Team Members:

1. R.RESHMA MASHUTHA - 921021104040

2. C.BHARATHI - 921021104003

3. B.KIRUTHIKA - 921021104019

4. P.RAVINA - 921021104038

Smart parking using IoT

Objectives:

● Smart parking is a system that uses the Internet of Things (IoT) to collect and analyse 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.

IoT Sensor Setup :

- The Smart Parking system uses ultrasonic sensors to detect the presence of vehicles in parking spaces. The sensors are placed at the entrance and exit of each parking space. When a vehicle enters a space, the sensor will send a signal to the Raspberry Pi. The Raspberry Pi will then update the database to indicate that the space is occupied. When the vehicle leaves the space, the sensor will send another signal to the Raspberry Pi, and the database will be updated to indicate that the space is available again.

Mobile App Development :

- The Smart Parking mobile app is developed using the Flutter framework. The app displays a map of nearby parking lots with the occupancy status of each space indicated. Drivers can use the app to filter the results by distance, price, and other criteria. The app also allows drivers to reserve parking spaces in advance.

Raspberry Pi Integration :

The Raspberry Pi is integrated into the Smart Parking project in two ways:

1. The Raspberry Pi is used to collect and process data from the ultrasonic sensors.
2. The Raspberry Pi is used to send the data from the sensors to the cloud or mobile app server.

➢ To collect data from the ultrasonic sensors, the Raspberry Pi uses a GPIO library. The GPIO library allows the Raspberry Pi to read and write to the GPIO pins on the board. The ultrasonic sensors are connected to the GPIO pins on the Raspberry Pi, so the Raspberry Pi can read the distance from the sensors by reading the voltage on the GPIO pins.

➢ To send the data from the sensors to the cloud or mobile app server, the Raspberry Pi uses a networking library. The networking library allows the Raspberry Pi to connect to the internet and send and receive data. The Raspberry Pi can be configured to periodically send the data from the sensors to the cloud or mobile app server.

Code Implementation :

The Smart Parking project is implemented using the following programming languages and technologies:

- **Python:** The Raspberry Pi code is written in Python. Python is a popular programming language that is easy to learn and use.
- **Flutter:** The mobile app is developed using Flutter. Flutter is a cross-platform mobile development framework that allows developers to build native apps for Android and iOS devices using a single codebase.
- **Firebase:** Firebase is a mobile platform from Google that provides a variety of services, such as databases, authentication, and real-time data synchronisation. The Smart Parking mobile app uses Firebase to store and manage the parking availability data.

Example Code:

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

# Set the GPIO pins for the ultrasonic sensors
ECHO_PINS = [18, 23, 24]

# Set the speed of sound
SPEED_OF_SOUND = 343

# Initialize GPIO
GPIO.setmode(GPIO.BCM)
for pin in ECHO_PINS:
    GPIO.setup(pin, GPIO.IN)

# Send a pulse to the ultrasonic sensors
for pin in ECHO_PINS:
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(pin, GPIO.LOW)

# Start the timer
start_times = [time.time() for pin in ECHO_PINS]

# Wait for the pulse to return
distances = []
for pin in ECHO_PINS:
    while GPIO.input(pin) == GPIO.LOW:
        pass

    # Stop the timer
    end_time = time.time()

    # Calculate the distance
    distance = (end_time - start_times[ECHO_PINS.index(pin)]) *
SPEED_OF_SOUND / 2
    distances.append(distance)
```

```
# Send the data to the cloud or mobile app server
url = "https://example.com/api/parking-availability"
data = {"distances": distances}
response = requests.post(url, json=data)

