R.O.M

Room Occupancy Map

PROBLEM

Estimate the occupancy of a building by counting people entering in its rooms

GOALS

Reduce Economical Costs

There are many existing approaches to this problem:

- Some are implementing expensive components
- Other don't reach a sufficient grade of accuracy

Our objective is to reduce the cost and reach the same high accuracy

Provide information regarding the building in a quick and accessible way

This can be useful in a variety of ways, such as reducing the amount of time one has to spend in order to find a free room.

In our current time this could be used to monitor and enforce COVID-19 regulations

How to achieve this goals ?

We have designed different configurations and put them in comparison to see advantages and disadvantages

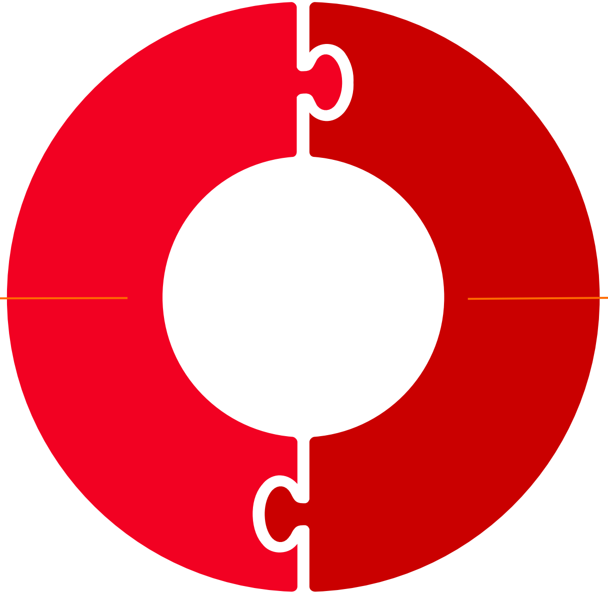
CONFIGURATIONS REQUIREMENT

The main requirement that is taken into account in comparing the different configurations is the **accuracy**

REQUIREMENTS

Physical placement

Mainly for safety reason, our system cannot be placed in such a way that it would ostracize the entrances.



