SMART PARKING SYSTEM USING IOT

TEAM MEMBER

Rajeswari .S: (912221106301)

Phase 4 Submission Document

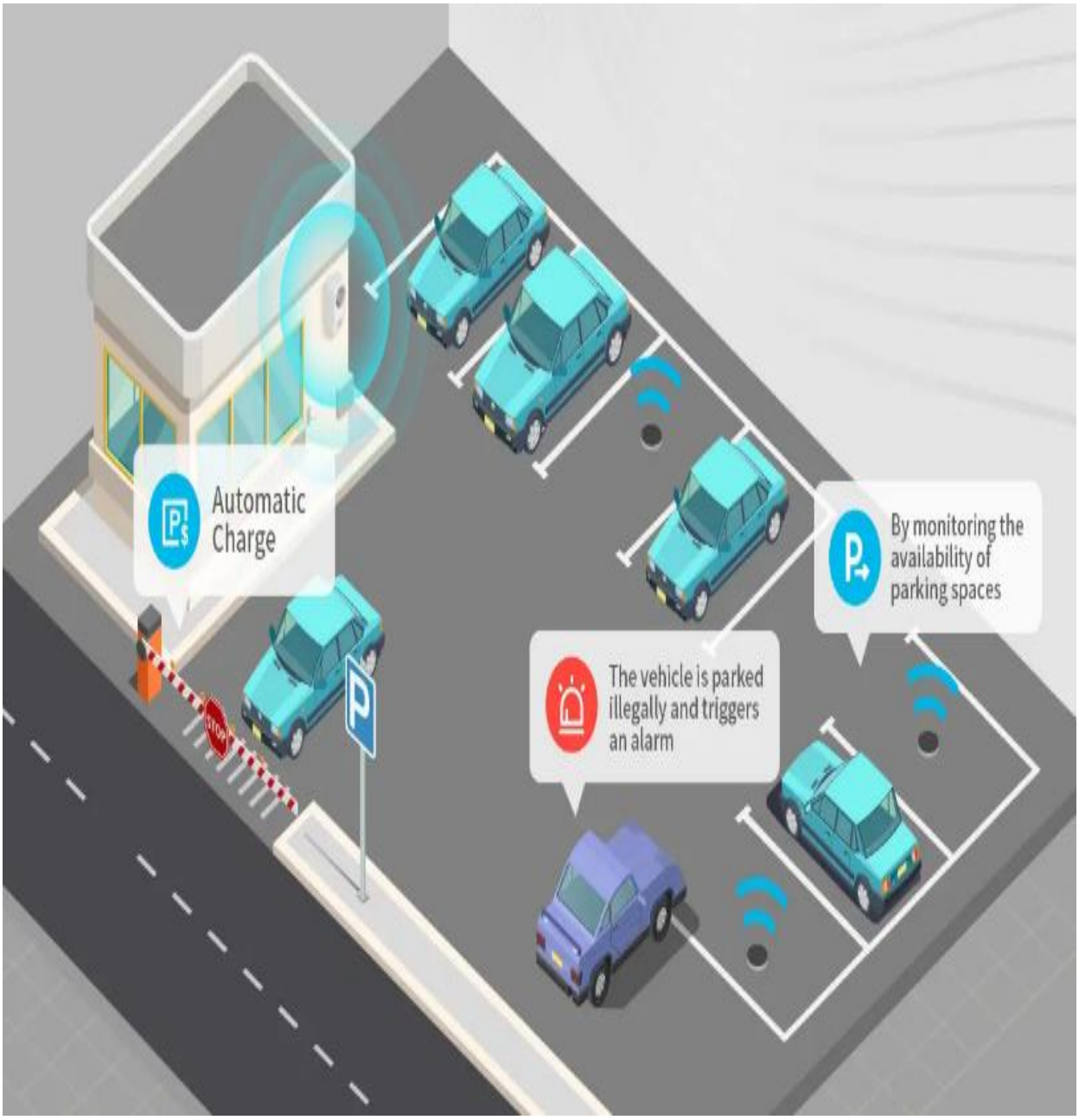
Project: SMART PARKING

INTRODUCTION:

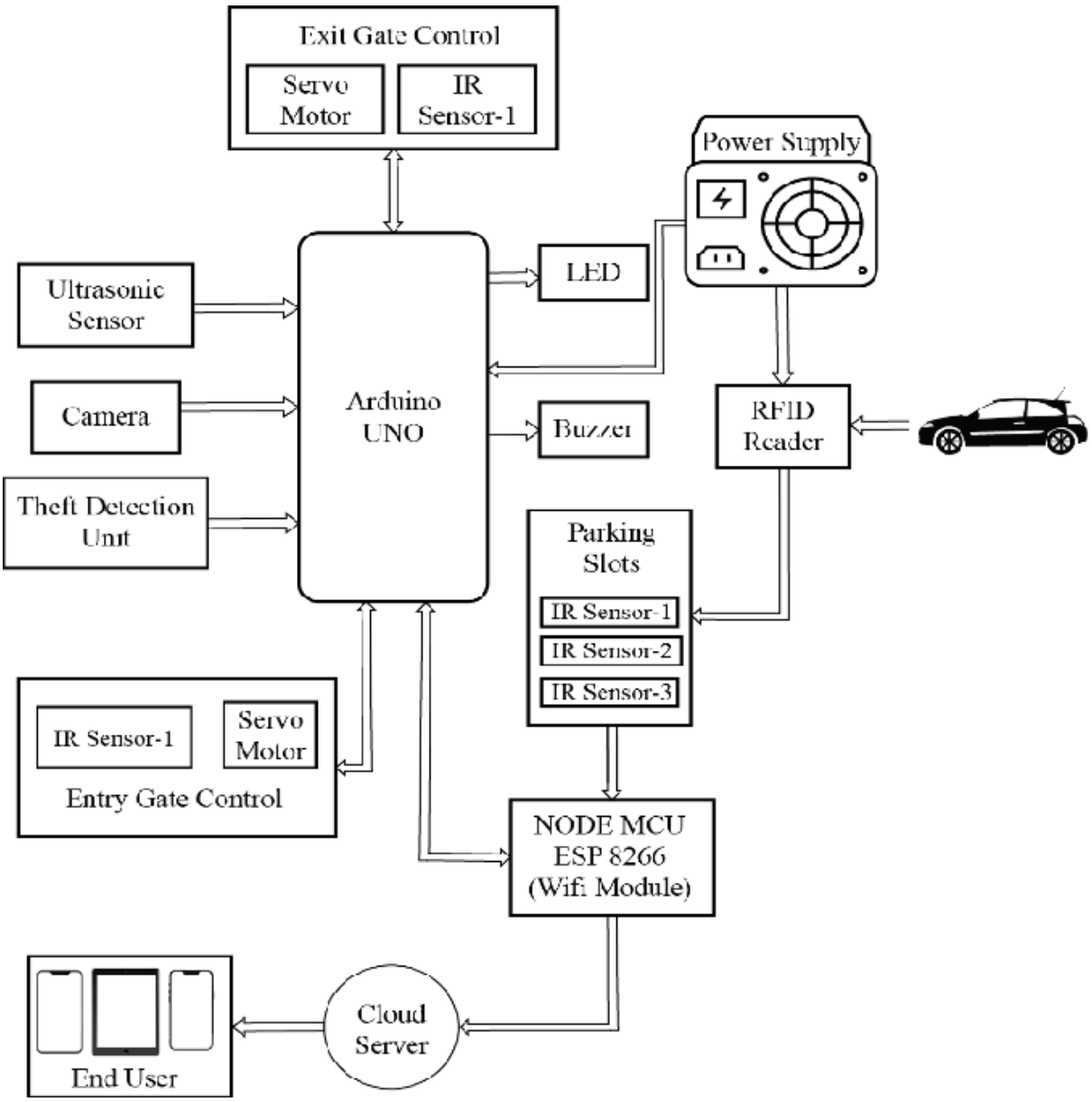
The idea of IoT started with the identity of things for connecting various devices. These devices can be controlled or monitored through computers. Over internet. IoT contains two prominent words “Internet” and “Things”, where Internet is a vast network for connecting servers with devices. Internet enables the information to be sent, receive, or even communicate with the devices. The parking problem causes air pollution and traffic congestion. In today’s scenario, parking space is hard to search in a day-to-day life for the people. According to the recent survey, there will be a rapid increase in the vehicle’s population of over 1.6 billion around 2035.Around one million barrels of world’s oil is being burnt everyday Thus, smart parking system is the key solution to reduce the waste stage of the fuel. The solution for the problems that is being raised. The smart parking can be a solution to minimise user’s time and efficiency as well as the overall cost of the fuel burnt in search of the parking space. In this, the data is collected from the sensor and through analysing and processing, the output is obtained.

