Problem Statement:- Keeping the track record of total person inside a room.

Introduction

Our project "Count total number of persons present in a room at a time" emerges as a significant technological solution.

- Using the capabilities of the Ultrasonic Sensor (HC-SR04) and the Raspberry Pi Pico W microcontroller, our attempt focuses on the development of a sophisticated people counting system.
- Aim is to provide real-time, precise, and automated tracking of individuals as they enter or exit a room. By combining hardware and software elements, our system offers a reliable tool for applications spanning from building security and crowd management.

Motivation

The motivation behind our project is due to the critical need for an efficient and reliable system capable of tracking individuals as they enter or exit a secured area or room, such as an ATM booth. Where more than one person should not be present at a time.

Architecture

We have used two inexpensive sensors for this. The hardware we have used:

- Ultrasonic Sensor (HC-SR04)
- Raspberry Pi Pico W
- Jumper Wires
- Type-B cable
- Desktop/laptop

And for software we have used Thonny IDE. There are four terminals for the connection of the ultrasonic sensor:

- VCC (Power Supply): This pin is connected to the power supply, usually 5V. It provides the necessary voltage for the ultrasonic sensor to function.
- GND (Ground): The GND pin is connected to the ground or 0V reference of the power supply.
- TRIG (Trigger): The TRIG pin is an input pin that triggers the ultrasonic sensor to send out an ultrasonic pulse of 10µs. You typically send a short high-level pulse to this pin to initiate a distance measurement.
- ECHO (Echo): The ECHO pin is an output pin that senses the ultrasonic pulse after it reflects off an object. By measuring the duration of the high level on this pin, you can calculate the distance to the object.

How ultrasonic sensor works?

It measure distance **by using ultrasonic waves**. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target.

Sensor 1:

Vcc-5 volt

Trig-0

Echo-1

Ground- Ground

Sensor 2:

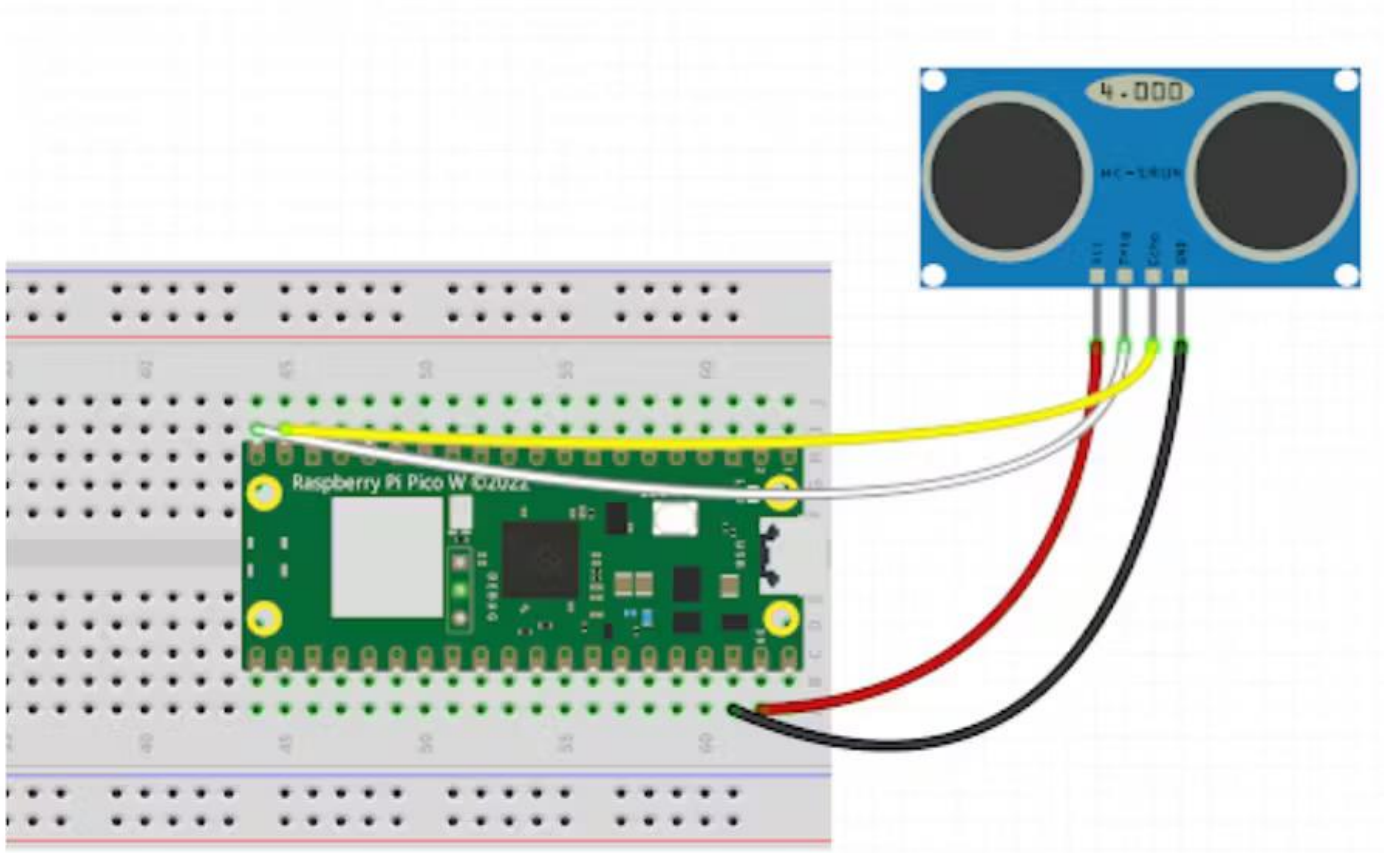
Vcc-5 volt

Trig-2

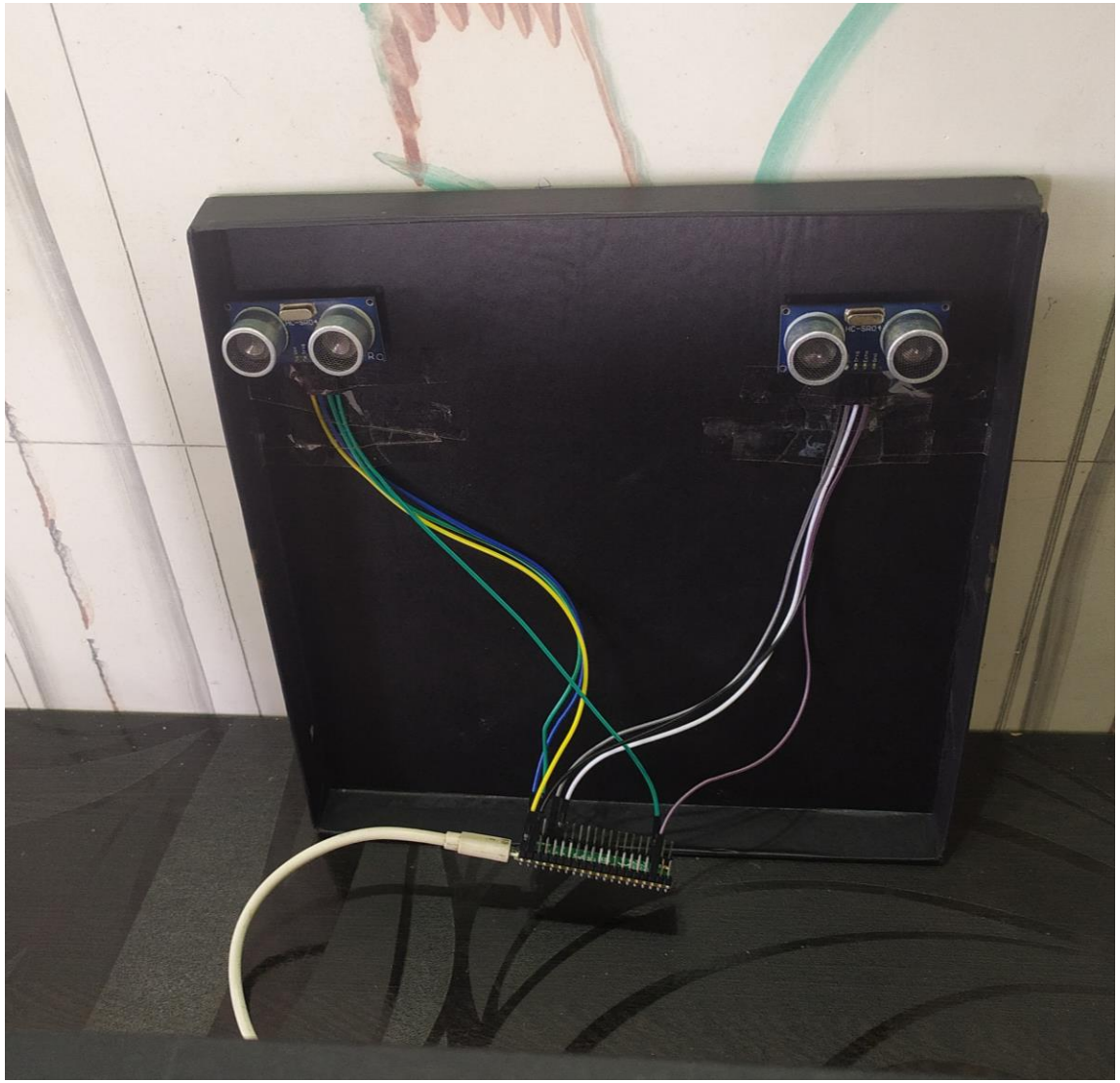
Echo-3

Ground- Ground

Connected Raspberry Pico W to the laptop through type-b cable

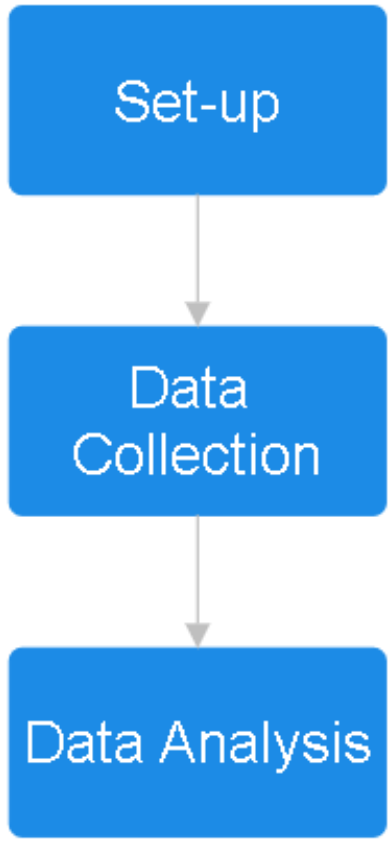


We attached the two Ultrasonic sensors to the cardboard with 5-6 cm gaps between them.



Framework

- We complete the full set up by using hardware like (Raspberry Pico W, jumper wires, 2 ultra-sonic sensor).
- We collect the data with the help of python code written in Thonny IDE which is then fetched in CSV file.
- We analyze the working of program how the person is entering and exiting a room with respect to time.



