Institute for advanced architecture of Catalonia

MRAC

Master in Robotics and Advanced Construction

Hardware II
IaaC, 28.01.2022

Faculty / Oscar Gonzalez
Assistants/ D. Koshelyuk

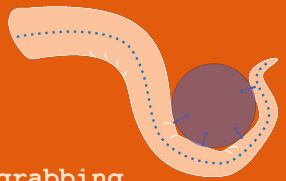
INFLATABLE AGGRESSION

Andrea Nájera
Grace Boyle
Jordi Vilanova
Amy Kim

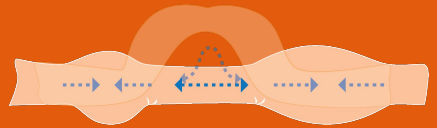
1.1 SOFT ROBOTIC

< Soft robotic is still a more novel research area. It can bring new ranges of movement; it can address problems that come from rigidity of materials in our robotics. >

Crucially, it can also reframe how we experience computation.



grabbing



locomotion

Nasa's Langley Research Center

1.2 STATE OF THE ART

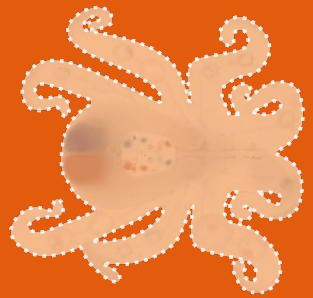
uses hydrogen peroxide as its main power source and also contains a microfluidic logic circuit.

 # NASA:
year: 2019
material: 3D printed silicon

description:
researching + creating soft robot actuators for outer space exploration.

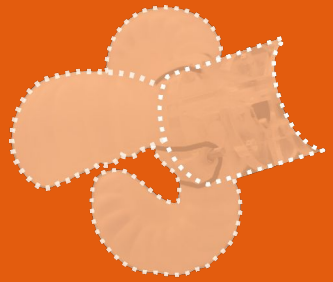
 # OCTOBOT:
year: 2017
material: 3D printed silicon

description:
fully autonomous robot using hydrogen peroxide as main power source + contains microfluidic logic circuit



 # OCTOBOT:
year: 2020
material: silicon

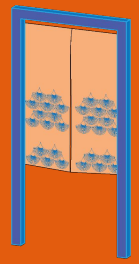
description:
Soft robotic gripper as an robotic arm actuator to pick up and screw in a light bulb.



1.3 CONCEPT

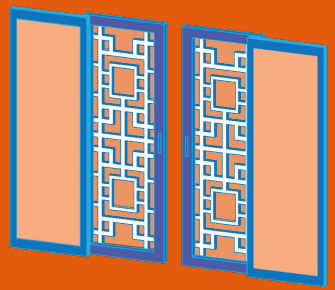
 # Considerations in architecture:
doors

<"material of doors in the other, more traditional, culture reveal a more variety of material.
We revisit the notion of door as barrier/gate, and how it is executed.">



noren door
[Japan]

curtain door
[Namibia]



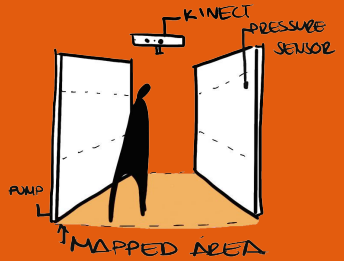
hanok door
[Korea]

Pneuhaus

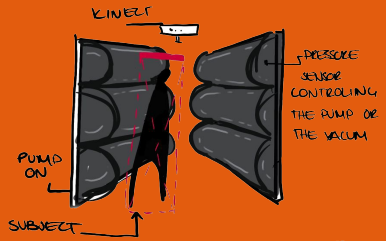
Institute for advanced architecture of Catalonia

1.3 CONCEPT

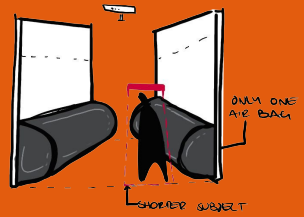
<an interactive soft doorway that senses human behaviour and responds to permit or deny passage>



1



2 ACCESS NOT ALLOWED



3 AMOUNT OF AIR DEPENDING THE HEIGHT OF THE SUBJECT

4 ACCESS ALLOWED = AIR PUMP OFF

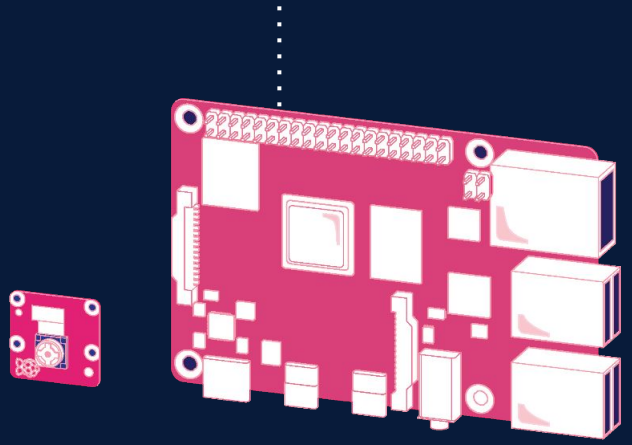
1.3 CONCEPT

```
# Pseudocode:

-Detect human face
-Inflate
-Stop
- Analyse facial
expression
-If smile detected,
deflate
```

