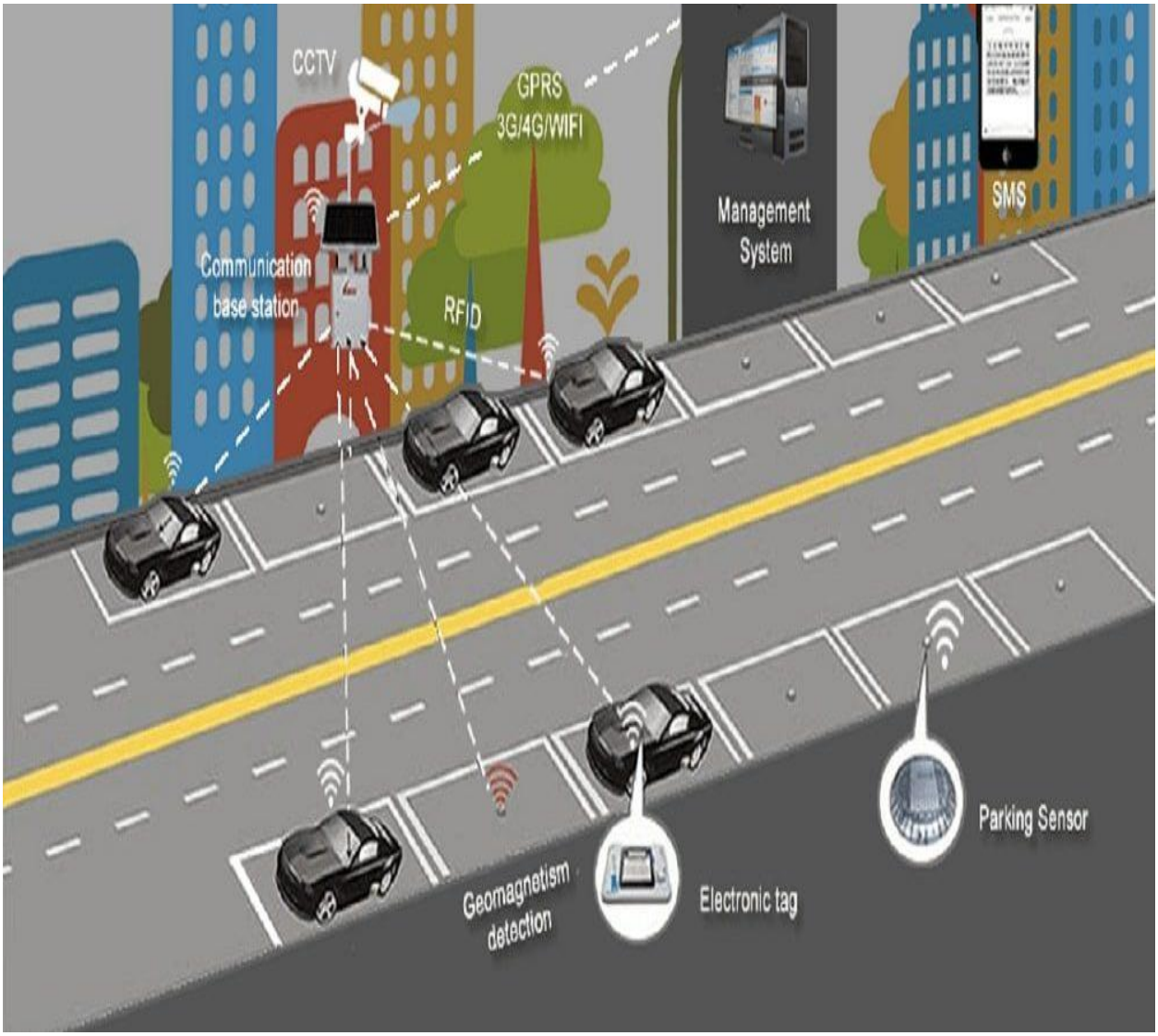
DOCUMENTATION

SMART PARKING USING IOT

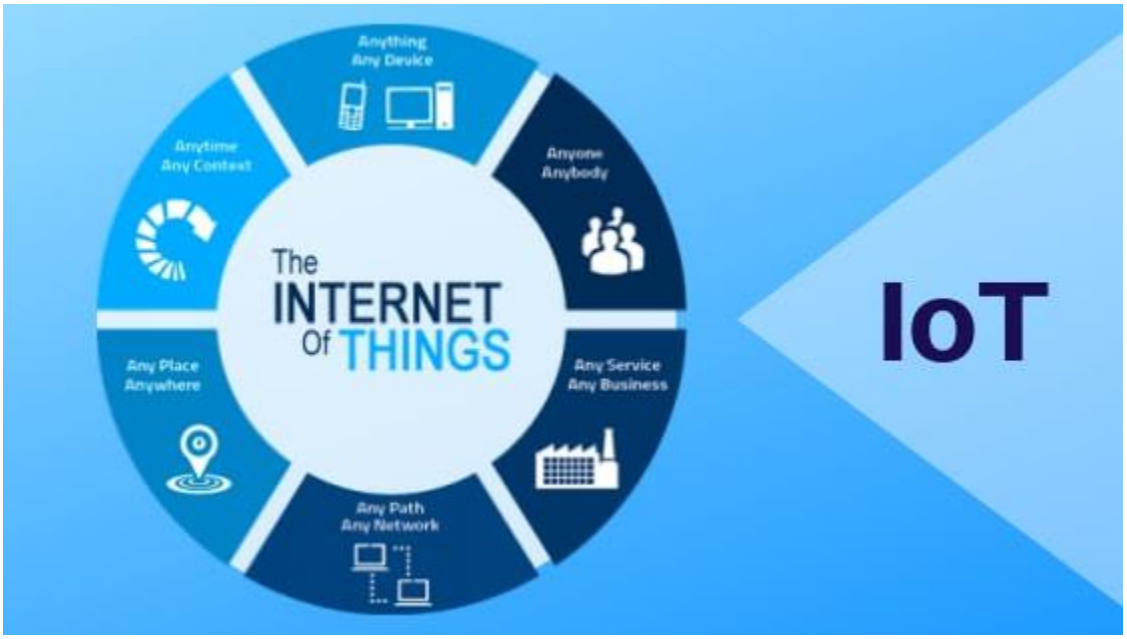


TEAM DETAILS

Mentor	Mrs.M.Maheswari
Leader	Monika M (310121104063)
Members	Anupriya R (310121104009) Bhavani G (310121104014) Divyabharathi C L (310121104028) Kanimozhi S (310121104048)
Problem Description	Describing the project objectives,IOT sensor setup,Raspberry pi integration and code implementation by including diagrams,schematics and screenshots of the IOT sensors and web application.

INTERNET OF THINGS (IOT):

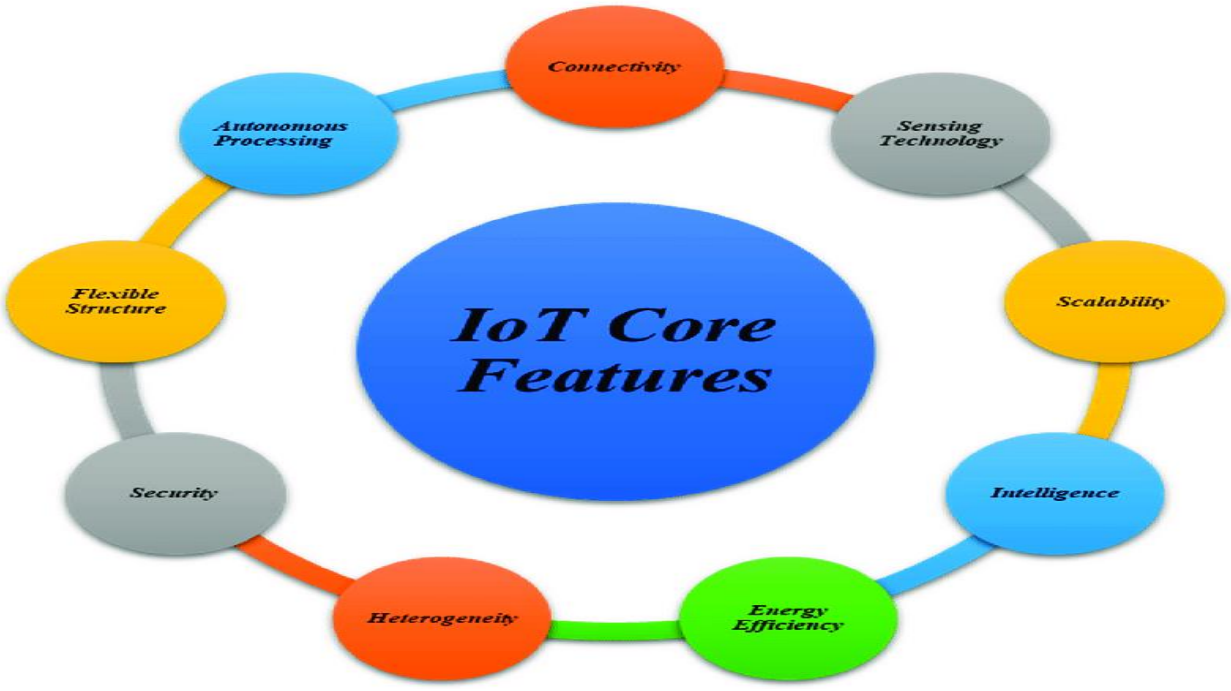
- The Internet of Things (IoT) describes the network of physical objects that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.