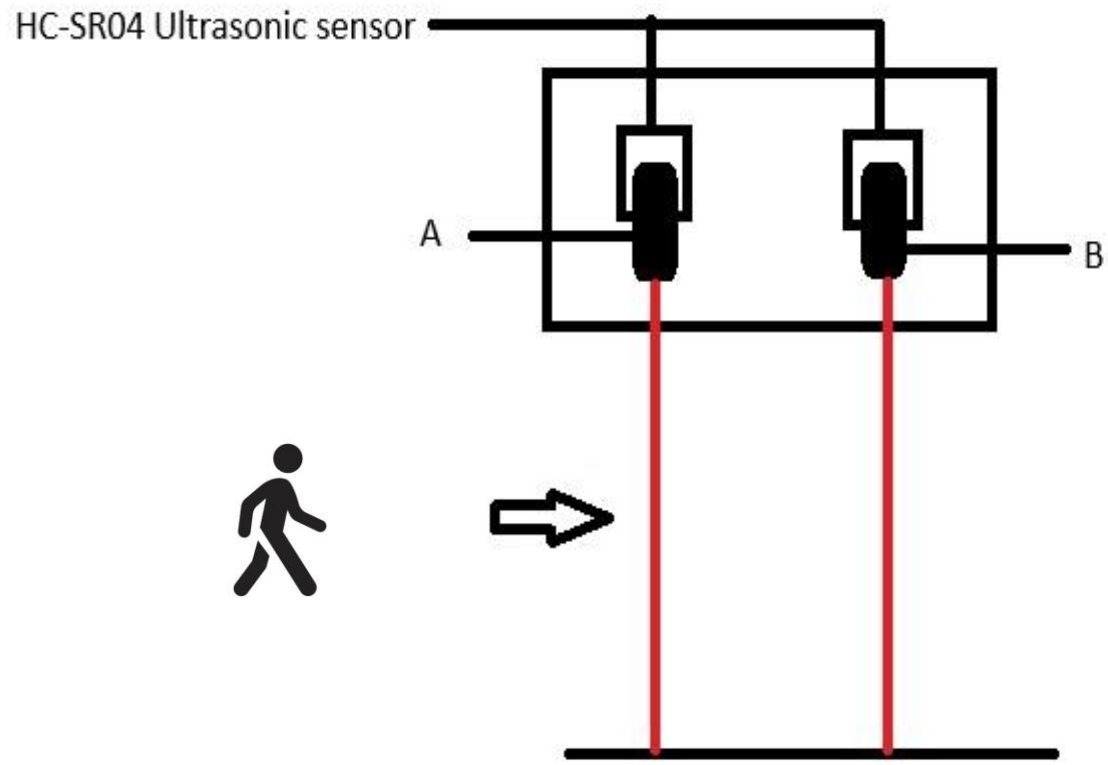
Working process

When a person or an object passes through both the sensors then either it enters or exits from the room.

Set-Up:- Sensor 1----> Sensor 2-----> Room

<u>Conditions:-</u>

1. Person marked as entered if it passes through sensor 1 to sensor 2
2. Person marked as exit if it passes through sensor 2 to sensor 1



The basic principle is to distinguish between *IN* and *OUT* of the person or people.
Suppose the sensors in the following figures, A and B. If the A sensor detects a person first, it means *IN*. And the reverse, is *OUT*.

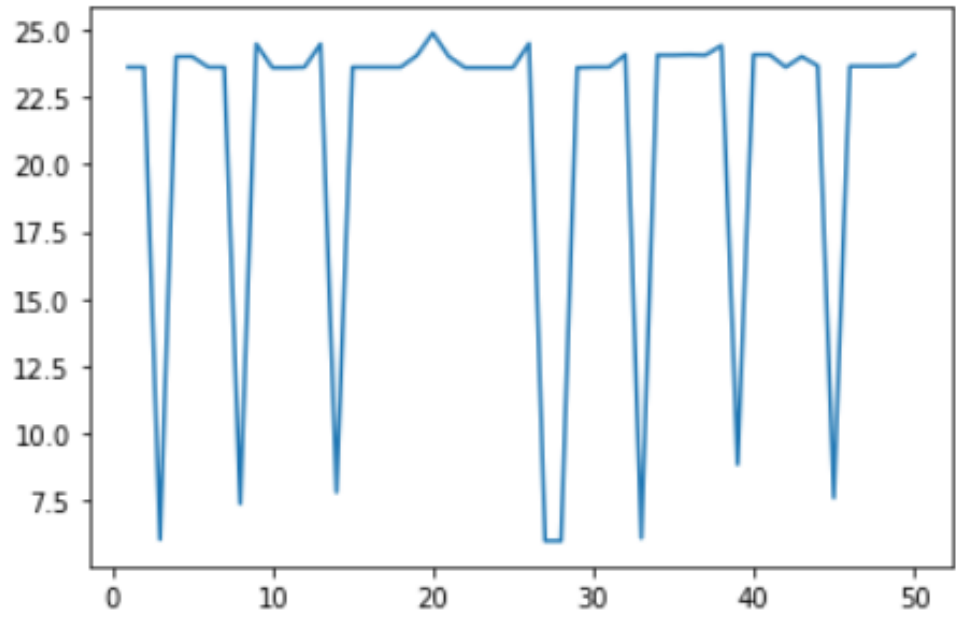
Data Acquisition

SL NO .	Sensor1 (cm)	Sensor2 (cm)	Activity
1	23.62069	23.18966	
2	23.62069	23.18966	
3	6.034483	23.18966	
4	24.01724	3.775862	
5	24.01724	3.775862	Entered
6	23.62069	22.72414	
7	23.62069	22.77586	
8	7.362069	23.18966	
9	24.48276	23.18966	
10	23.60345	5.793103	
11	23.60345	5.793103	Entered
12	23.62069	23.18966	
13	24.48276	22.72414	
14	7.793103	23.2069	
15	23.62069	23.18966	
16	23.62069	6.396552	
17	23.62069	6.396552	Entered
18	23.62069	22.32759	
19	24.05172	22.77586	
20	24.89655	22.7931	
21	24.01724	23.12069	
22	23.60345	6.793103	
23	23.60345	22.7069	
24	23.60345	22.72414	
25	23.60345	22.77586	
26	24.5	6.293103	
27	5.982759	22.31034	
28	5.982759	22.31034	Exits
29	23.60345	22.77586	
30	23.62069	23.24138	
31	23.62069	23.24138	
32	24.08621	23.17241	
33	6.086207	23.17241	
34	24.06897	5.172414	
35	24.06897	5.172414	Entered
36	24.08621	22.7931	
37	24.06897	22.77586	
38	24.43103	22.7069	
39	8.827586	23.24138	
40	24.08621	6.241379	
41	24.08621	6.241379	Entered
42	23.62069	22.74138	
43	24.01724	22.72414	
44	23.67241	22.82759	
45	7.586207	22.7931	
46	23.65517	6.810345	
47	23.65517	6.810345	Entered
48	23.65517	22.77586	
49	23.67241	23.24138	
50	24.08621	22.36207	
51	24.06897	5.741379	
52	6.5	22.82759	
53	6.5	22.82759	Exits
54	24.08621	23.18966	
55	23.62069	22.77586	
56	2	23.62069	
57	24.08621	23.18966	
58	24.08621	22.77586	
59	24.08621	22.75862	

