingut d'aquest.



Posteriorment, només farà falta donar al botó "Deploy" a la part superior per fer córrer el codi i accedir a la URL `<ip>:1880/ui`. D'aquesta manera qualsevol persona pot tenir l'aplicació totalment gratuïta.