SMART PARKING

Problem statement

Free parking space is hard to find for public It increases traffic and more usage of fuel. It also gives bad experience for drivers and time consuming. In this project,sensors are used to find free spaces and this information is given to users .We can also help users book parking space in advance using mobile.

Design thinking

Project objective:

To help public find free parking space with help of sensors and avoid stress,traffic and time consumption.

Iot sensors:

1) ultrasonic sensor: It is used to detect whether the parking slot is free or not. A ultrasonic sensor is placed on parking slot and it emits high frequency pulse from trigger pin . In the presence of any vehicles ,the ultrasonic pulse is bounced back and detected using echo pin. Thus we can find out occupied spaces.distance=time*speed/2 formula it used to calculate distance of parked vehicle and sensor.

2)Servo motor:It can be used to control the entry and exit of vehicles. Real-Time Transit Information Platform: An application which helps drivers to find free and occupied parking space,with the help of data collected from sensors . In case of paid parking zone,it helps drivers to book free parking space in advance . Duration of parking can be calculated and drivers can chargers accordingly.

Integration Approach:

Ultrasonic sensor is connected with raspberry pi and the information coming from ultrasonic sensor are stored in it.Then , raspberry pi is connected with the mobile app using Bluetooth or network.

Micro controllers used:

1)ESP32: It detects information from the sensors like ultrasonic sensor and concludes whether parking slot is free or not. It can be connected with cloud to store data. It used wifi or Bluetooth for data transmission.

2) Arduino Uno: It is used to detect the presence of vehicles in parking slot.It can communicate information to a central controller or a web server.

Connectivity:

The Arduino Uno and esp32 are connected with the central using wifi and Bluetooth.

Cloud:

Beeceptor: It provides cloud service to users. It is used to store and intercept data.

It also used to modify and filter our data. Beeceptor can be used to simulate Http request and responses .

Protocols:

1)MQTT-Message queuing telemetry.It is used for communication between sensors used in parking slot and central server .

2)HTTP-Hyper text transfer protocol. It is used in while we are using mobile apps to present information regarding free parking slots.

3)Zigbee:This protocol is used for connecting sensors .

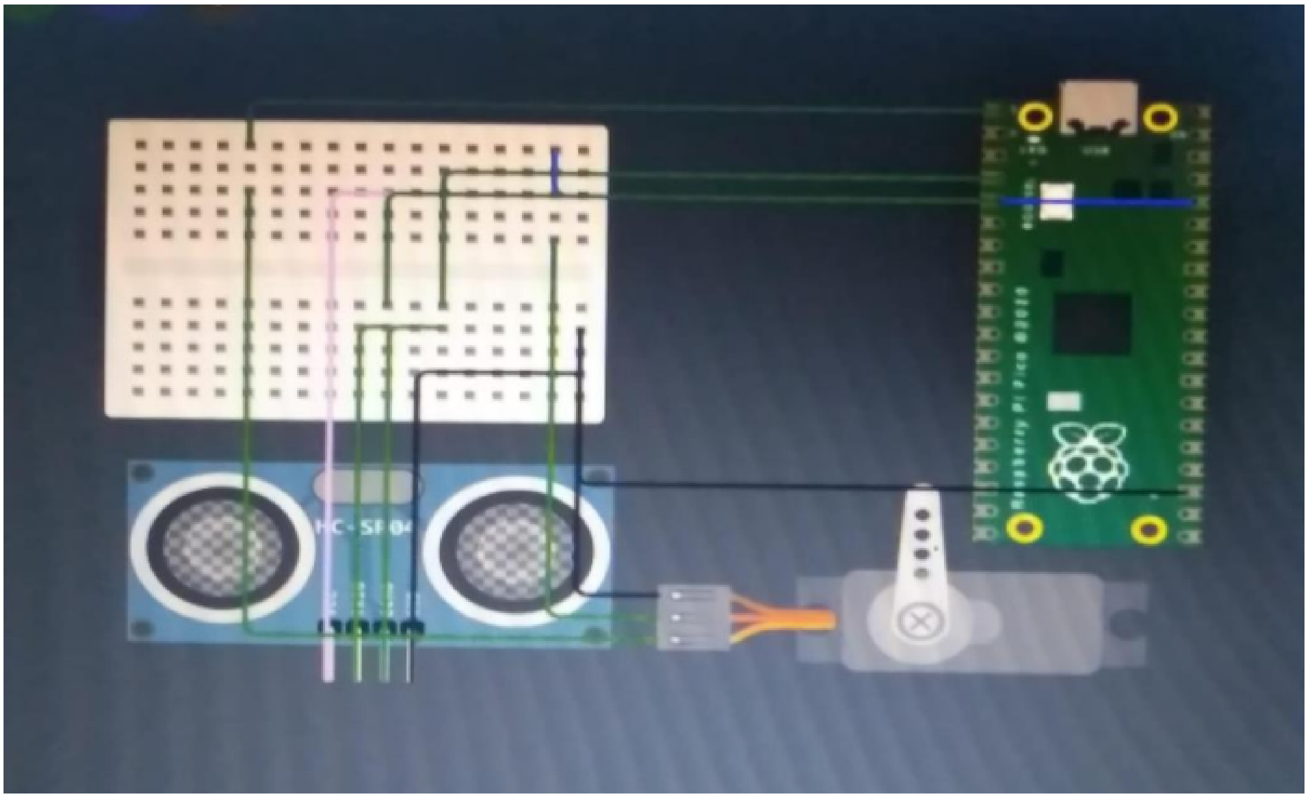
4)Bluetooth(BLE):This protocol is used for connecting mobile phones and parking sensors to provide information about free parking space.