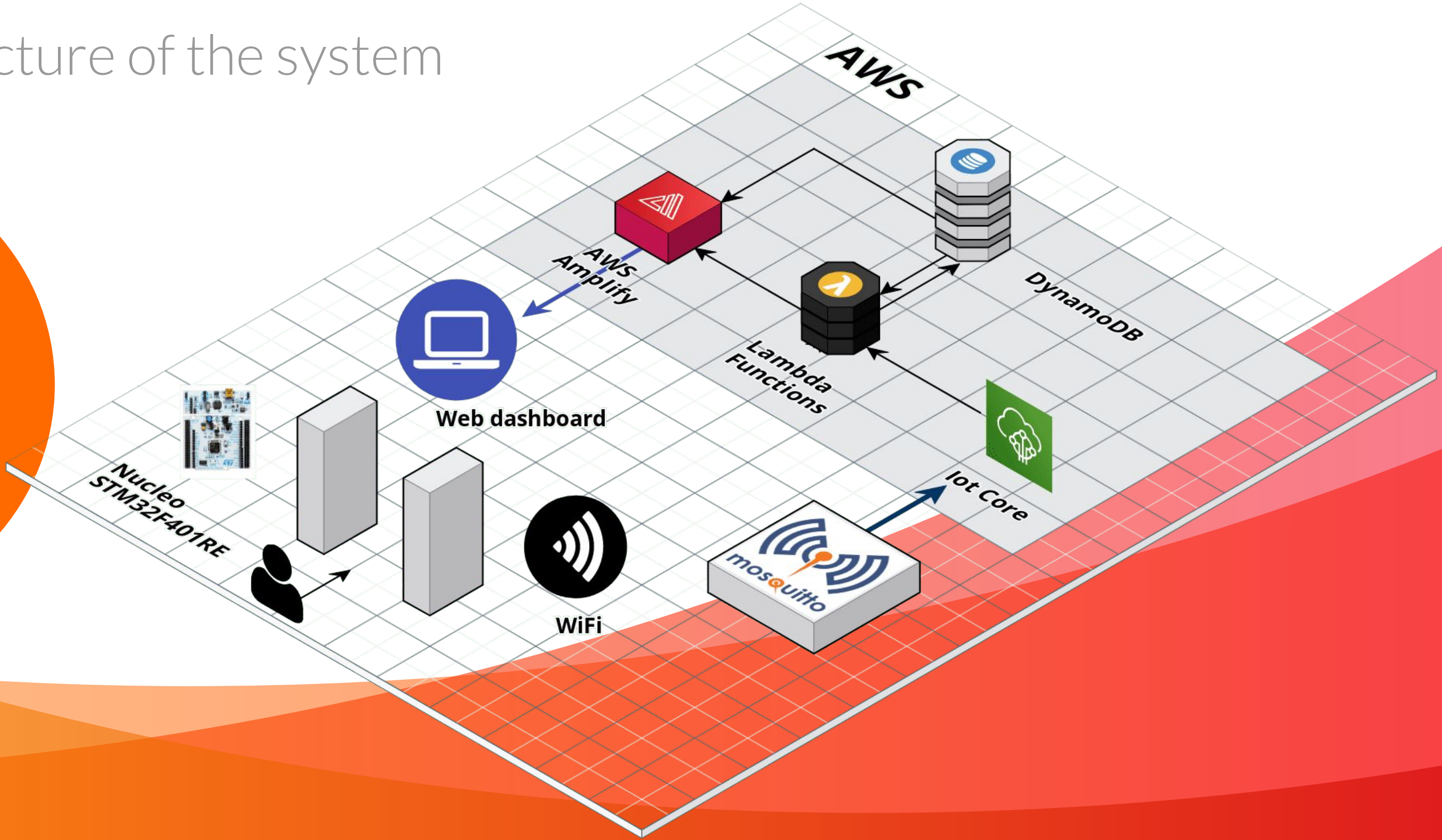
Real time update

The system can be considered working only when it’s providing meaningful data.

CLOUD NETWORK

The network structure of the system

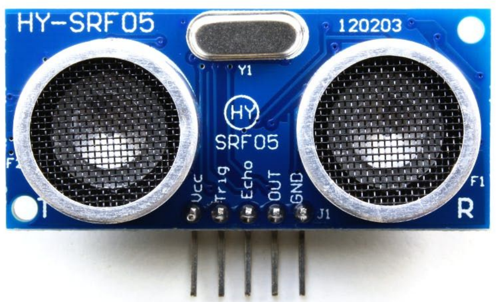
The configurations are different for their physical placement but maintain the same network structure