- Applications of IOT are diverse and include smart homes,industrialautomation,healthcaremonitoring,agriculture and smart cities.
- It has the potential to improve efficiency,reduce cost and create new possibilities for innovation across various industries.

KEY ASPECTS OF IOT:

1. **Connectivity**: IOT devices are connected to the internet for communication.
2. **Sensors and Actuators**:Devices are equipped with sensors to collect data and actuators to perform actions based on that data.
3. **Data Processing:** IOT devices often process data locally or send it to a central server or cloud for analysis.
4. **Automation:** IOT enables automation by allowing devices to make decisions and perform tasks without human intervention,



SMART PARKING USING IOT:

- Smart parking using IOT involves the integration of internet of things technologies to optimize the management and utilization of parking spaces.

OVERVIEW OF WORKING:

1. **Sensors:** IOT sensors such as ultrasonic sensors are deployed in parking spaces to detect the presence or absence of vehicles.
2. **Connectivity:** These sensors are connected to a network allowing them to transmit real-time data to a central server or cloud.
3. **Data Processing:** The central server often powered by IOT devices like Raspberry pi ,process the incoming data to determine the occupancy status of each parking space.
4. **User Interfaces:** Information about parking space availability made accessible to users through various interfaces. This can include mobile apps and websites
5. **Real-Time Updates**: Users can receive real-time updates on available parking spaces,reducing the time spent searching for parking and minimizing traffic congestion.

GOAL:

- The ultimate goal of a smart parking system in today's world is
    - To enhance urban mobility
    - To reduce traffic congestion
    - To improve the overall parking experience
    - To develop user friendly automated system
    - To eliminate the unnecessary traveling of vehicles across the filled parking slots
    - To offer safe and secure parking slots with in a limited area
    - To reduce fuel consumption and lower emissions by minimizing the time vehicles spend to find parking spaces

- In overall the major goal is to leverage technology and data to create a more efficient,sustainable,and user-centric approach to urban parking,ultimately contributing to the advancement of smart cities

ABSTRACT:

- In this innovative project, we propose a smart parking system leveraging the power of Raspberry pi.
- Our solution addresses the challenges of urban parking by integrating ultrasonic sensors with Raspberry pi's computational capabilities.
- These sensors detect the presence of vehicle in parking spaces,transmitting the real-time data to the Raspberry pi for processing.
- Through IOT connectivity the system communicates with a server , enabling users to access parking availability information via web application
- Our approach aims to revolutionize urban parking management , promoting efficient space utilization, Reducing congestion and enhancing the overall parking experience.
- Thus Exploring the efficiency of smart parking system by implementing a solution based on Raspberry pi technology.

COMPONENTS:

- Ultrasonic Sensor
- Raspberry pi

Ultrasonic Sensor:

The ultrasonic sensor plays a crucial role in a smart parking system by providing essential data about the occupancy status of parking spaces.

1. **Occupancy Detection:**It detect the presence of vehicles by emitting ultrasonic waves and measuring the time it takes for the waves to bounce back**.**The data is used to determine whether a parking space is occupied or vacant
2. **Real-Time Data:** It continuously monitor the parking spaces,it provide real-time data on the availability of parking spots

3. **Accuracy:** It offers high accuracy in detecting the presence of vehicles making them reliable for determining the actual occupancy status.
4. **Low cost:** Compared to other sensor it is cost-effective,making them a pratical choice for large-scale deployment in parking system.
5. **Easy Installation:** It is relatively easy to install and integrate with minimal setup which contributes the simplicity of parking system
6. **Integration:** It integrate seamlessly with the central processing unit(e.g.,Raspberry pi) and other components of the smart parking system
7. **Low Power Consumption:** It has low power consumption , contributing to energy efficiency in the overall system

By fulfilling these roles , ultrasonic sensor enable smart parking system to accurately monitor and manage parking space occupancy, leading to improve efficiency and enhanced user experience.

Raspberry pi:

In smart parking system, Raspberry pi plays a crucial role by serving as a central processing unit and facilitating various functions

1. **Data Processing**: It processes data from sensors, such as ultrasonic sensors to determine the occupancy status of parking spaces.
2. **Decision Making**: It makes real-time decisions based on the data received, deciding whether parking space is available or occupied.
3. **Communication Hub**: It acts as a communication hub, facilitating the exchange of data between sensors ,actuators and the central server or cloud

4. **Connectivity**: It enables connectivity to the internet, allowing the smart parking system to transmit data to remote servers and receive updates.
5. **Integration**: It integrates with other components of the system such as user interface(website) to provide information to end-users
6. **Automation**: It can be programmed to automate certain processes such as sending alerts to users when a parking space becomes available or managing the data.
7. **Flexibility**: It allows for customization and adaptation to specific requirements,making it suitable for various smart parking implementations.
8. **Scalability**:It offers a scalable platform , making it feasible to expand the smart parking system to cover larger areas or accommodate more sensors.

By performing these functions ,Raspberry pi contributes to the overall efficiency ,responsiveness and intelligence of the smart parking system

WORKING PRINCIPLE:

- ✓Ultrasonic sensors emit sound waves and measure the time it takes for the waves to bounce back after hitting an object.
- ✓Raspberry Pi processes this data to calculate the distance between the sensor and the vehicle, determining if a parking space is occupied or vacant

SYSTEM ARCHITECTURE:

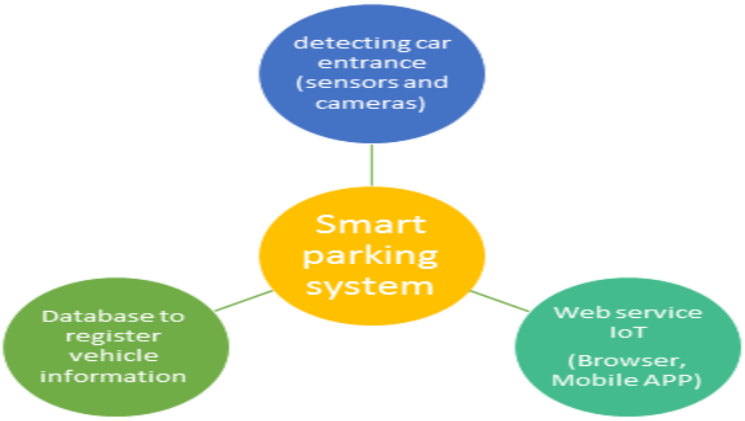
1) **Sensor Network:** Ultrasonic sensors are strategically placed in each parking space ,forming a network
2) **Data Processing:** Raspberry pi processes sensor data in real-time ,categorizing spaces as occupied or available.
3) **User Interface:** The system can have a user interface ,accessible through a mobile app or a display ,showing real-time parking availability

WORKFLOW ARCHITECTURE:

Sensing the data with the help of ultrasonic sensor

Processing the collected data in a Raspberry pi

Displaying it in a real time transit information through a web application

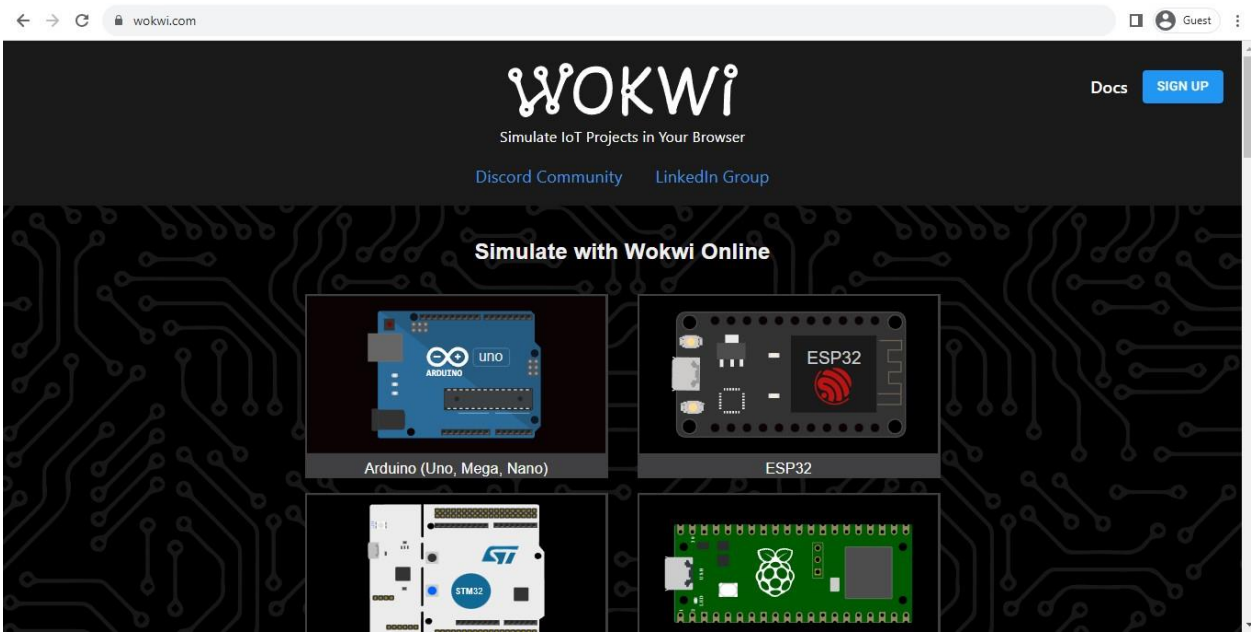


SIMULATION:

Steps For Simulating

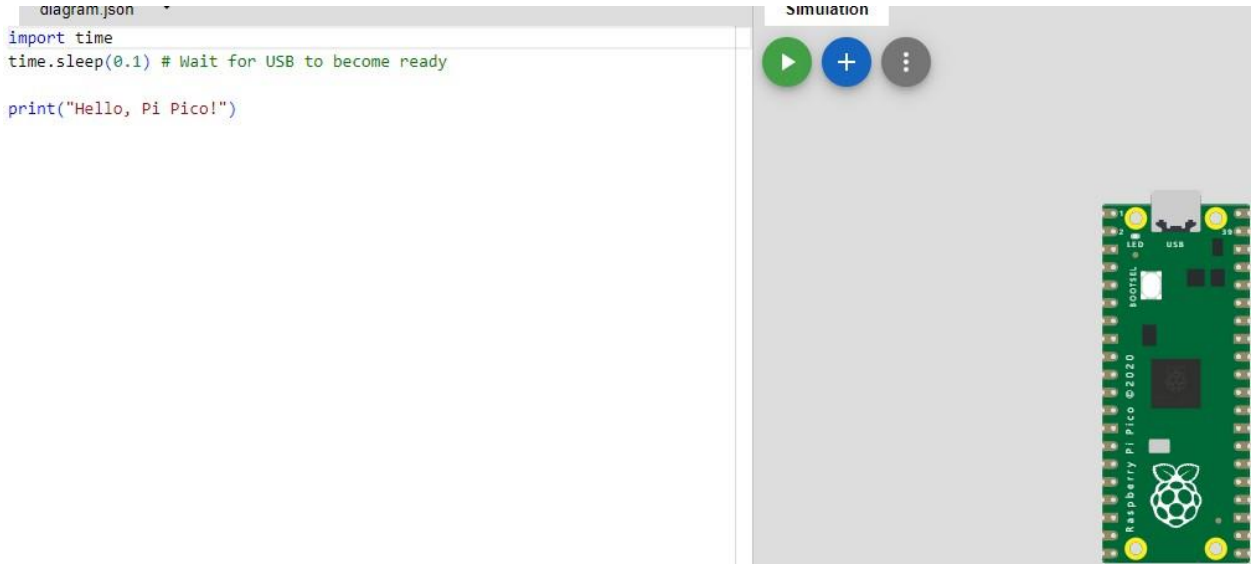
Step 1:Accesswokwi

- Go to the wokwi website and choose raspberry pi project(https://wokwi.com/projects/new/pi-pico)

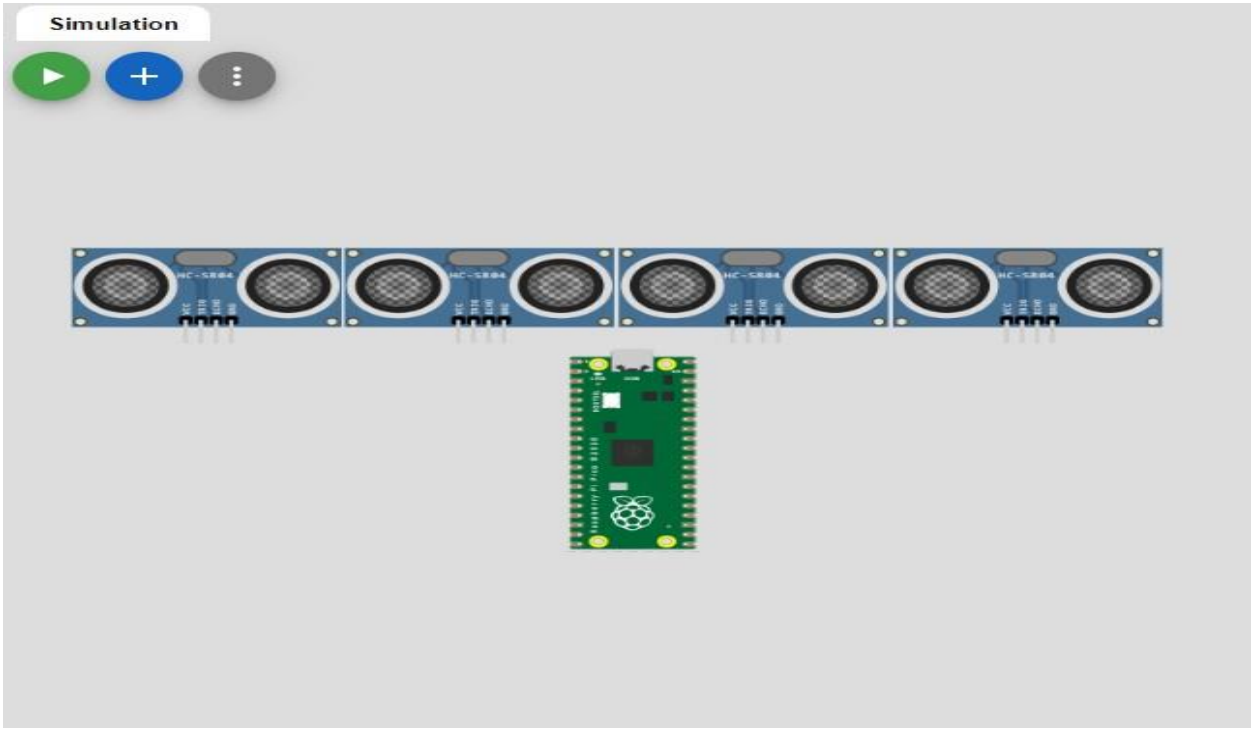


Step 2: Create a New Project

- Click on “create a new project”