4.1 FUNCTIONAL DIAGRAM

input:

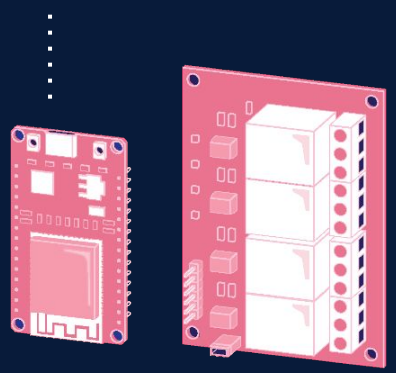
[camera sensor + Raspberry Pi]



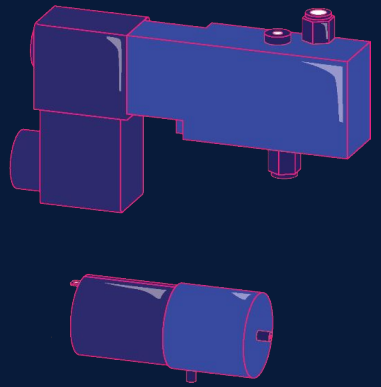
output

[ESP32 > relay]

pump/vacuum]



[air]

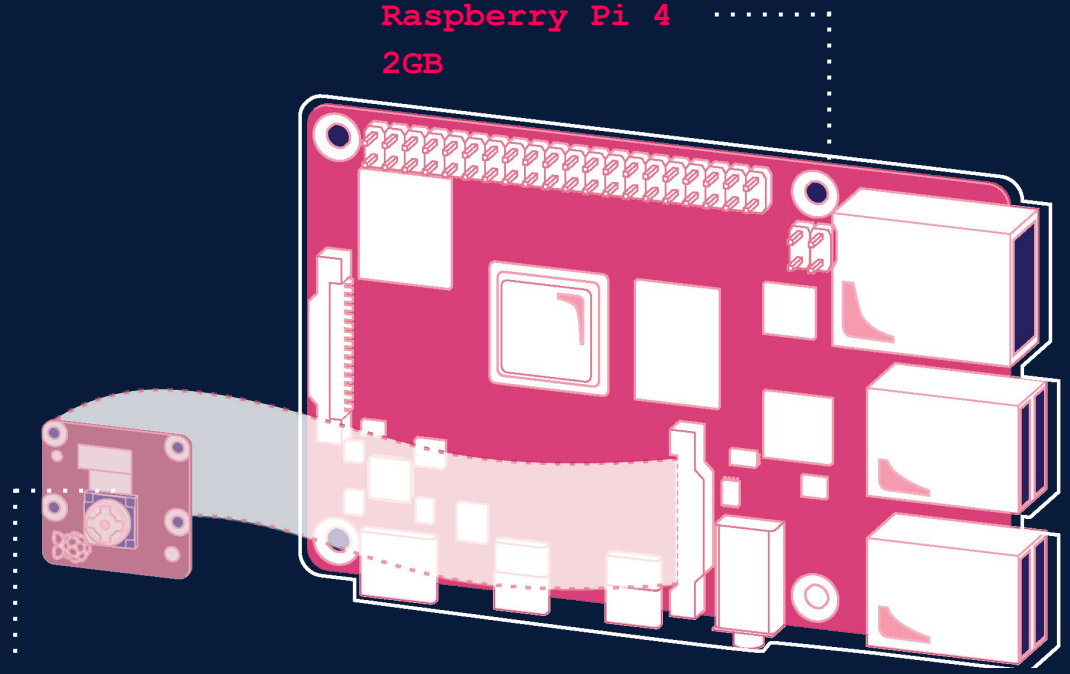


4.1 INPUT

<"The data analysis will happen on the Raspberry Pi, which will need its own operating system.">

requirement :

- Install Bullseye 64 bit
- Install zram
- SSH on
- Wireless LAN updated
- Install OpenCV - build from source
- Install mosquitto-clients
- Install paho-mqtt
- Smile_recognition.py onto RPi
- $ Mosquitto -c MQTT/mosquitto.comf -d $ python smile_recognition.py

