

Team Members:

1. R.RESHMA MASHUTHA - 921021104040
2. C.BHARATHI - 921021104003
3. B.KIRUTHIKA - 921021104019
4. P.RAVINA - 921021104038

Smart parking using IoT

Smart parking:

- Smart parking is a system that uses the Internet of Things (IoT) to collect and analyze data about parking availability and occupancy. This data can then be used to improve the efficiency and convenience of parking for both drivers and parking lot operators.
- One of the key benefits of smart parking is that it can help drivers to find parking spaces more quickly and easily. This is done by providing drivers with real-time information about parking availability in nearby parking lots. Drivers can access this information through a mobile app or website, or through signs that are posted in the parking lot itself.

To build an IoT sensor system and Raspberry Pi integration for smart parking, we will need the following components:

- Raspberry Pi
- Ultrasonic sensors
- Jumper wires
- Breadboard
- Power supply
- Cloud or mobile app server

Configure ultrasonic sensors:

- Ultrasonic sensors work by emitting high-frequency sound waves and measuring the time it takes for the waves to return to the sensor after hitting an object. The distance to the object is then calculated based on the speed of sound.
- To configure ultrasonic sensors for parking space occupancy detection, we will need to mount them in a suitable location, such as on the ceiling or walls of the parking garage. The sensors should be positioned so that they have a clear view of the parking spaces.
- Once the sensors are mounted, we will need to connect them to the Raspberry Pi. The ultrasonic sensors typically have three wires: VCC, GND, and ECHO. Connect the VCC wire to a 5V power pin on the Raspberry Pi, the GND wire to a ground pin, and the ECHO wire to a GPIO pin.

Python scripts:

Once the ultrasonic sensors are connected to the Raspberry Pi, we can write Python scripts to collect data from the sensors and send it to the cloud or mobile app server.

The following Python script shows how to read the distance from an ultrasonic sensor:

```
import RPi.GPIO as GPIO

# Set the GPIO pin for the ultrasonic sensor
ECHO_PIN = 18

# Set the speed of sound
SPEED_OF_SOUND = 343

# Initialize GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(ECHO_PIN, GPIO.IN)
```

```
# Send a pulse to the ultrasonic sensor
GPIO.output(ECHO_PIN, GPIO.HIGH)
time.sleep(0.00001)
GPIO.output(ECHO_PIN, GPIO.LOW)

# Start the timer
start_time = time.time()

# Wait for the pulse to return
while GPIO.input(ECHO_PIN) == GPIO.LOW:
    pass

# Stop the timer
end_time = time.time()

# Calculate the distance
distance = (end_time - start_time) * SPEED_OF_SOUND / 2

# Print the distance
print(distance)
```

We can use this script as a starting point to write a Python script that collects data from all of the ultrasonic sensors in our parking garage and sends it to the cloud or mobile app server.

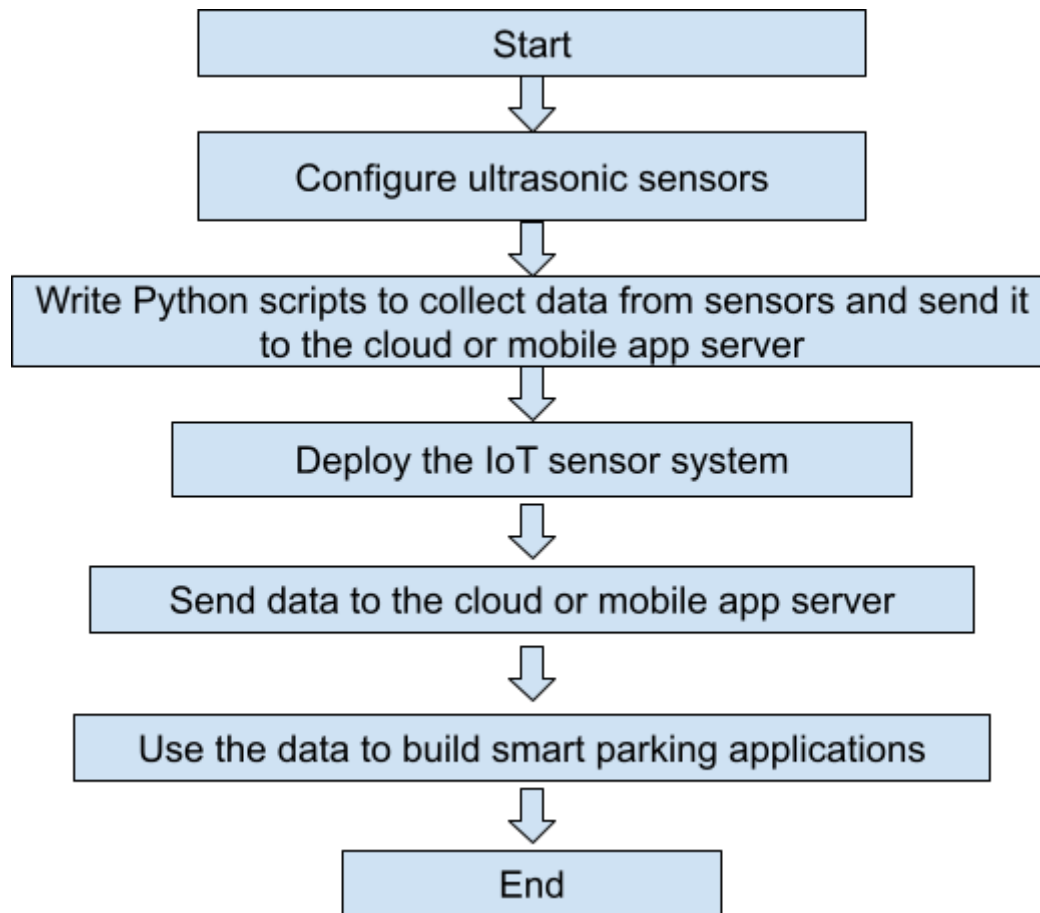
Send data to the cloud or mobile app server:

Once we written a Python script to collect data from the ultrasonic sensors, we can send the data to the cloud or mobile app server using a variety of methods, such as:

- MQTT
- HTTP
- WebSockets

The specific method we will use depends on the cloud or mobile app server that we are using.

Flowchart:



Conclusion:

Once we configured the ultrasonic sensors and wrote a Python script to collect data from the sensors and send it to the cloud or mobile app server, we will have a complete IoT sensor system for smart parking.

We can use the data from the cloud or mobile app server to build a variety of applications, such as:

- A mobile app that shows drivers where there are available parking spaces
- A website that shows the occupancy of parking garages in real time
- A system that alerts parking garage operators when there are no parking spaces available

In addition to the above, we can also use the data from the IoT sensor system to develop other smart parking applications, such as:

- A system that optimizes traffic flow in parking garages
- A system that provides personalized parking recommendations to drivers
- A system that enables drivers to reserve parking spaces in advance

The possibilities are endless!