CONFIGURATIONS

Let’s start with the core of our work

SENSORS UTILIZED

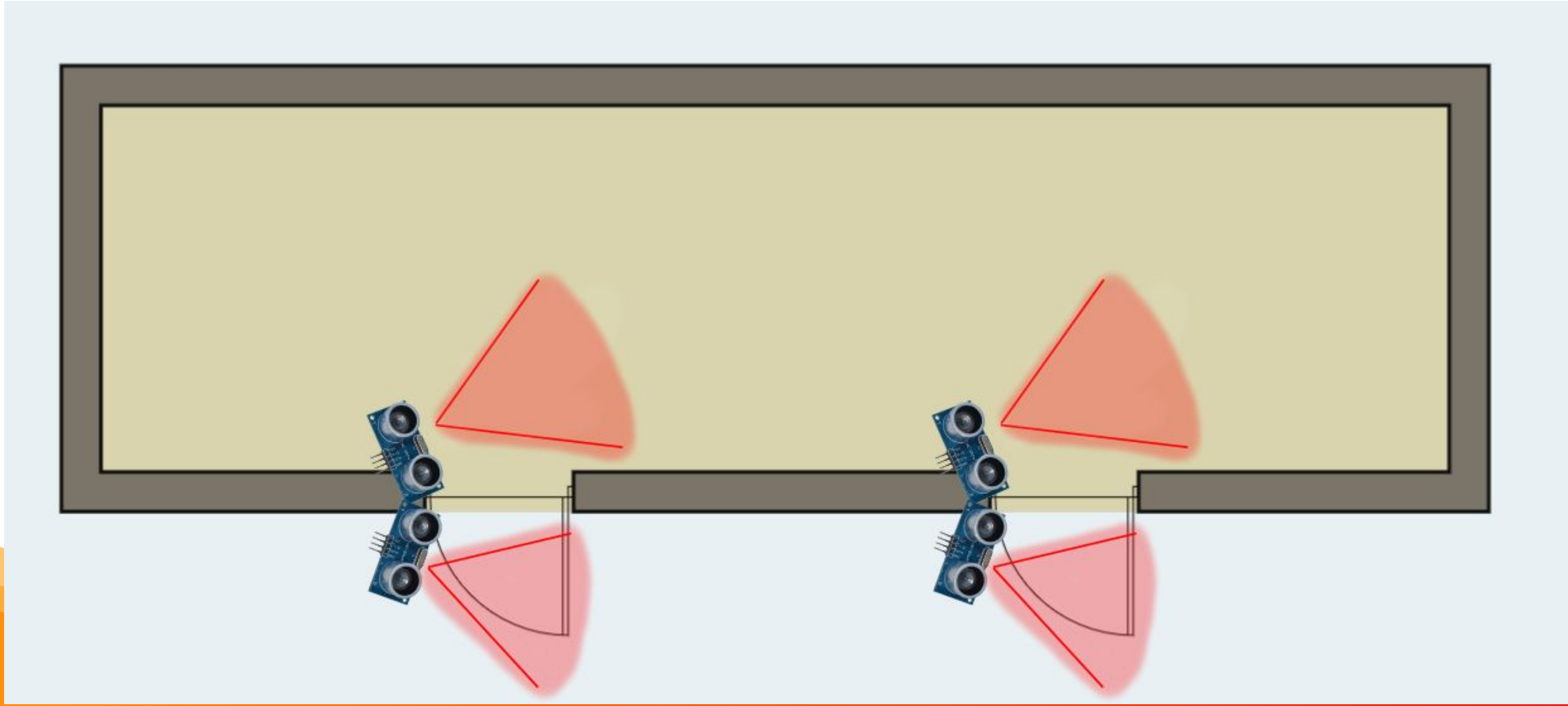


SRF05

Ultrasonic Sensor (x2)

- Range: 1cm - 4mt
- Angle: 30°

CONFIGURATION



ALGORITHM

Set initial flags Flag1 = 0 and Flag2 = 0;

Read the readings from both the Ultrasonic sensors and update the sequence Flag according to the trigger sequence of the sensors.

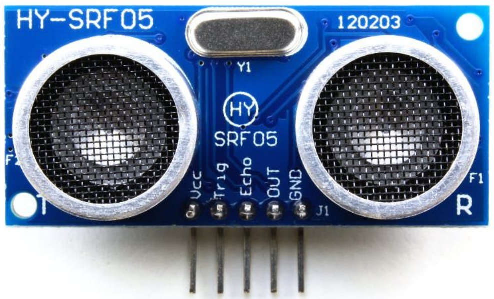
Flag1 == 1 and Flag 2 == 2 → Entry Motion;

Flag1 == 2 and Flag2 == 1 → Exit Motion;

Clear the flags to Flag1 = 0 and Flag2 = 0;

Wait 200 ms before next reading.

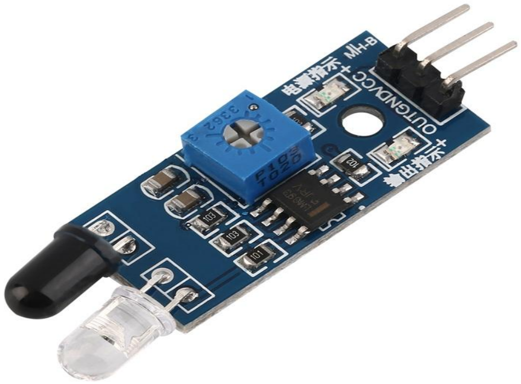
SENSORS UTILIZED



SRF05

Ultrasonic Sensor (x2)