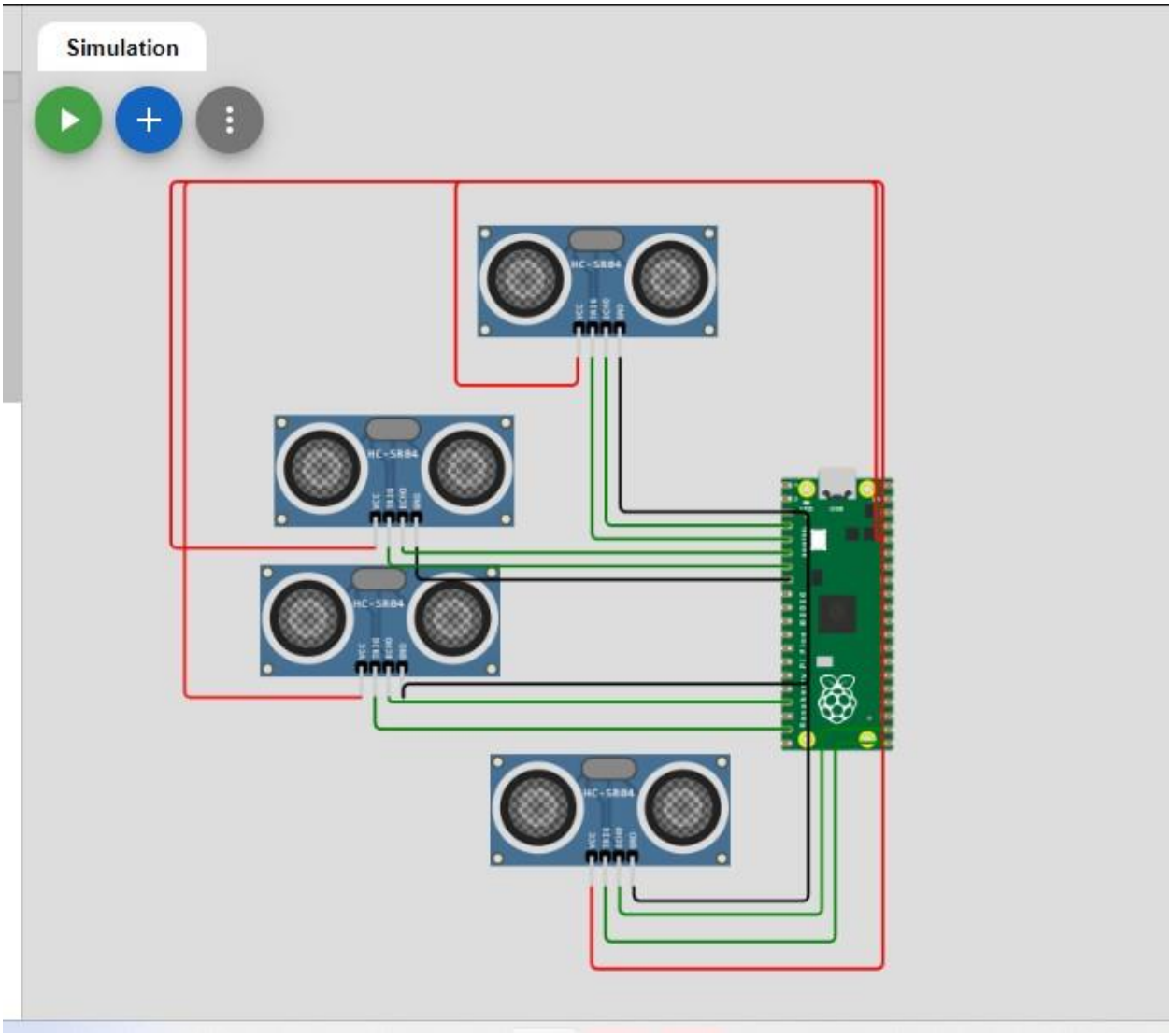
Step 3: Add Ultrasonic sensor



Step 4: Make connections to ultrasonic sensor with a Raspberry pi

- Connect ultrasonic1 VCC pin to Pico 3V3
- Connect ultrasonic1 TRIG pin to Pico GP3
- Connect ultrasonic1 ECHO pin toPico GP2
- Connect ultrasonic1 GND pin to Pico GND1
- Connect ultrasonic2 VCC pin to Pico 3V3
- Connect ultrasonic2TRIG pin to Pico GP14
- Connect ultrasonic2ECHO pin to Pico GP13
- Connect ultrasonic2GND pin to Pico GND1
- Connect ultrasonic3 VCC pin to Pico 3V3
- Connect ultrasonic3TRIG pin to Pico GP5
- Connect ultrasonic3ECHO pin to Pico GP4
- Connect ultrasonic3GND pin to Pico GND2
- Connect ultrasonic4 VCC pin to Pico 3V3
- Connect ultrasonic4TRIG pin to Pico GP16
- Connect ultrasonic4ECHO pin to Pico GP17
- Connect ultrasonic4GND pin to Pico GND1

CIRCUIT DIAGRAM:



Step 5: code

- Click on the “code” tab in wokwi to access the code editor

CODE:

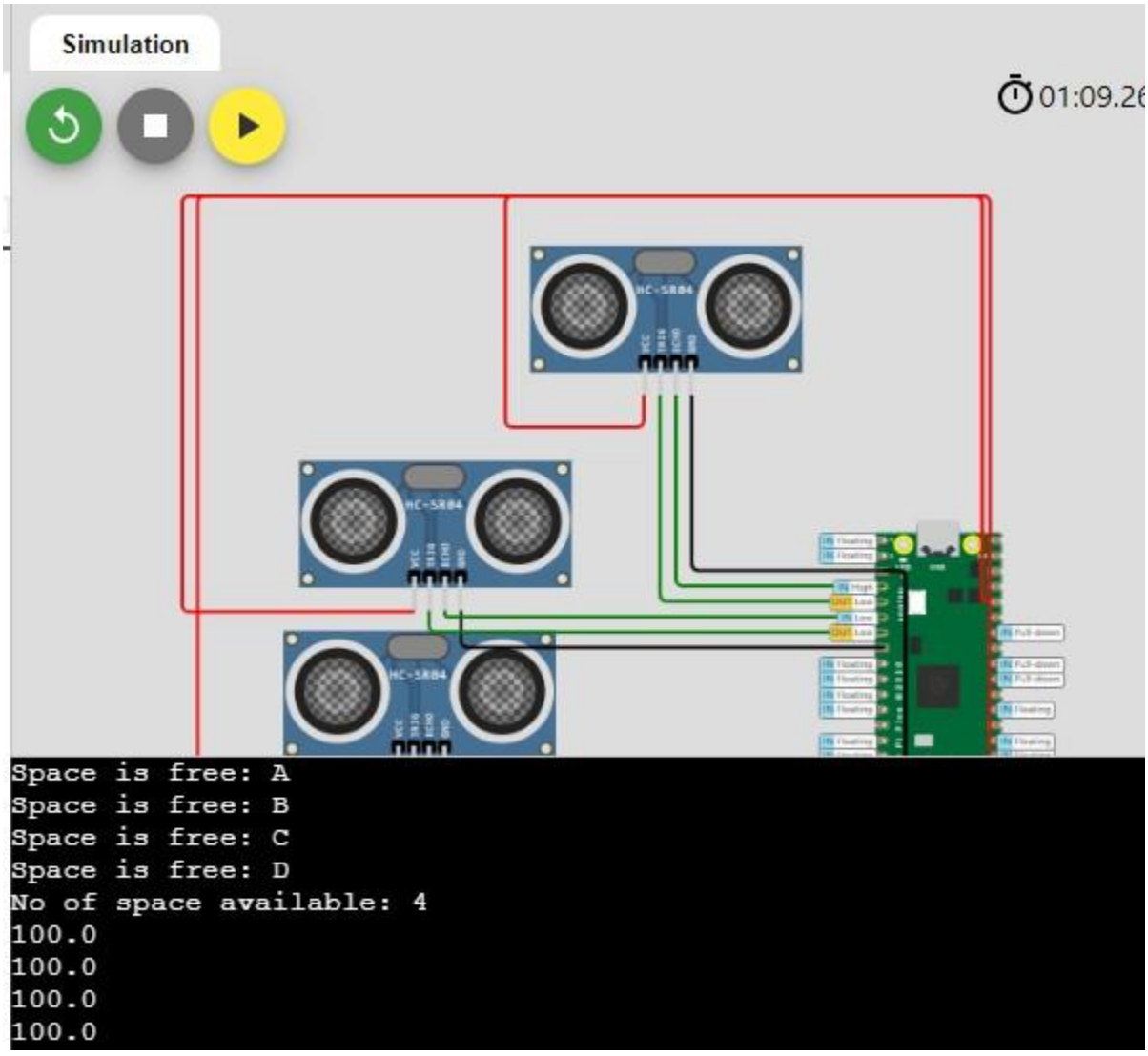
```
from ultra import DistanceSensor
from time import sleep
dsa = DistanceSensor(echo=2, trigger=3)
dsb = DistanceSensor(echo=4, trigger=5)
dsc = DistanceSensor(echo=13, trigger=14)
dsd = DistanceSensor(echo=17, trigger=16)
while True:
distance_a = dsa.distance * 100
distance_b = dsb.distance * 100
distance_c = dsc.distance * 100
distance_d = dsd.distance * 100
a = float(distance_a)
b = float(distance_b)
c = float(distance_c)
d = float(distance_d)  # Convert to a floating-point number
print(a)
print(b)
print(c)
print(d)
A="A"
B="B"
C="C"
D="D"
no=0
def parking(distance, n,slot):
if distance < 30:
```

```
# Code to execute if the distance is less than 30
print("Space is not free:"+slot)
if(n==0):
n=0
else:
n=n-1
else:
# Code to execute if the distance is not less than 30
print("Space is free: "+slot)
if(n==4):
n=4
else:
n=n+1
return n
no=parking(a,no,A)
no=parking(b,no,B)
no=parking(c,no,C)
no=parking(d,no,D)
no=no
print("No of space available:",no)
sleep(0.1)
```

