CHAPTER 3

Smart Parking System using IOT

3.1 Introduction

The Smart Parking System using IoT is designed to tackle urban parking challenges by leveraging advanced technologies. Traditional parking systems often lead to congestion, inefficient space utilization, and security concerns. The integration of IoT sensors, RFID authentication, cloud computing, and mobile applications provides a seamless and automated approach to parking management, ensuring efficiency, security, and convenience.

This chapter explores the core functionalities, components, and working principles of the IoT-based Smart Parking System, detailing how it enhances the overall parking experience while optimizing space usage.

3.2 Component Selection

The hardware components included the following:

- Arduino UNO
- Gear Motor
- ESP-WROOM-32
- Motor Driverr
- IR Sensor
- RFID RC522 Module
- Adapter

- Capacitor
- Trasnsistor
- Bread Board
- Pin-Pin Wires

The software components include the following:

- Flutter
- Firebase
- arduino IDE

3.3 Hardware Components

3.3.1 Arduino UNO

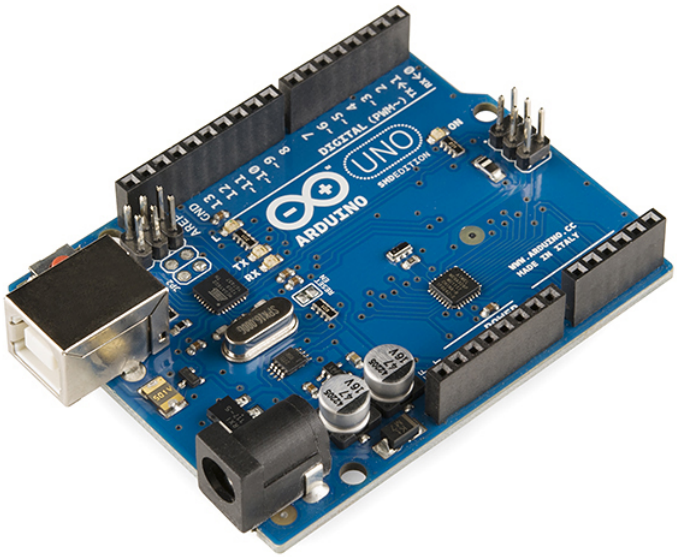

FIGURE 3.1: Arduino UNO

The Arduino Uno serves as the main microcontroller in the system, handling data processing from the RFID module and IR sensor. It controls the gate mechanism by sending

signals to the gear motor based on authentication results. It also communicates with the ESP32 module to update the Firebase database with real-time parking information.