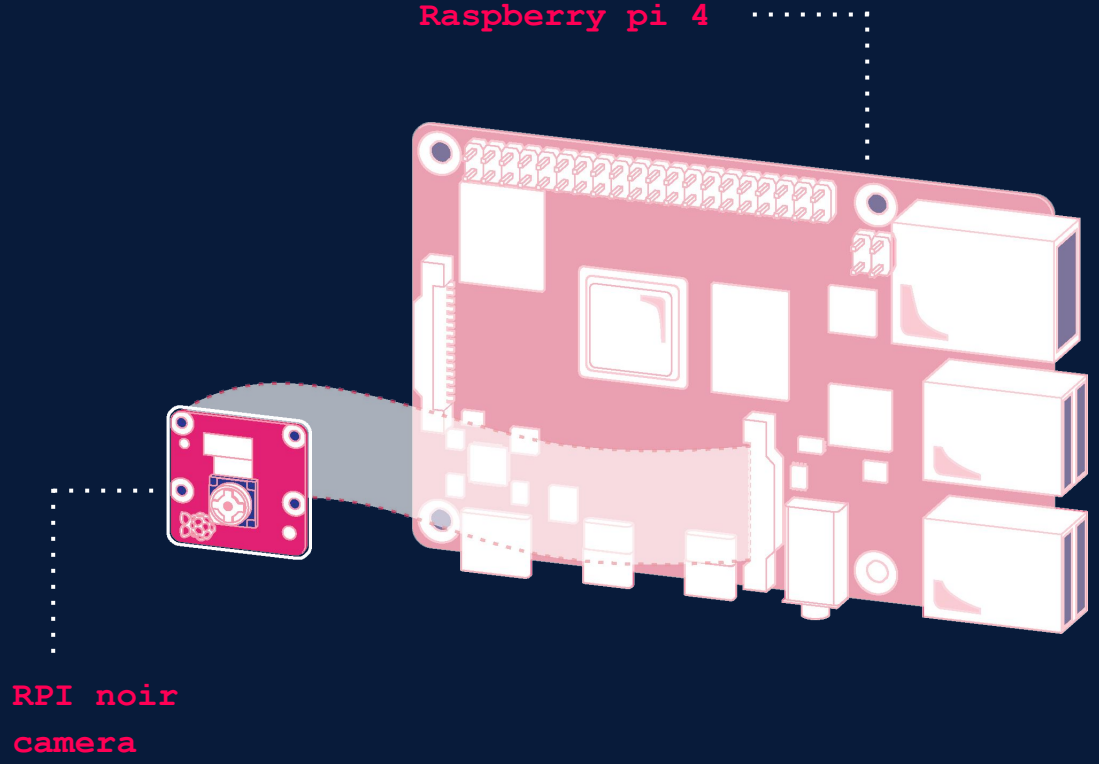


4.1 INPUT

<”The data analysis begins with detecting a human figure. Then the trigger is defined by the expression of the individual.”>

PSEUDOCODE

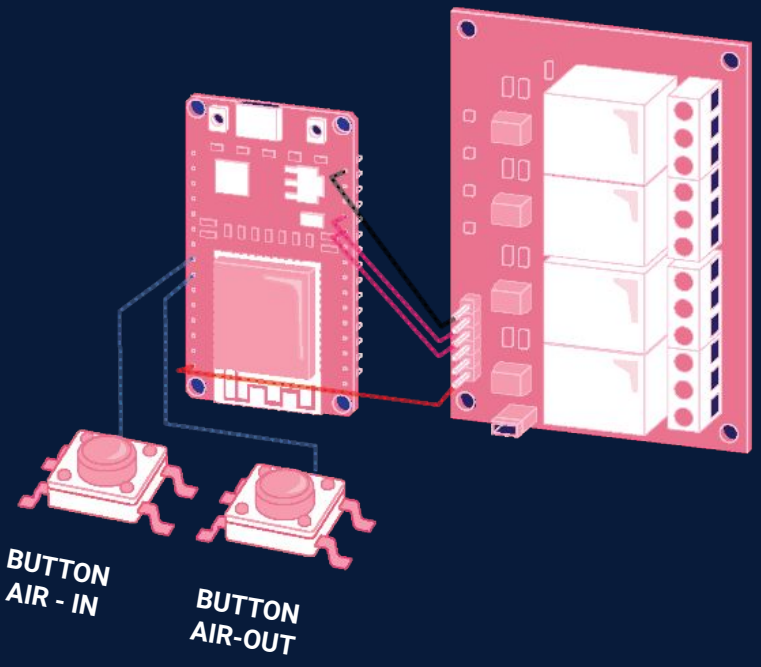
```
     Video = cv2.VideoCapture(0)
Try:
      face_cascade =
cv2.CascadeClassifier
      Smile_cascade =
cv2.CascadeClassifier

While True:
     check, frame = video.read()

      If len (smile) > 0:
          print (“smile detected”)

      elif len (faces) > 0:
            print (“face detected”)

client.publish
```



4.1 OUTPUT

<"The output side began with one button imitating the "air_in" signal, another button imitating "air_out".>