- Range: 1cm - 4mt
- Angle: 30°

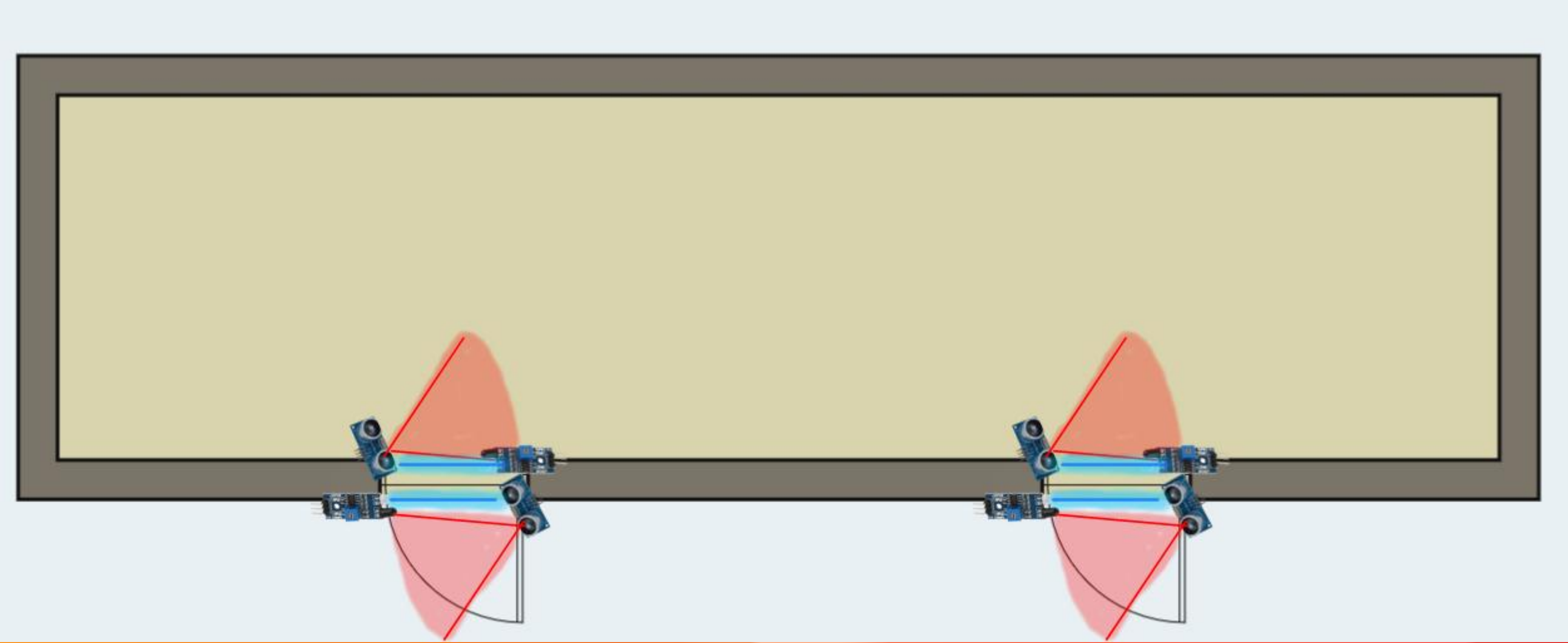


IR

Infrared Sensor (x2)

- Range: 3 mt
- Flexible sensibility

SECOND CONFIGURATION



ALGORITHM

Here we add two Infrared Sensors along with the Ultrasonic Sensors.

(Infrared sensors are used to increase the sensitivity of reading person walking in fast pace)

Set initial flags Flag1 = 0 and Flag2 = 0 for Ultrasonic Sensors and Flag3 = 0 and Flag4 = 0 for Infrared Sensors .

Read the readings from both the Ultrasonic sensors and both the Infrared sensors and update the sequence flag according to the trigger sequence of the sensors.