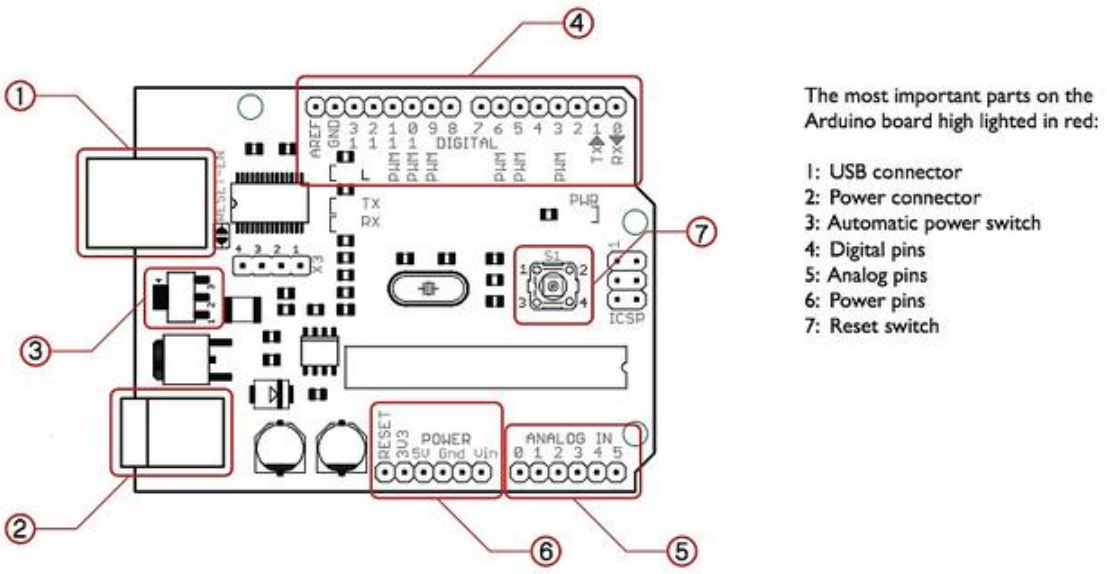

FIGURE 3.2: Arduino UNO Pin Diagram

3.3.2 Gear Motor

The 30 RPM gear motor is responsible for opening and closing the parking gate upon successful authentication of an RFID tag. It operates based on signals from the Arduino Uno and is controlled through a relay circuit. The motor ensures smooth operation of the gate, enhancing automation in the system.

FIGURE 3.3: Gear Motor

3.3.3 ESP-WROOM-32

The ESP-WROOM-32, part of the ESP32 series.This module provides wireless connectivity for the parking system, enabling real-time data transfer to Firebase. It receives

data from the Arduino Uno, processes it, and updates parking slot availability in the cloud. Its built-in WiFi capabilities allow remote monitoring of parking spaces through the mobile app.

FIGURE 3.4: ESP-WROOM-32

3.3.4 Motor Driver

The L298N motor driver is used to control the gear motor responsible for opening and closing the parking gate. Since the Arduino Uno cannot directly supply the required current to the motor, the motor driver acts as an interface, allowing efficient control of motor speed and direction. It receives signals from the microcontroller and adjusts the motor's operation accordingly, ensuring smooth gate movement.

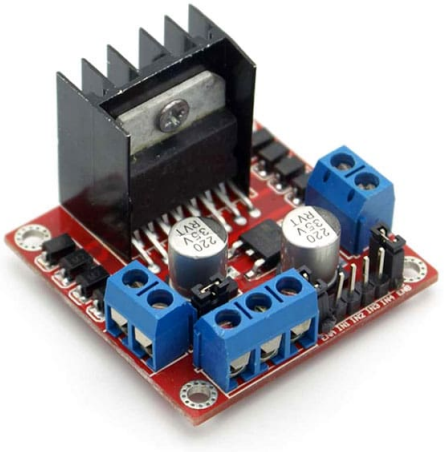

FIGURE 3.5: Motor Driver

3.3.5 IR Sensor

The IR sensor is used to detect the presence of a vehicle in a parking slot by emitting infrared light and measuring the reflection from an object. When a vehicle is parked, the sensor detects the obstruction and sends a signal to the Arduino Uno, which updates the parking slot status in Firebase. This enables real-time monitoring of available and occupied parking spaces. The IR sensor ensures accurate detection and enhances the automation of the parking system.

FIGURE 3.6: IR Sensor

3.3.6 RFID-RC522 Module

The RFID-RC522 module is a contactless radio-frequency identification device that operates at 13.56 MHz. It is used to authenticate vehicles by scanning RFID tags assigned to authorized users. When an RFID tag is placed near the module, it reads the unique ID and sends it to the Arduino Uno for verification. If the tag is authorized, the system triggers the gear motor to open the parking gate and logs the entry in Firebase. This enhances security and ensures only registered vehicles can access the parking facility.