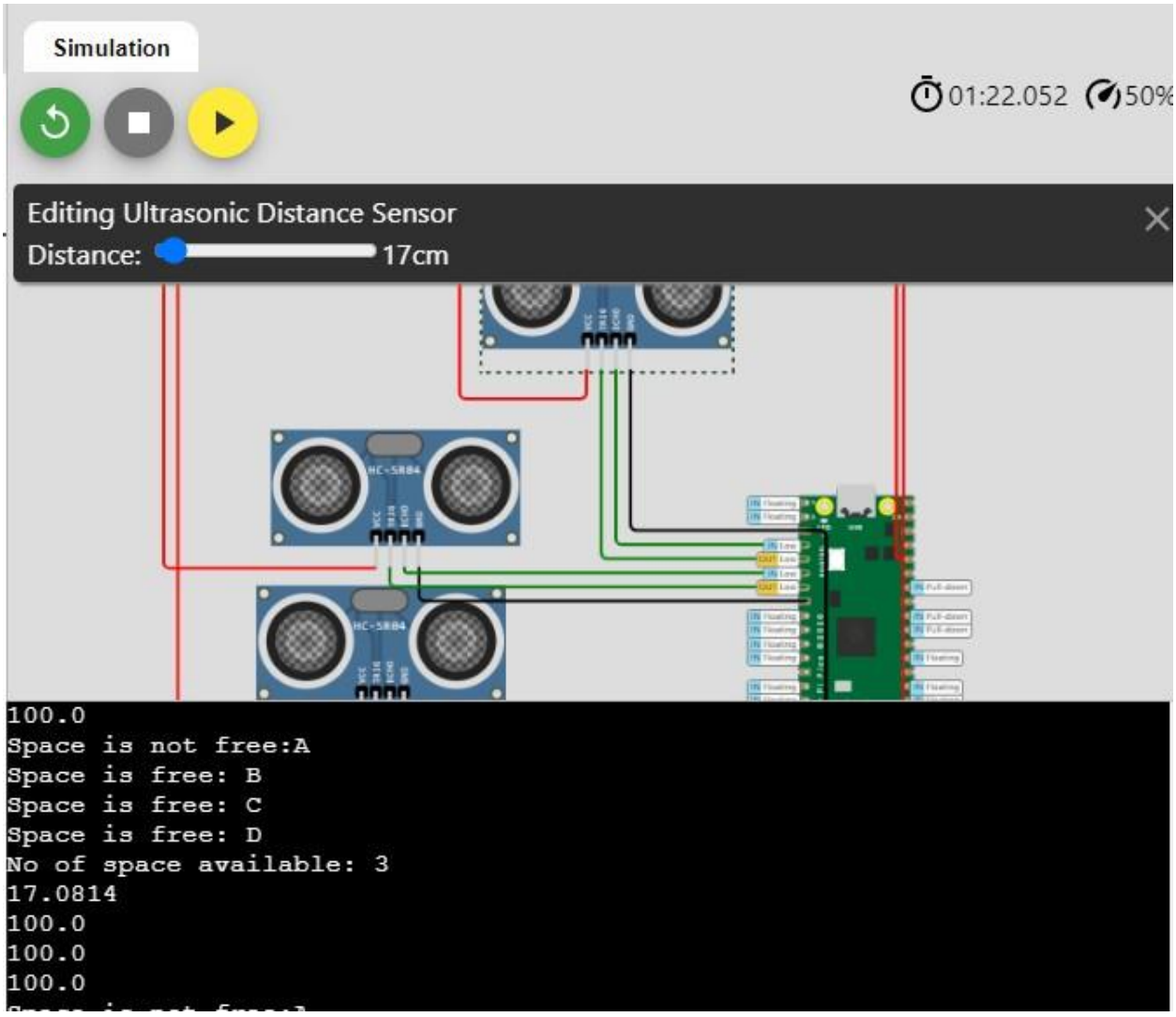
Step 6: Simulation

- Click on the “simulate” button to start the simulation
- Based on the distance echoed by the ultrasonic sensor ,the raspberry pico will show the available parking slot is either free or vacant

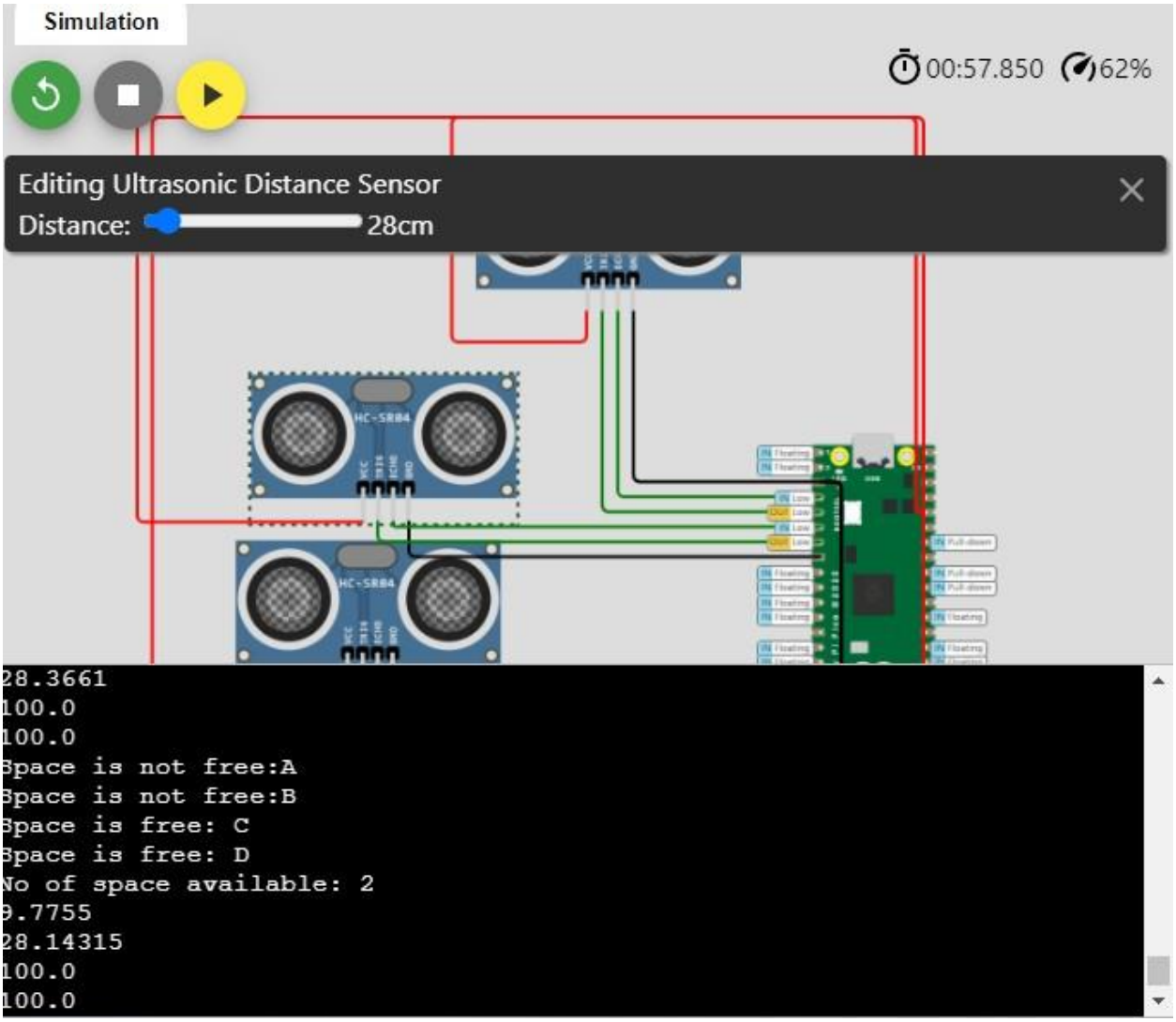
Step 1:when the distance of all the ultrasonic sensor is above 30 cm then it will display all the free spaces along with its echoed distance