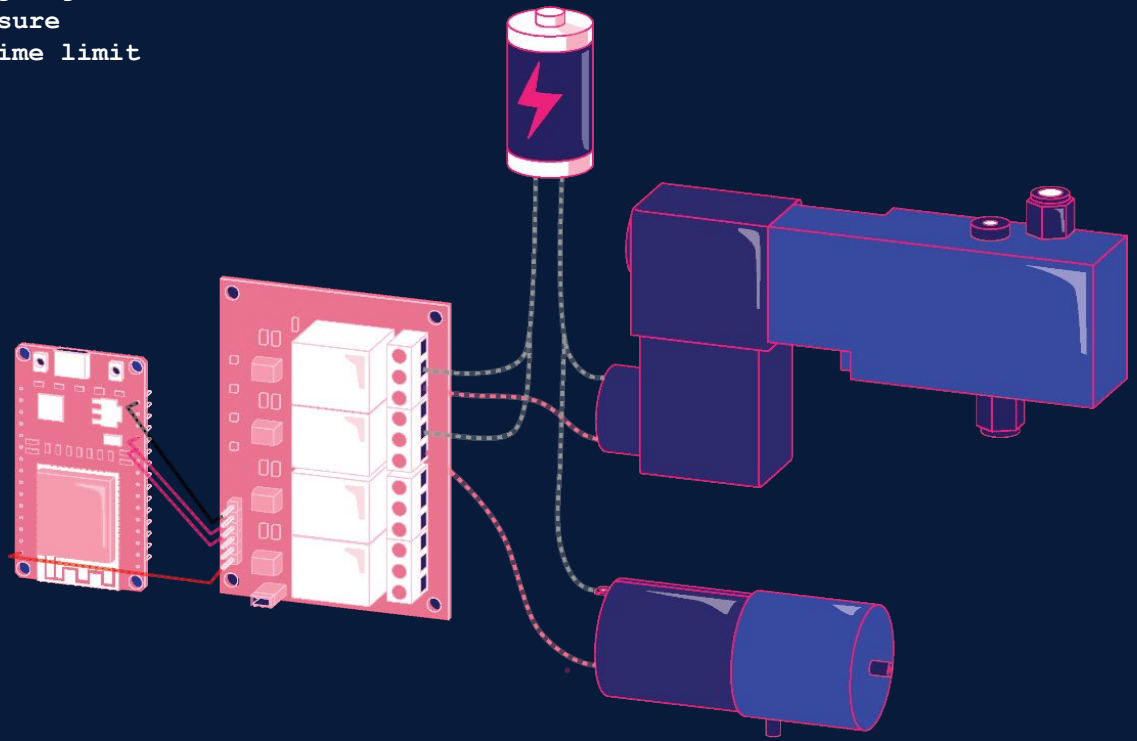
```
> Button1 = Pin 16.Pin OUT = "Air_In"
> Button 2 = Pin 17.Pin OUT =
"Air_Out"

> Solenoid = Pin 12.Pin OUT
> DC_Pump = Pin 14.Pin OUT

> While code runs:
      >if button1.pressed:
            Solenoid.value(0)

      >elif button2.pressed:
            DC_Pump.value(0)

      >else:
            Solenoid.value(1)
            DC_Pump.value(1)
```



4.2 HARDWARE SCHEMATICS

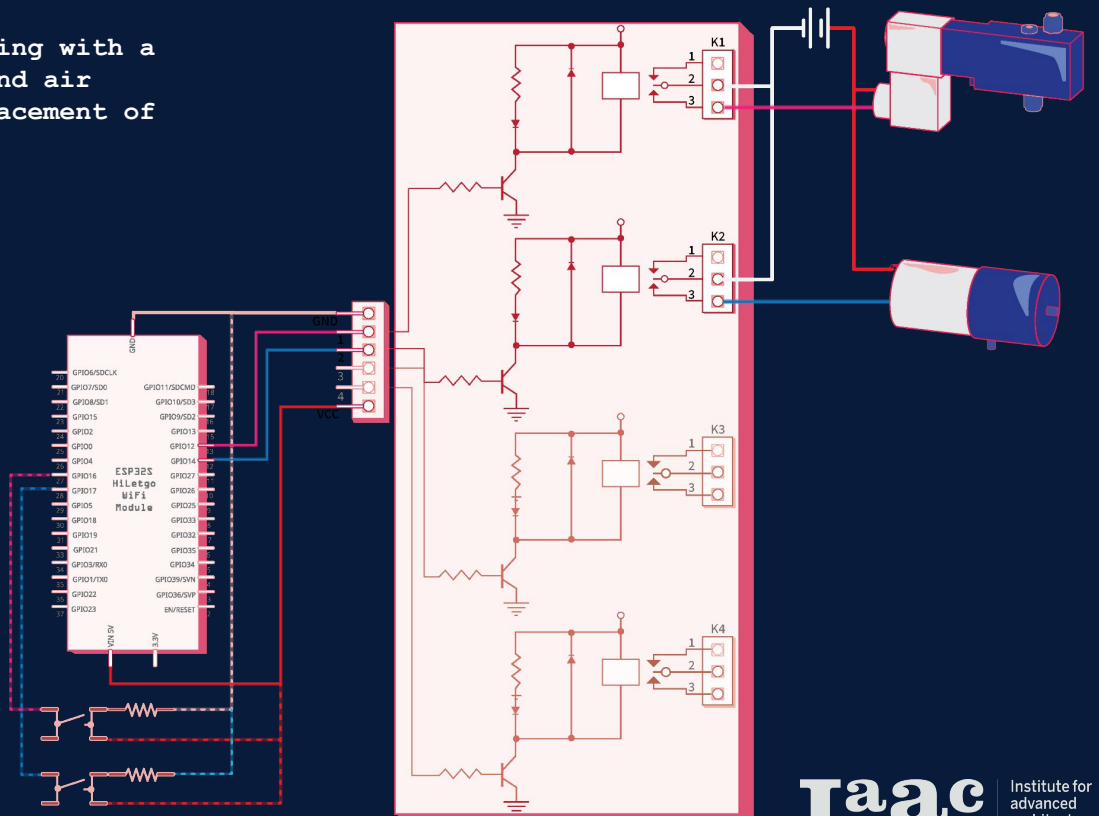
<"We chose a 12V DC air pump as vacuum, going with a 12V solenoid to blow air in. Due to pressure inequality on both ends, we had to set time limit for air flowing in.">

```
>start = time.time()

> While code runs:
        Solenoid.value(1)
        DC_Pump.value(1)

        >if signal == "air_in" :
                pressed = time.time()

                elapsed = pressed -
start

                if elapsed < 3 sec and
                signal != "air_out:
                        solenoid.value(0)
                elif elapsed > 3 sec:
                        solenoid.value(1)

        >if signal == "air_out":
                …
```