# Check the response status code
if response.status_code == 200:
    # The data was successfully sent to the cloud or mobile app server
    print("Data sent successfully")
else:
    # There was an error sending the data
    print("Error sending data")
```

The following diagram shows the overall architecture of the Smart Parking system:

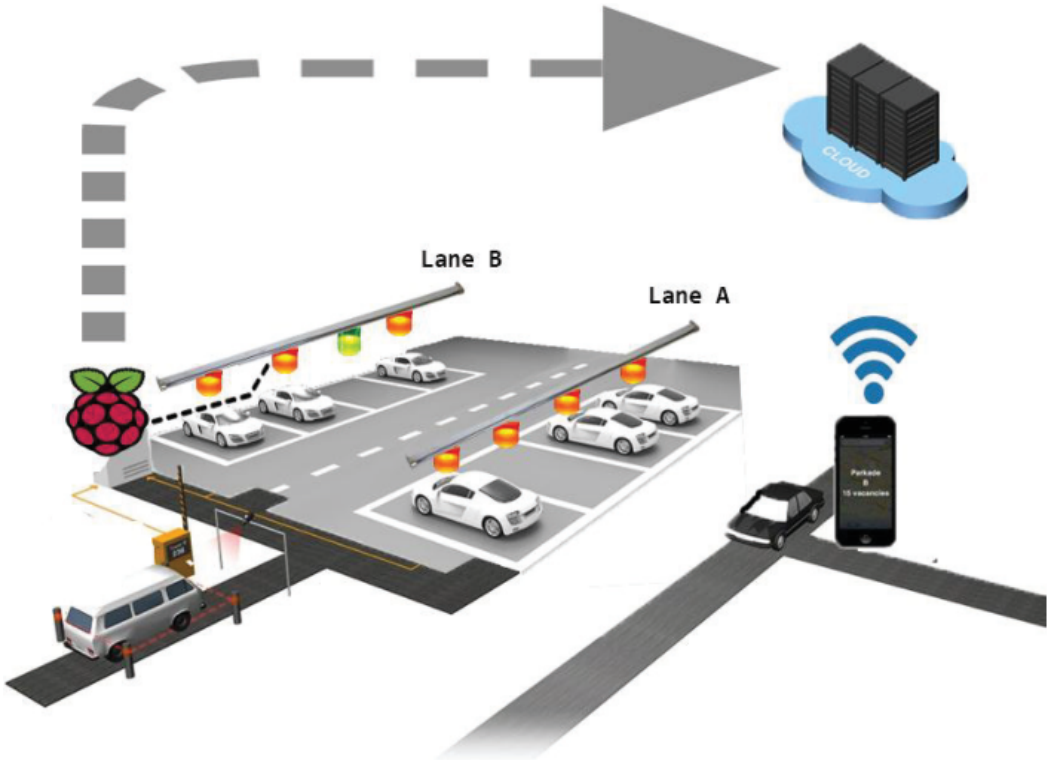
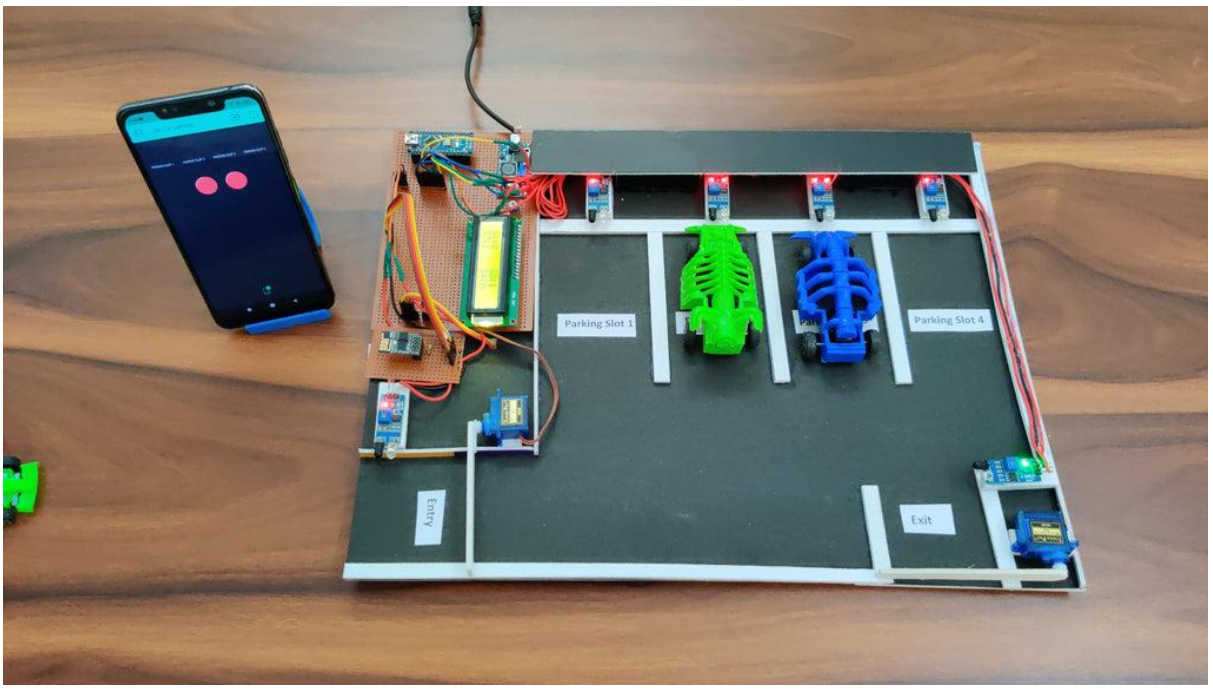
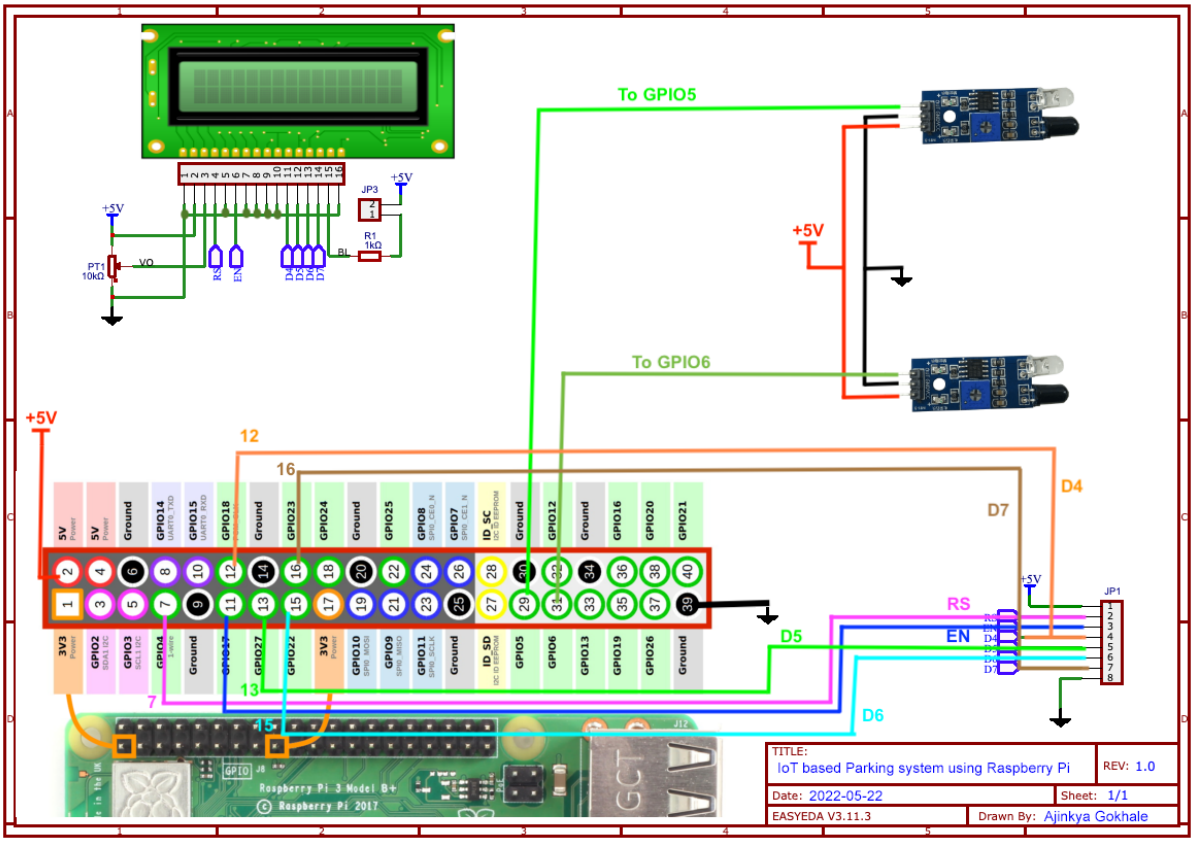


Figure 1: Smart Parking System





Smart parking schematic

The real-time parking availability system can benefit drivers and alleviate parking issues in the following ways:

- **Reduce driver frustration and stress:** The system provides drivers with real-time information on parking availability, which can help them to avoid circling around looking for parking spaces. This can reduce driver frustration and stress.
- **Improve traffic flow and reduce congestion:** By eliminating the need for drivers to circle around looking for parking spaces, the system can improve traffic flow and reduce congestion.
- **Increase parking revenue for parking garage operators**: By making it easier for drivers to find and pay for parking, the system can help parking garage operators to increase their revenue.
- **The seamless flowing of traffic:**Public transport routes can be adjusted in real-time according to need, and smart traffic lights systems can improve congestion.
- **Energy efficiency can be improved:**One can easily track down the power consumption & energy consumption by monitoring in real-time.
- **Cities can be made safer:** Cities can use technology to improve residents' safety and improve response times with the widespread use of Wi-Fi communications and IoT technology.
- **Encouragement of greater citizen engagement:**Citizens can respond to daily problems enabling neighbours to connect

and share resources to improve communities and neighbourhoods.

Conclusion :

The Smart Parking project is a viable solution for alleviating parking issues in urban areas. The system uses IoT sensors and Raspberry Pi to detect parking space occupancy and send the data to a mobile app. The mobile app provides drivers with real-time information on parking availability, which can help them to avoid circling around looking for parking spaces. This can reduce driver frustration and stress, improve traffic flow and reduce congestion, and increase parking revenue for parking garage operators.