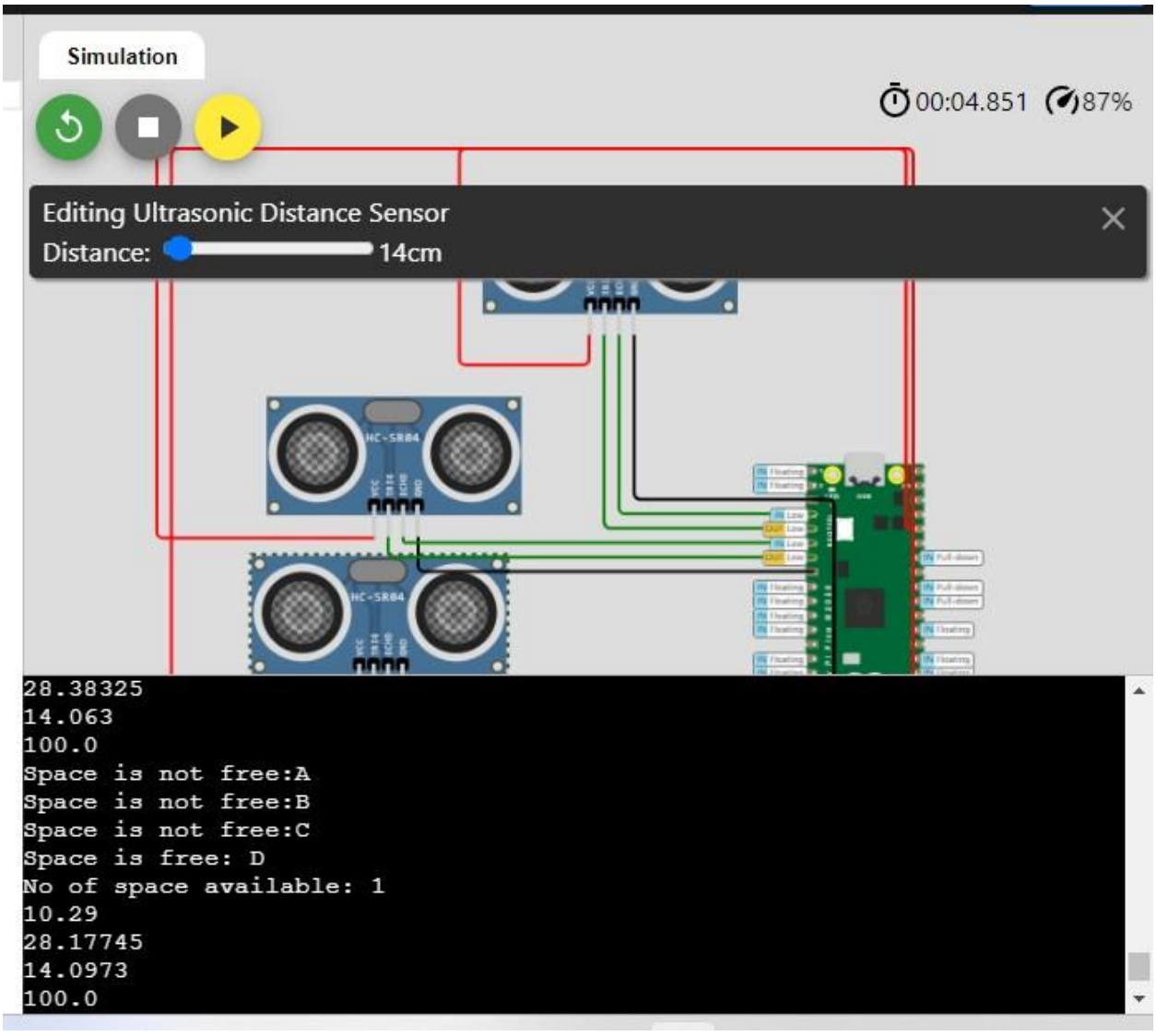
Step 2: when the distance of the ultrasonic1 sensor is below 30 cm (i.e.,17cm) then it will display that the slot A is occupied along with the echoed distance of slot A and it will display the remaining available spaces as free



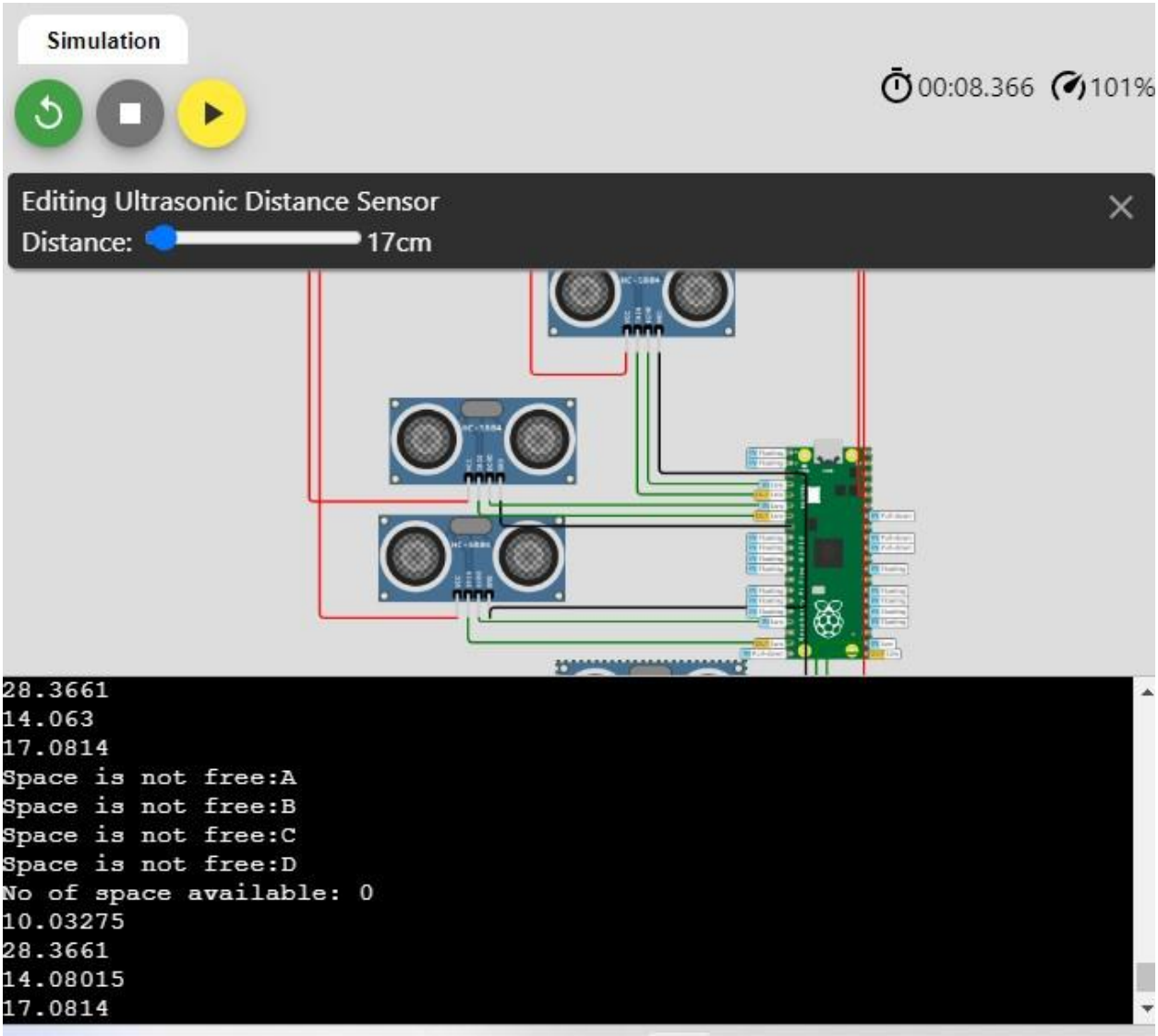
Step 3:when the distance of the ultrasonic1 and ultrasonic2 sensor is below 30 cm (i.e.,17cm and 28cm) then it will display that the slot A and B is occupied along with the echoed distance of slot A and B and it will display the remaining available spaces as free