4.2 HARDWARE SCHEMATICS

<”We chose a 12V DC air pump as vacuum, going with a 12V solenoid to blow air in. The signals and air pressure was tested with the buttons (replacement of MQTT signal).”>



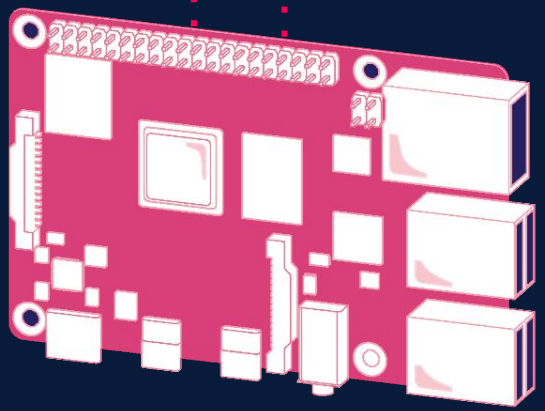
4.3 MQTT SETUP

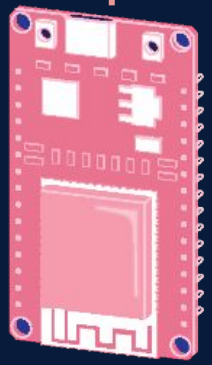
<”Output of face detection, as well as the ‘buttons’ of the ESP32, had to unify as topics for MQTT communication. While MQTT setup for raspberry had more resources, such as paho library, ESP32 required work-arounds.”>

MQTT BROKER

mqtt_publish.py

mqtt_subscriber.py





RaspberryPi setup requirement:

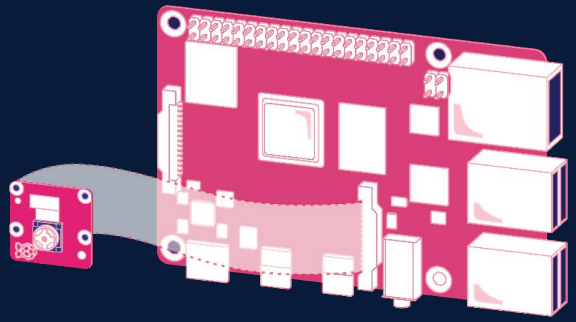
#install mosquitto-clients
#publish.py
- import paho-mqtt.client
- Make sure: client.connect == 0
- Setup: connection_flag (client.connected_flag)

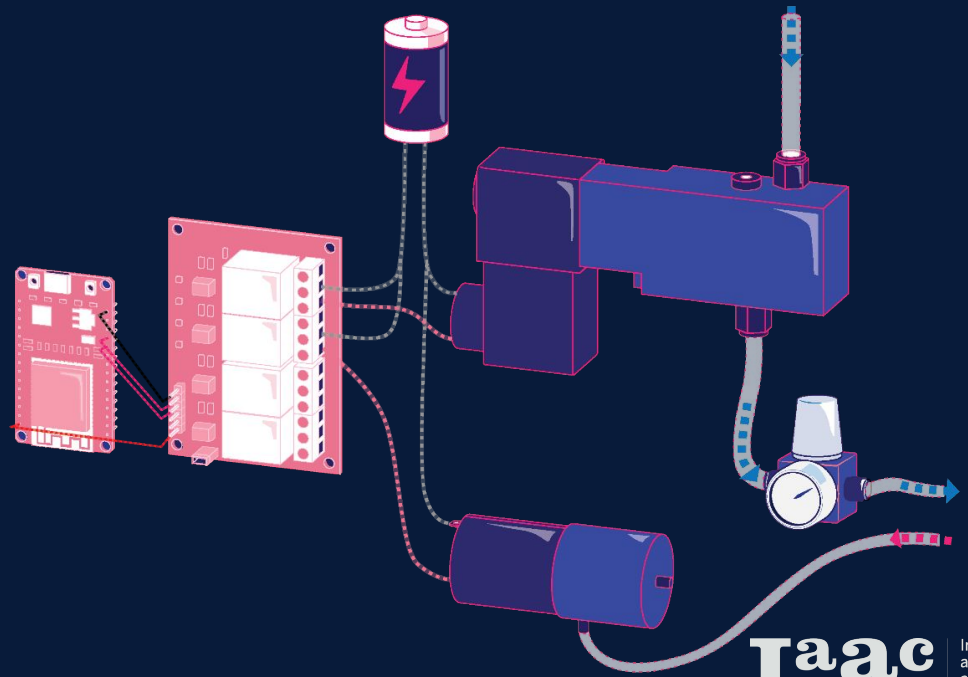
ESP32 setup requirement:

#Setup mqtt.py script as module for main.py
#Setup connectWiFi.py as module for main.py

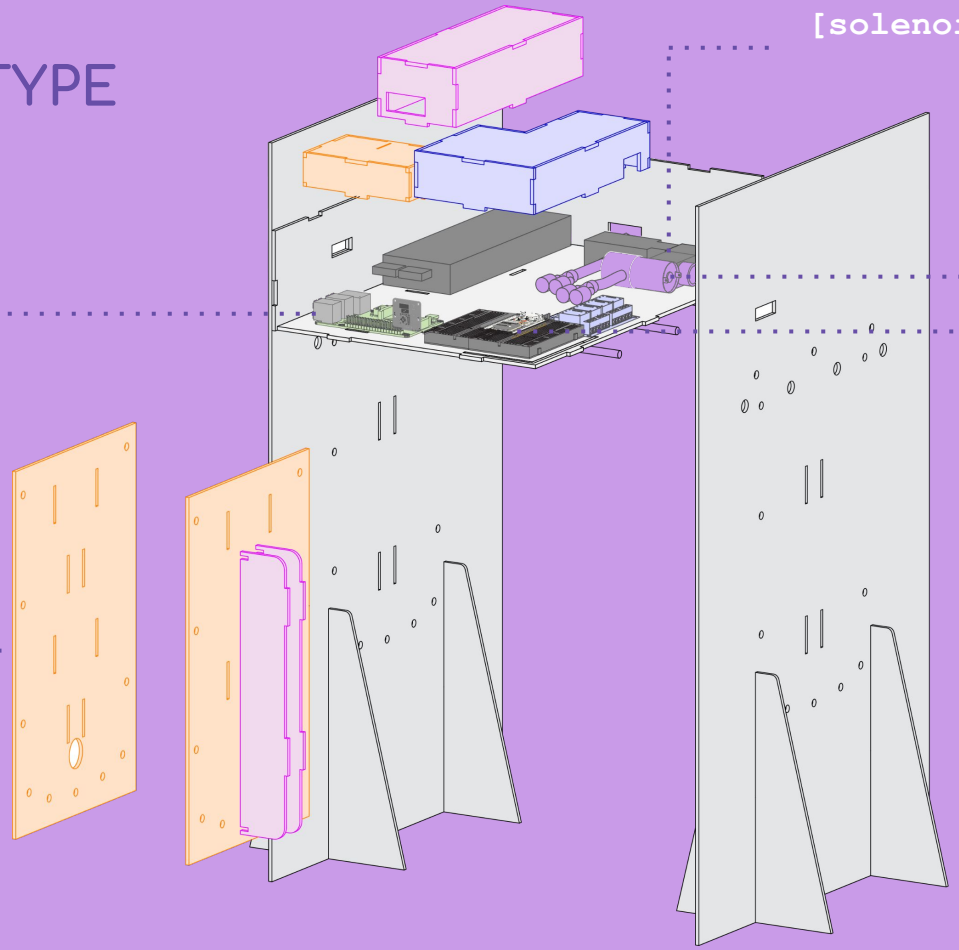
4.4 OVERALL DIAGRAM

<”With the MQTT signal setup, the communication between the electronics is set.”>