Line Graph representation of Sensor1 data

```
In [5]: data=pd.read_csv('data.csv')
        data=data.head(50)
        plt.plot(data['SL_NO'],data['Sensor1(cm)'])
```

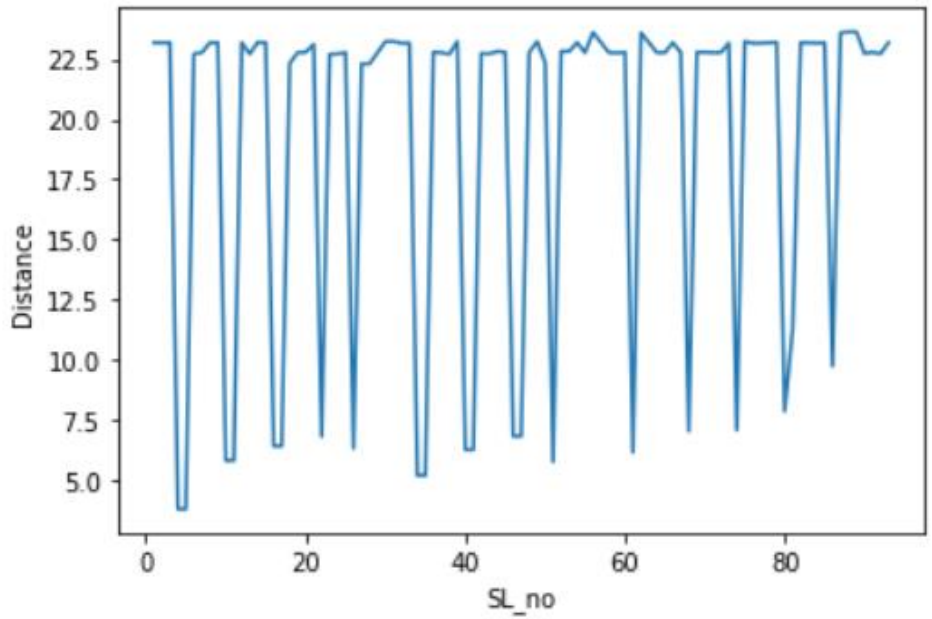
```
Out[5]: [<matplotlib.lines.Line2D at 0x16ff40f2e50>]
```



Line Graph representation of Sensor2 data

File Edit View Insert Cell Kernel Help

In [12]:
```
plt.plot(data['SL NO . '],data[' Sensor2 ( cm ) ']);
plt.xlabel('SL_no')
plt.ylabel('Distance');
```



Conclusion

In conclusion, our project successfully demonstrated the utility of Ultrasonic HC-SR04 sensors in conjunction with the Raspberry Pi Pico W for accurately counting the number of people present in a room. By implementing a two-sensor system to track entries and exits, we have created an efficient and cost-effective solution for real-time people counting.

This technology offers valuable applications in security, occupancy management, and data-driven decision-making. With its affordable components and reliable performance, our system represents a practical approach to enhancing room occupancy monitoring and contributing to a more secure and efficient environment.

Future work

In the context of our project on counting the number of people entering and exiting a room using the Ultrasonic Sensor (HC-SR04) and Raspberry Pi Pico W, our proposed future work involves:

- ✓ The application of machine learning techniques to address the challenge of accurately counting individuals when multiple people enter or exit the room simultaneously.
- ✓ Creating alarming system in android application if it exceeds the limit.

Furthermore, we can use other machine learning algorithms to extend our work.