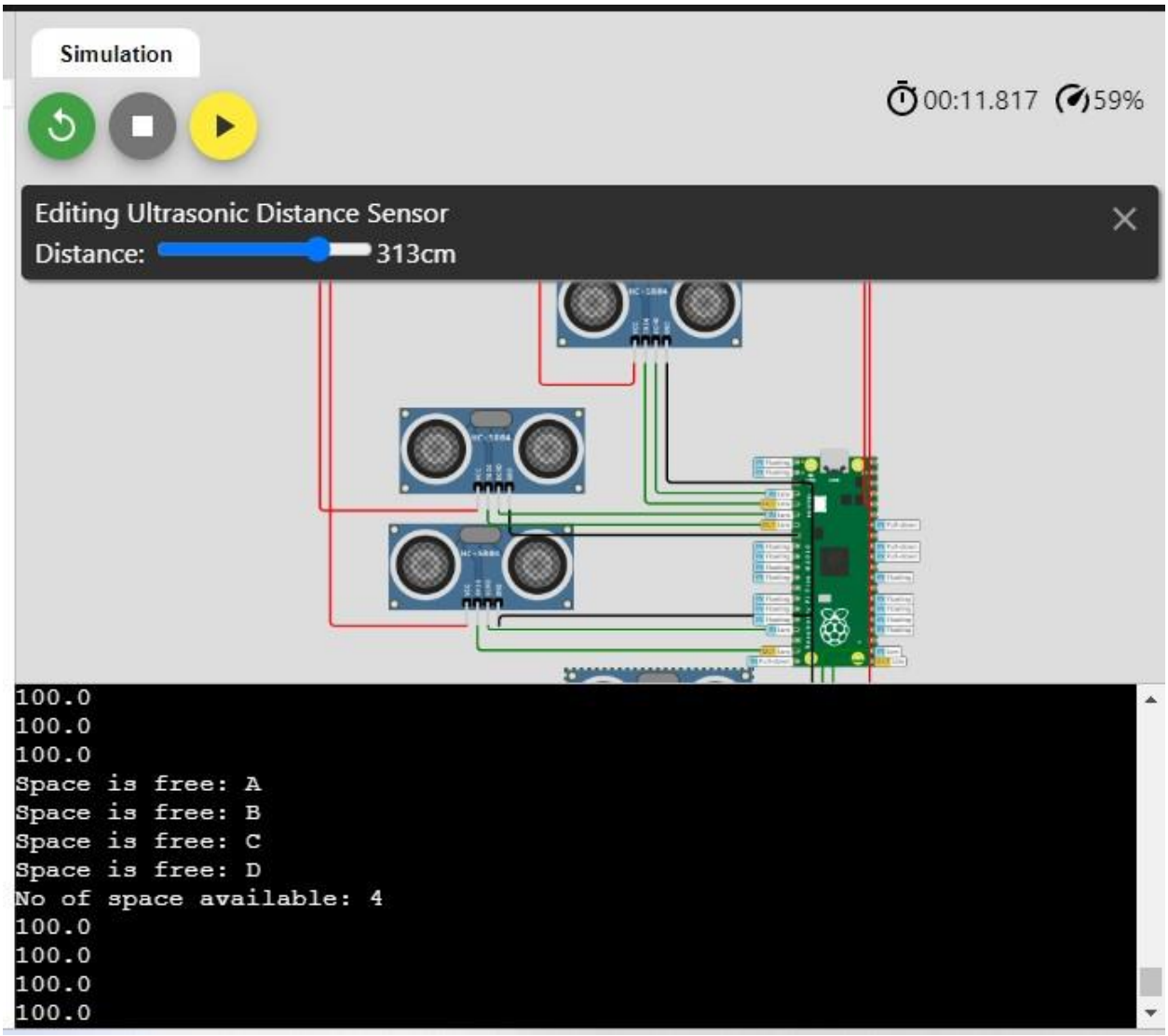
Step 4:when the distance of the ultrasonic1 ,ultrasonic2 and ultrasonic3 sensor is below 30 cm (i.e.,17cm,28cm and 14cm) then it will display that the slot A , B and C is occupied along with the echoed distance of slot A , B and C and it will display the remaining available spaces as free



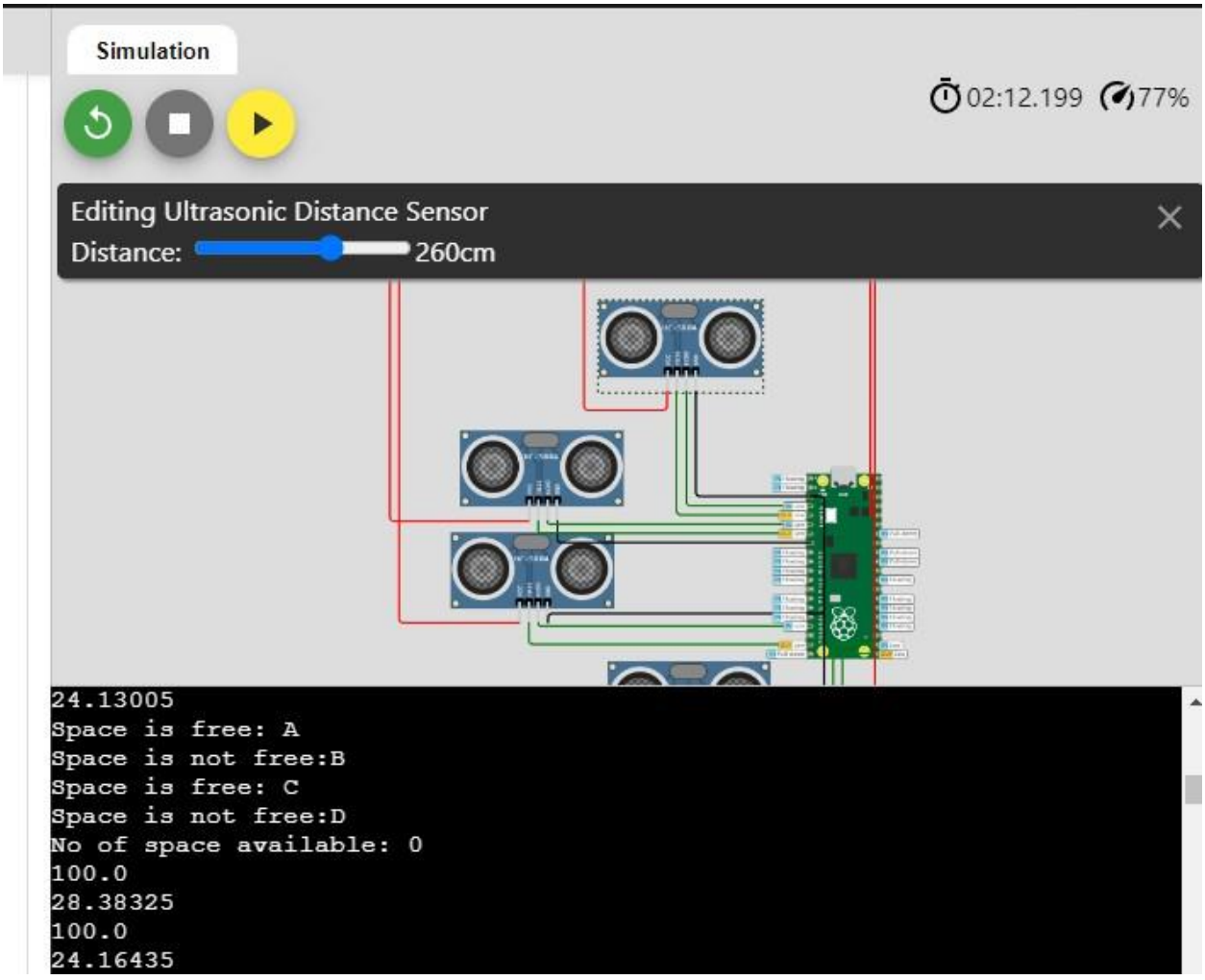
Step 5:when the distance of the ultrasonic1 ,ultrasonic2 ,ultrasonic3 and ultrasonic4 sensor is below 30 cm (i.e.,17cm,28cm ,14cm and 2cm) then it will display that the slot A , B , C and D is occupied along with the echoed distance of slot A , B , C and D and it will display there is no available spaces. All slots are occupied.