Flag1 == 1 and Flag 2 == 2 and Flag3 == 1 and Flag4 == 2;

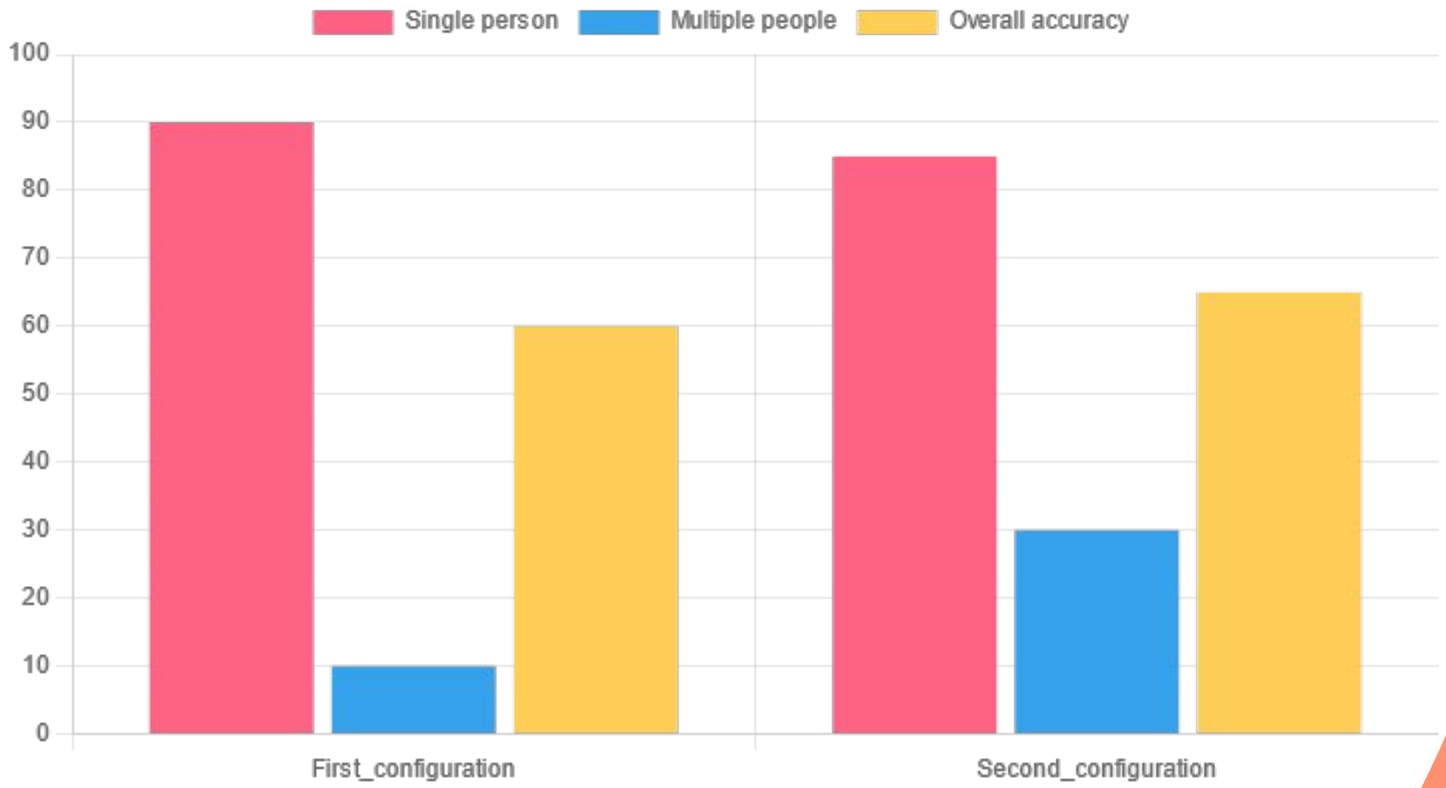
Flag1 == 2 and Flag 2 == 1 and Flag3 == 2 and Flag4 == 1;

Clear the flags to Flag1 = 0 and Flag2 = 0 and Flag3 = 0 and Flag4 = 0;

Wait 200 ms before next reading.

