Building an IoT Parking Space Occupancy Detection System with Raspberry Pi:

Components Required:

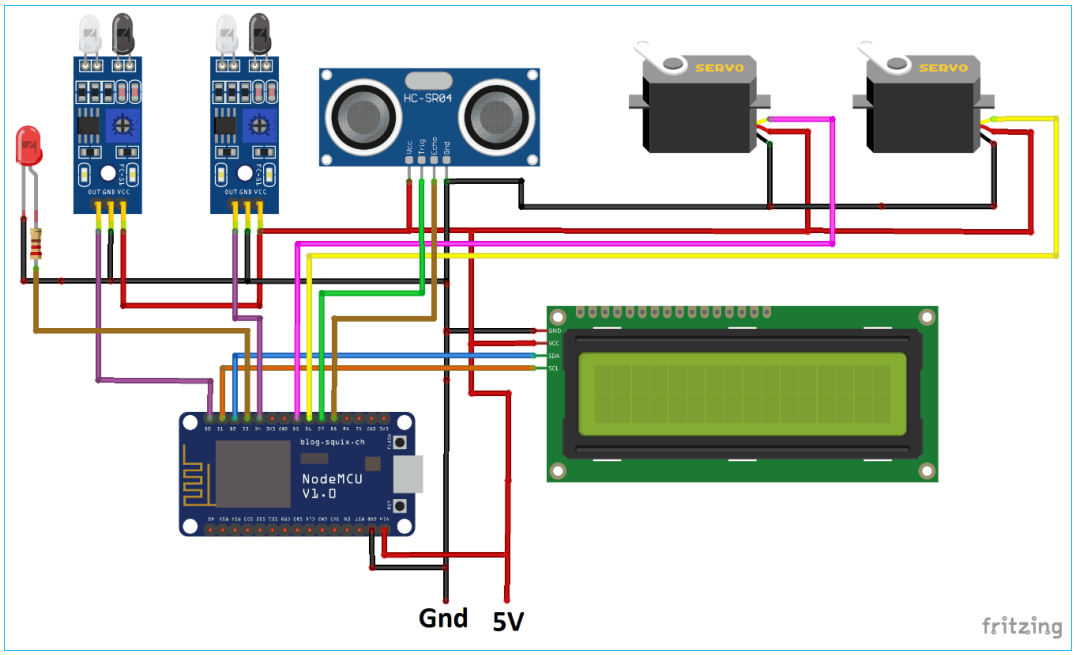
- Raspberry Pi (with Wi-Fi capabilities)

- Ultrasonic Sensors

- Jumper wires

- Cloud or Mobile App Server

Step 1: Circuit Diagram :



Step 2: Hardware Setup:

- Connect the ultrasonic sensors to the Raspberry Pi's GPIO pins as per the circuit diagram.

Step 3: Python Code (Sample Script):

Here's a sample Python script for collecting data from ultrasonic sensors and sending it to a hypothetical cloud server:

Python:

```
import RPi.GPIO as GPIO
import time
import requests

# Ultrasonic Sensor GPIO pins
TRIG_PIN = 23
ECHO_PIN = 24

# Cloud Server Endpoint (add your url)
SERVER_URL = "http://your-server-url.com/endpoint"

# Set up GPIO mode
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)

def get_distance():
    # Send a trigger pulse
    GPIO.output(TRIG_PIN, True)
    time.sleep(0.00001)
    GPIO.output(TRIG_PIN, False

    start_time = time.time()
    end_time = time.time()

    # Wait for the echo to return
    while GPIO.input(ECHO_PIN) == 0:
        start_time = time.time()
```

```
    while GPIO.input(ECHO_PIN) == 1:
        end_time = time.time()

    # Calculate distance
    duration = end_time - start_time
    distance = (duration * 34300) / 2  # Speed of sound is 343 m/s

    return distance

try:
    while True:
        distance = get_distance()

        # Define a threshold for parking space occupancy
        threshold = 10  # Adjust based on your setup

        if distance < threshold:
            # Parking space is occupied
            occupancy_status = "Occupied"
        else:
            # Parking space is vacant
            occupancy_status = "Vacant"

        # Send data to the server
        data = {"occupancy": occupancy_status}
        response = requests.post(SERVER_URL, json=data)
        time.sleep(5)  # Adjust the interval as needed
except KeyboardInterrupt:
GPIO.cleanup()
```

Step 4: Explanation:

This IoT system uses ultrasonic sensors connected to a Raspberry Pi to detect parking space occupancy and sends the data to a server for further processing. Here's how it works:

- Hardware Setup: Connect ultrasonic sensors to Raspberry Pi's GPIO pins as per the provided circuit diagram (not included).

- Python Script: The Python script uses RPi.GPIO to interact with the sensors, measures distance, and determines parking space occupancy based on a predefined threshold.

- Data Transmission: The script sends occupancy status data to a cloud or mobile app server using HTTP POST requests.

- Continuous Monitoring: The script runs continuously, periodically checking occupancy status and sending updates to the server.

By integrating these components and fine-tuning the script to match your hardware setup, you can create an IoT Parking Space Occupancy Detection System using a Raspberry Pi.