Program:

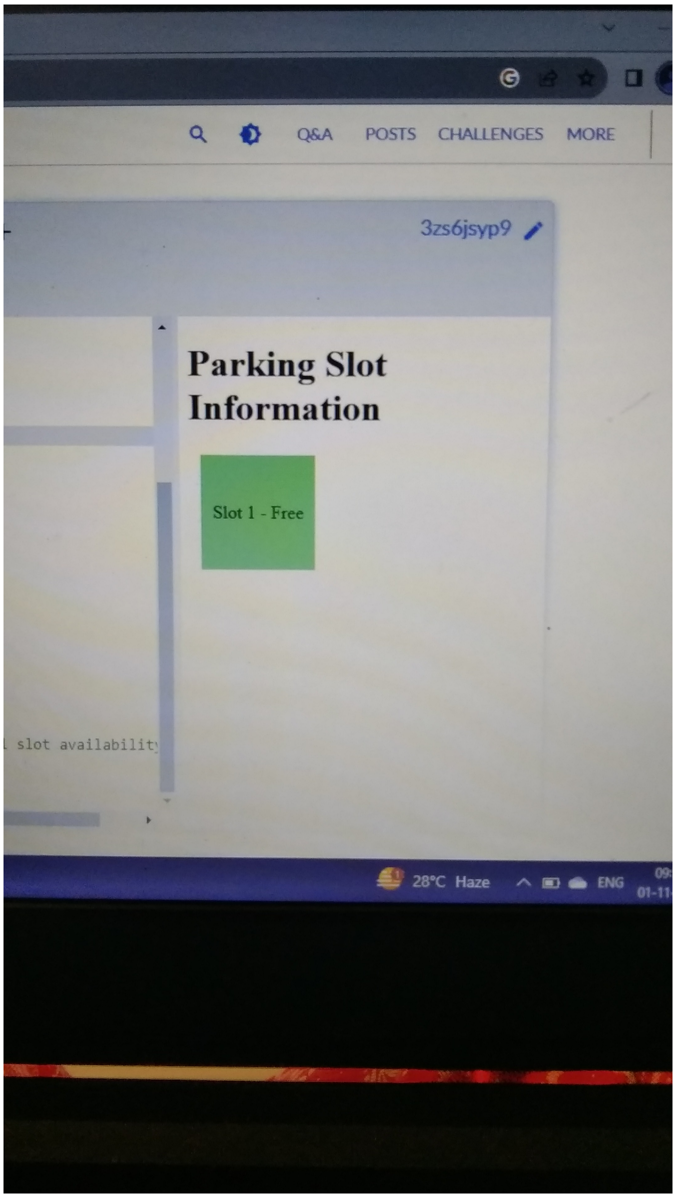
```
from machine import Pin

import utime

from time import sleep

from machine import PWM

pwm = PWM(Pin(0))

pwm.freq(50)

trigger = Pin(3, Pin.OUT)

echo = Pin(2, Pin.IN)
```

```
def setServoCycle (position):
 pwm.duty_u16(position)
 sleep(0.01)
def ultra():
 trigger.low()
 utime.sleep_us(2)
 trigger.high()
 utime.sleep_us(5)
 trigger.low()
 while echo.value() == 0:
 signaloff = utime.ticks_us()
 while echo.value() == 1:
 signalon = utime.ticks_us()
 timepassed = signalon - signaloff
 distance = (timepassed * 0.0343) / 2
 if(distance<100):
 print("THE SLOT IS OCCUPIED")
 else:
 print("THE SLOT IS free")
print("please enter")
for pos in range(1000,9000,50):
 setServoCycle(pos)
 for pos in range(9000,1000,-50):
 setServoCycle(pos)
```

```
while True:

 ultra()

 utime.sleep(1)

beeceptor_url = "https://https://iotproject.free.beeceptor.com"

try:

 while True:

 print(f"Distance: {distance} cm")

 data = {"distance": distance}

 response = requests.post(beeceptor_url, json=data)

 if response.status_code == 200:

 print("Data sent to Beeceptor successfully")

 else:

print(f"Failed to send data. Status code: {response.status_code}")

 time.sleep(1)
```