This data gets transmitted in the devices which extracts the relevant information and sends it to the Arduino device which gives the command instruction for the data to devices simultaneously. Arduino sends the signal to the servo motor along with GSM module which further gives instructions and notification to the user. When the user enters in the parking area, RFID card allotted to the registered user is scanned by the reader module thus ensuring the security of the user identity. The second section of the paper includes the cloud web services which act as a mediator between the user and the car parking area. The cloud is updated according to the availability of parking area. The cloud service is administered by the admin

but it can also be viewed by the user to check the availability. The third section of the paper is the user side. The user gets notified for the availability via SMS through GSM module. The user interacts with the cloud as well as parking area. The user gets the notification when the parking availability is full which saves the time for the user.



BLOCK DIAGRAM:



Program:

Smart Parking:

```
def __init__(self, total_spaces):
```

```
        self.total_spaces = total_spaces
        self.parking_spaces = {i: False for i in range(1, total_spaces + 1)}

    def park_car(self):
        for space, status in self.parking_spaces.items():
            if not status:
                self.parking_spaces[space] = True
                print(f"Car parked at space {space}.")
                break
        else:
            print("Parking lot is full. Car cannot be parked.")

    def remove_car(self, space):
        if 1 <= space <= self.total_spaces and self.parking_spaces[space]:
            self.parking_spaces[space] = False
            print(f"Car removed from space {space}.")
        else:
            print(f"No car found at space {space}.")

    def display_status(self):
        print("Parking Lot Status:")
        for space, status in self.parking_spaces.items():
            print(f"Space {space}: {'Occupied' if status else 'Available'}")



def main():
```

```
    total_spaces = 10  # Total parking spaces in the parking lot
    parking_system = SmartParkingSystem(total_spaces)

    while True:
        print("\n1. Park Car")
        print("2. Remove Car")
        print("3. Display Parking Lot Status")
        print("4. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            parking_system.park_car()
        elif choice == '2':
            space = int(input("Enter the space number to remove car: "))
            parking_system.remove_car(space)
        elif choice == '3':
            parking_system.display_status()
        elif choice == '4':
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")


if __name__ == "__main__":
    main()
```

In this program, we have two classes: ParkingSpace and ParkingLot. The ParkingSpace class represents an individual parking space and has methods to occupy and release the space. The ParkingLot class manages a collection of parking spaces and has methods to add spaces, occupy spaces, and release spaces.

HARDWARE DESCRIPTION:

Microcontroller:

Raspberry pi is used as microcontroller for processing the data coming from sensors. Due to its Small size, high processing power, we preferred using Raspberry pi over other processors available in the market. Another advantage is that, Raspberry pi is a general purpose computer having its own operating system such as Raspbian, Windows 10 IOT core, Moebius etc. For our model we installed Raspbian which is the most popular operating system for IOT applications Raspbian has been used extensively because it is based on Debian (Linux) which makes it easy to use and it also protects against malware. The output of the IR sensors is saved in the text file in Raspberry pi.

Sensors:

We used IR sensors at the parking slots and at both the gates. These sensors are connected to the microcontroller through wires. The output pins are connected to the GPIO pins of Raspberry pi. Raspberry pi has total 26 GPIO pins out of which 2 are used by the servo motors. Hence total 24 sensors can be connected to Raspberry Pi through wires. These connections can also be made wireless and the number of sensors integrated can be increased by using MUX.

Servo Motors:

Two servo motors are used at the ENTRANCE and EXIT gate which are interfaced using the python script. The angle of rotation of both the motors is specified in the python script. Like IR sensors servo motors are also connected with wired connection.

Pi-Camera Module:

The function of camera module is to capture the image and pass these pictures to the processing unit. In our model, pictures are captured when the camera module receives the signal from IR signal which is situated at the entrance gate.

Network Protocols:

A parking system also requires protocols to ensure IoT devices' and sensors' connectivity in the parking lot. These can be MQTT, LoRaWAN, Zigbee protocol for wireless IoT networks, or else.

Such a system also requires video transmission protocols if it uses video surveillance. For example, Webby Lab used RTCP for our Propuskator project. This way, we provided a real-time video stream users can monitor through the application.

User Interface:

A mobile or web application is the final component of an IoT-based smart parking system. As a rule, such apps ensure parking management, time tracking, reservation, billing tools, data logging, remote video surveillance, guest passes, and driver authorization.

Conclusion and Future Work :

The concept of smart cities have always been a dream. There have been advancements made from the past couple of years to make smart city dream to reality. The advancement of internet of things and cloud technologies has given rise to new possibilities in terms of smart cities. Smart parking facilities have always been the core of constructing smart cities. The system provides a real time process and information of the parking slots. This paper enhances the performance of saving users time to locate an appropriate parking space. It helps to resolve the growing problem of traffic congestion. As for the future work the users can book a parking space from a remote location. GPS, reservation facilities and license plate scanner can be included in the future.