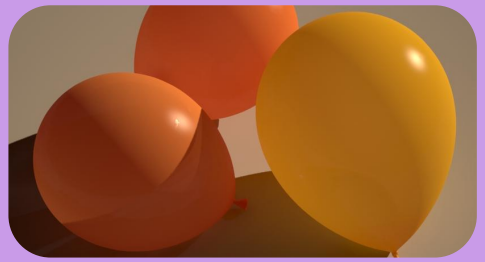


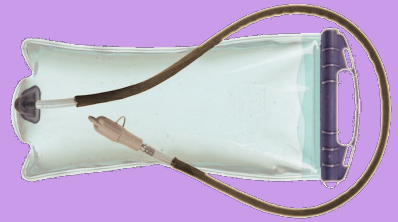
5.1 THE PROTOTYPE

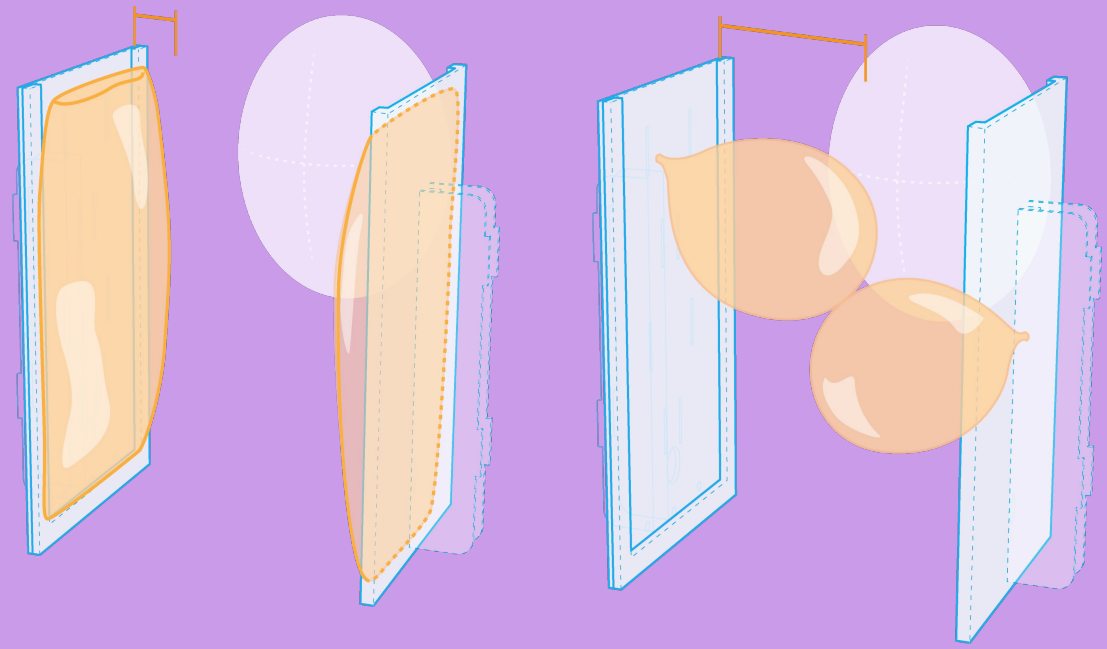


5.1 THE PROTOTYPE

< Materials we tested were latex, rubber balloons, and vinyl camel bag.>







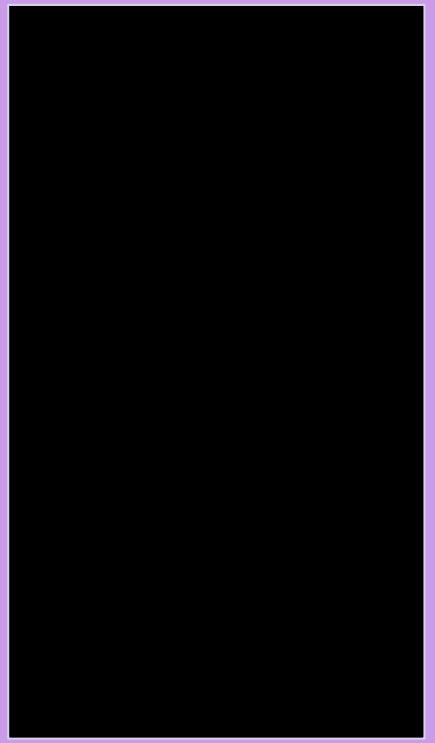
The choice of inflatable was influenced by the inflation behaviour, pneumatic performance. Each type of inflatable has different implications of the frame design,