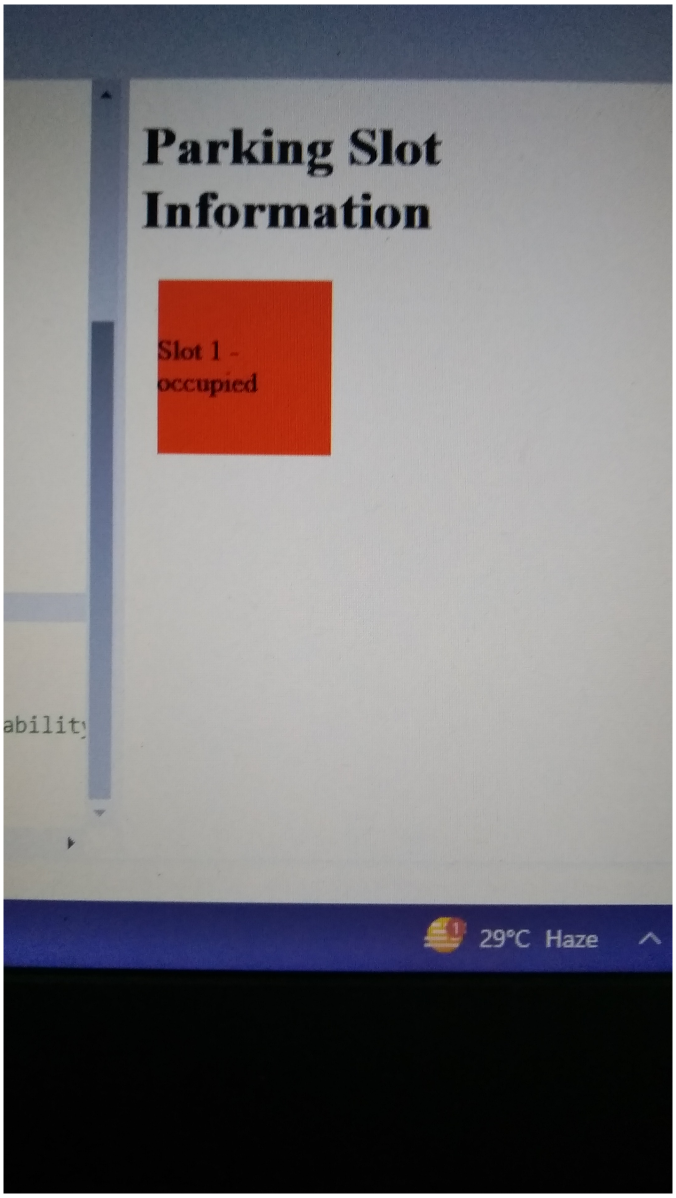
```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <meta name="viewport" content="width=device-width, initial-scale=1.0">

 <title>Parking Slot Information</title>

 <style>

 .slot {

 width: 100px;

 height: 100px;

 margin: 10px;

 display: flex;
 justify-content: center;

 align-items: center;

 }

 .free-slot {

 background-color: green;

 }

 .occupied-slot {

 background-color: red;

 }

 </style>

</head>

<body>

 <h1>Parking Slot Information</h1>


 <div id="parking-slots"></div>

 <script>


 fetch('https://https://iotproject.free.beeceptor.com')

 .then(response => response.json())

 .then(data => {

 const parkingSlots = document.getElementById('parking-slots');

 data.forEach(slot => {

 const slotDiv = document.createElement('div');

 slotDiv.className = `slot ${slot.isFree ? 'free-slot' : 'occupied-slot'}`;
 slotDiv.textContent = `Slot ${slot.slotNumber} - ${slot.isFree ? 'Free' : 'Occupied'}`;
 parkingSlots.appendChild(slotDiv);
 });
 })
 .catch(error => {
 console.error('Error fetching data:', error);
 });
</script>
</body>
</html>
```



Parking Slot Information

Slot 1 - Free

```
l slot availabilit
```



Parking Slot Information

Slot 1 - occupied

ability

29°C Haze