CONFRONTATION

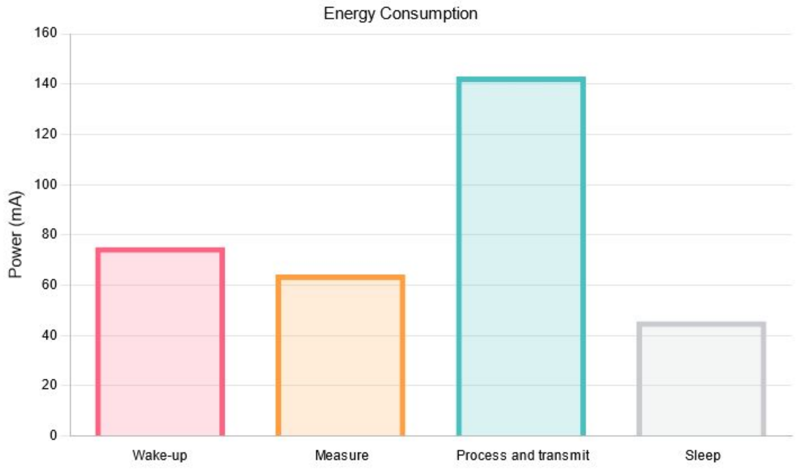
Evaluation, advantages and disadvantages of our configurations


Single person
Multiple people
Overall accuracy
100
90
80
70
60
50
40
30
20
10
0
First_configuration
Second_configuration

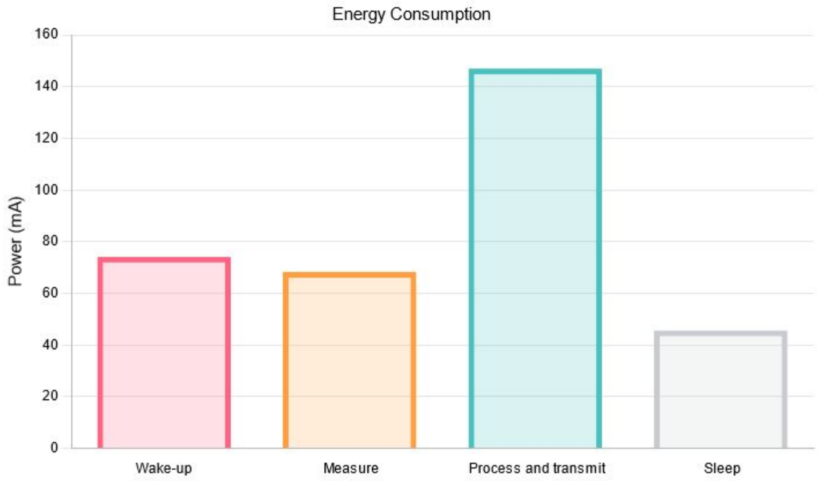
ACCURACY

POWER CONSUMPTION

- First configuration



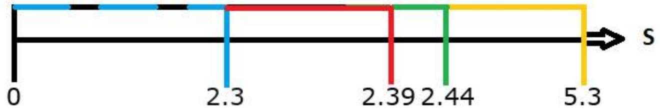
- Second configuration



UPDATE TIME

- First configuration

- Second configuration

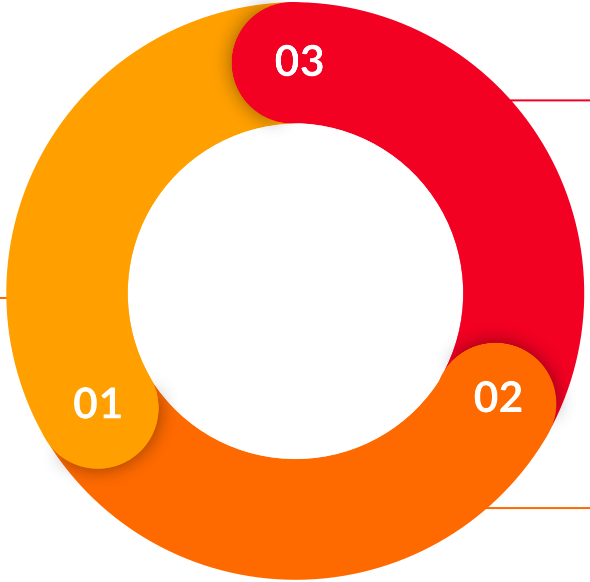


LEGEND

FUTURE UPDATES

What it’s still missing from the project
that keeps it from being complete

WHAT TO EXPECT



IMPLEMENT WIFI

As of now, the connectivity of our system relies on the native communication with the board. This has to be fixed as soon as possible

IMPLEMENT AND DOCUMENT OTHER CONFIGURATIONS

As of now, a third configuration is in the work.

PROVIDE A FULLY FLEDGED DEMO

A video that encapsulates the totality of our project

Thanks!

Any questions?