5.2 MATERIAL OPTIONS

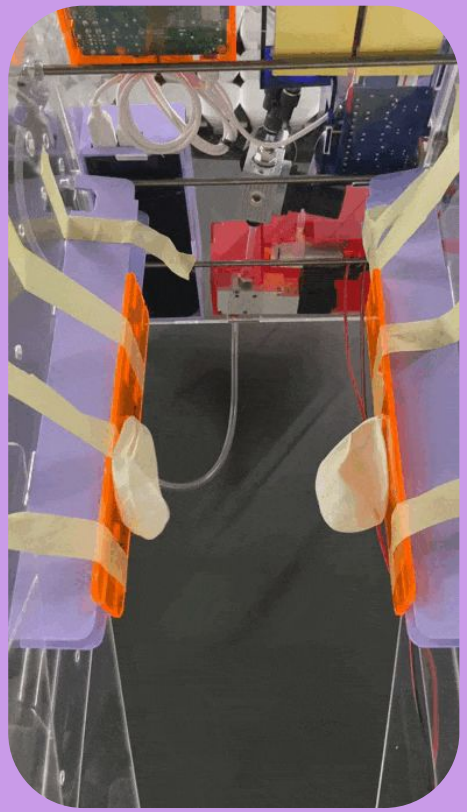
#Latex

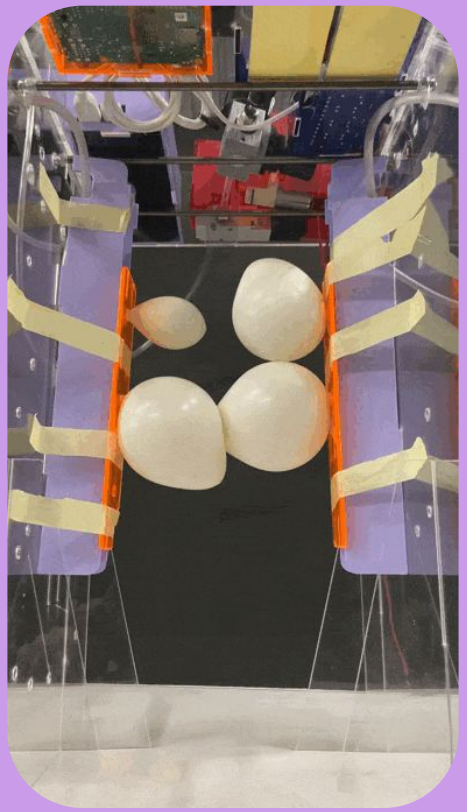
#Rubber balloon

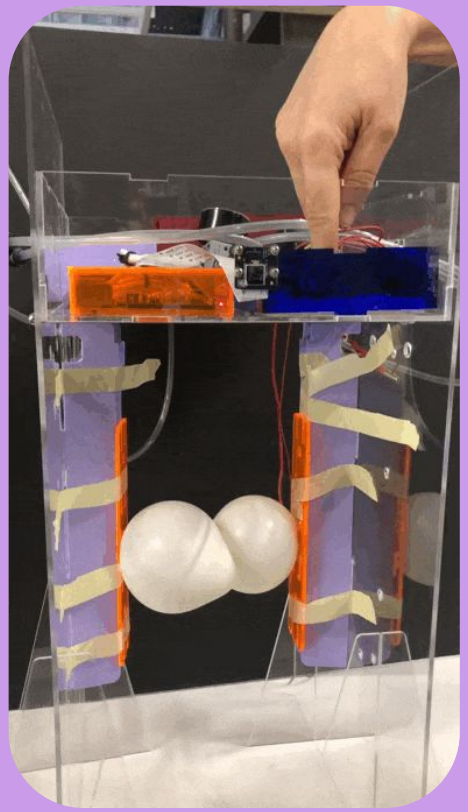
#Vinyl Camel bag



5.1 THE PROTOTYPE



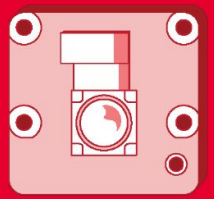




5.1 THE PROTOTYPE

5.1 THE PROTOTYPE

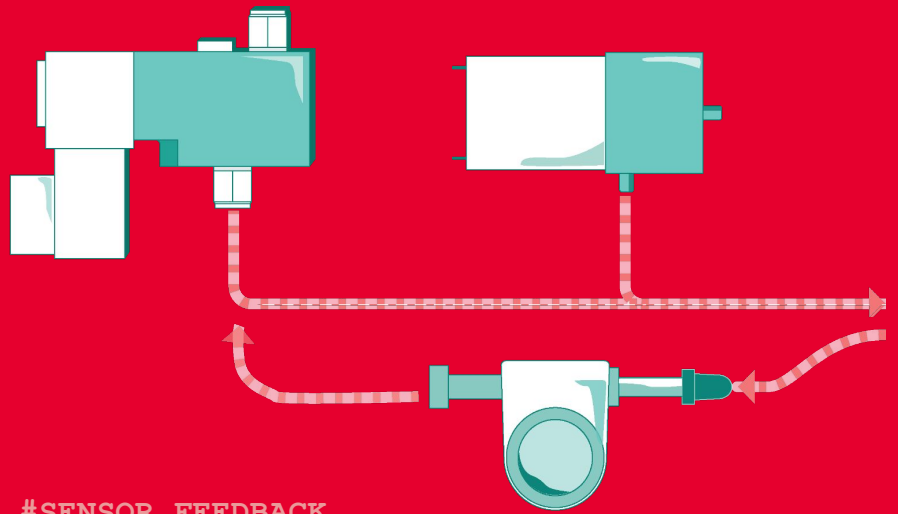
6.0 NEXT STEPS



#DETECTION

```
SMILE: #sincerity detection
      If sincerity == 0:
            deflate
      Else:
            inflate
```

```
("Hello world")
```



#SENSOR FEEDBACK

```
#Added component, pressure sensor

... inflating ...
If reading(pressure_sensor) > ##:
      solenoid.value(1)
      dc_air.value(0)
```


REMEMBER TO SMILE
WITH YOUR EYES
AND . . .
THANK YOU