Step 6:when the distance of the ultrasonic1 ,ultrasonic2 ,ultrasonic3 and ultrasonic4 sensor is above 30 cm (i.e.,157cm,321cm,245cm and 298cm) then it will display that the slot A , B , C and D is free along with the echoed distance of slot A , B , C and D and it will display there is available spaces. All slots are free.



Step 7:when the distance of the ultrasonic1 andultrasonic3 sensor is above 30 cm (i.e.,157cm,321cm) then it will display that the slot A and C is free along with the echoed distance of slot A and C.when the distance of ultrasonic2 and ultrasonic4 sensor is below 30cm (i.e.,14cm and 12cm)then it will display that the slot B and D is Occupied along with the echoed distance of Slot B and D.



WEB APPLICATION FOR SMART PARKING

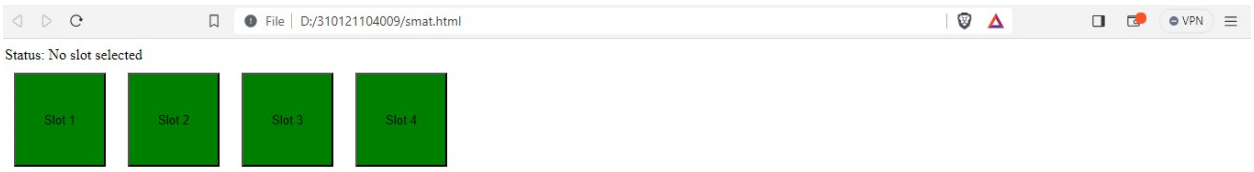
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
.parking-slot {
width: 100px;
height: 100px;
margin: 10px;
cursor: pointer;
}
.empty {
font-size:72;
background-color: green;
}
.full {
```

```
font-size:72;
background-color: red;
}
body
{
font-size:72;
font-family:Times New Roman;
}
</style>
<title>Smart Parking System</title>
</head>
<body>
<div id="status-message">Status: No slot selected</div>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 1</button>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 2</button>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 3</button>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 4</button>
```

```
<script>
function toggleSlot(slot) {
if (slot.classList.contains('empty')) {
slot.classList.remove('empty');
slot.classList.add('full');
document.getElementById('status-message').innerText =
`Status:  ${slot.innerText} is full`;
} else {
slot.classList.remove('full');
slot.classList.add('empty');
document.getElementById('status-message').innerText =
`Status:  ${slot.innerText} is empty`;
}
}
</script>
</body>
</html>
```

Link:///D:/310121104009/smat.html

Step 1: Initially no slots are selected all the slots are empty. So it shows the status No slot Selected

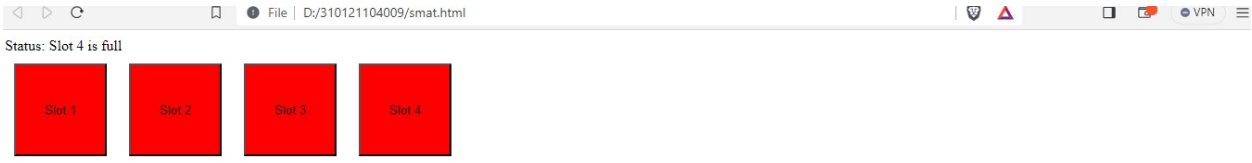


Step 2: when slot 1 is clicked it will show the status slot 1 is full which means the slot is occupied and indicates in red colour

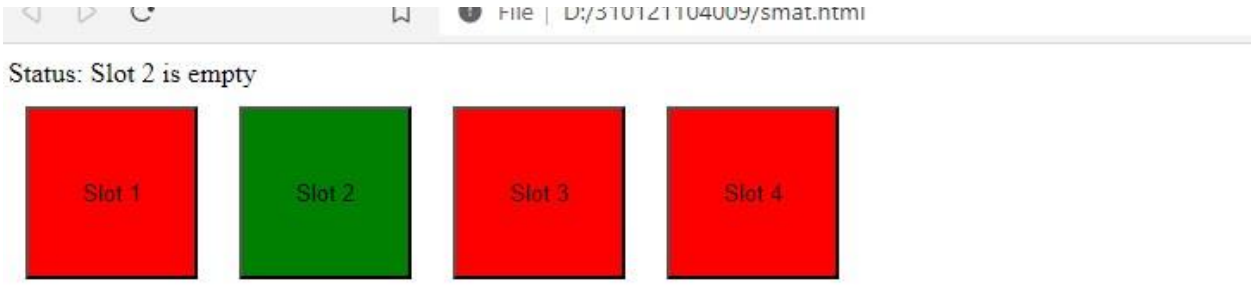
Step 3: When slot 2 is clicked it will show the status slot 2 is full which means the slot is occupied and indicates in red colour and other slots are free

Step 4: When slot 3 is clicked it will show the status slot 3 is full which means the slot is occupied and indicates in red colour and other slots are free

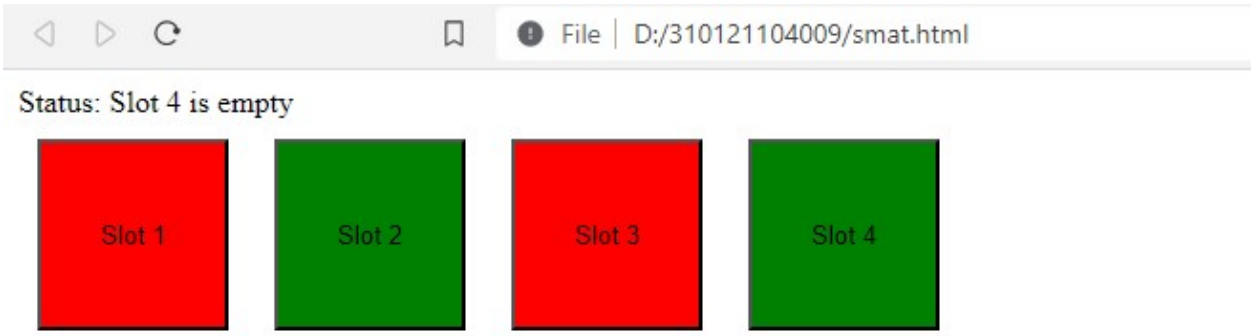
Step 5: When slot 4 is clicked it will show the status slot 4is full which means the slot is occupied and indicates in red colour and there is no more slots left .All are occupied



Step 6: When slot 2 is clicked it will show the status slot 2 is emptywhich means the slot is free and indicates in green colourwhereas the other slots are occupied.



Step 7: When slot 4 is clicked it will show the status slot 4 is empty which means the slot is free and indicates in green colour whereas the other slots are occupied.



ADVANTAGES:

1. **Efficiency:** Drivers can quickly locate vacant parking spaces ,reducing congestion and saving time
2. **Cost-Effective**: Raspberry Pi and Ultrasonic sensors are affordable compared to traditional parking management systems
3. **Real-Time Monitoring**:Provides up-to-the minute information on parking space availability

FUTURE ENHANCEMENTS:

- The future scope of smart parking system holds significant potential for further advancements and widespread adoption
  - ✓ Integration with Autonomous vehicles
  - ✓ AI and Machine learning
  - ✓ Dynamic pricing Models
  - ✓ Augmented Reality
  - ✓ 5G Connectivity
  - ✓ Block chain for security

CONCLUSION:

- In conclusion, smart parking system using Raspberry Pi and Ultrasonic sensors offers an intelligent and cost-effective solution to modernize parking management ,making the process more convenient
- It is able to better track the availability of parking spots on a given lot ,making it easier to find an available parking slot.