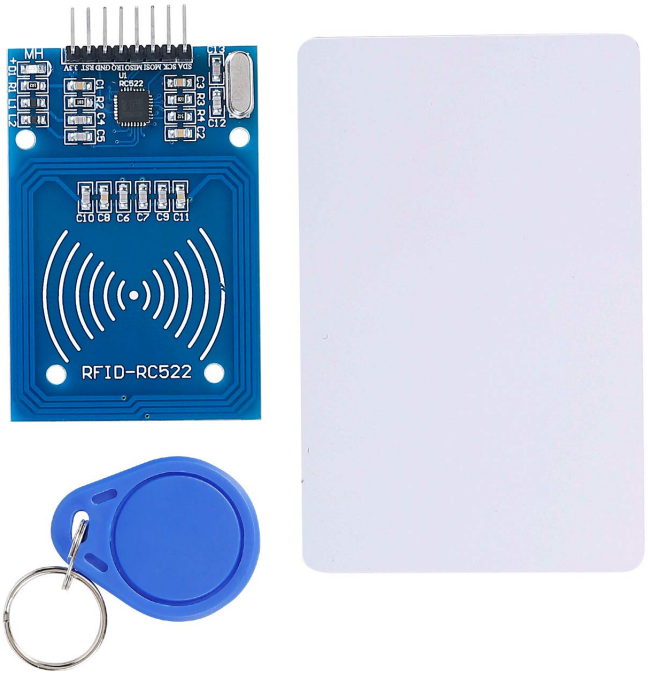

FIGURE 3.7: RFID-RC522 Module

3.3.7 Adapter

The 12V/1A adapter is a regulated power supply that provides a stable 12V DC output to power various components of the parking system. It ensures reliable operation of the Arduino Uno, ESP32 module, motor driver, and gear motor by preventing voltage fluctuations. A consistent power supply is essential to maintain system stability and avoid malfunctions, especially during continuous operation.

FIGURE 3.8: Adaptor

3.3.8 Capacitor

Capacitors, such as the 10µF, 100µF, and 0.1µF, play a fundamental role in stabilizing voltage levels and filtering out electrical noise in the system. These capacitors store and release electrical energy as needed, preventing sudden fluctuations that could affect the performance of the RFID-based parking system. The 10µF and 100µF capacitors are primarily used in power supply circuits to smooth out voltage variations, ensuring a steady 12V output for the components. The 0.1µF capacitor is often used for noise filtering in high-frequency circuits, helping to maintain reliable communication between the ESP32 module, Arduino Uno, and Firebase. By ensuring stable voltage and minimizing electrical interference, capacitors contribute to the efficient and fault-free operation of the system

FIGURE 3.9: Capacitor

3.3.9 Transistor

The BC547 is an NPN bipolar junction transistor (BJT) used for switching and amplification in the RFID-based parking system. It plays a fundamental role in controlling low-power signals that activate larger components, such as the motor driver for the gear motor. When a valid RFID tag is scanned, the Arduino Uno sends a signal to the transistor, which then allows current to flow, triggering the motor driver to open or close the parking gate. Additionally, the BC547 transistor helps in signal conditioning and prevents faults caused by voltage fluctuations, ensuring smooth and efficient operation of the system.

FIGURE 3.10: Transistor

3.3.10 Bread Board

The breadboard is a fundamental tool used for prototyping the RFID-based parking system, allowing components to be connected without the need for soldering. It provides a flexible and fault-free platform for assembling the circuit, making it easy to modify connections during development and testing. The breadboard enables efficient wiring of the Arduino Uno, ESP32 module, motor driver, IR sensor, and RFID module, ensuring that all components communicate effectively. By using a breadboard, the system can be tested and debugged before finalizing the hardware design, reducing errors and improving reliability.

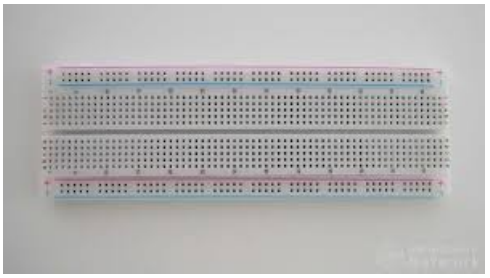

FIGURE 3.11: Bread Board

3.3.11 Pin-Pin Wire

Pin-to-pin wires, also known as jumper wires, are fundamental components used to establish electrical connections between different modules of the RFID-based parking system. These wires enable fault-free communication between the Arduino Uno, ESP32 module, motor driver, IR sensor, and RFID module by providing flexible and efficient wiring. Available in male-to-male, male-to-female, and female-to-female configurations, these wires allow easy prototyping on the breadboard and ensure secure connections without the need for soldering. Their versatility makes them essential for testing and modifying the circuit during development.

FIGURE 3.12: Pin-Pin Wire

3.4 Software Components

3.4.1 Flutter

Flutter is a fundamental open-source UI toolkit developed by Google for building cross-platform applications. In the RFID-based parking system, Flutter is used to develop a mobile application that allows users to monitor parking slot availability in real-time. The app communicates with Firebase to fetch and display data related to parking occupancy, gate access status, and user authentication. With its fast development features like hot reload and a rich set of customizable widgets, Flutter ensures a smooth and efficient user experience.

3.4.2 Firebase

Firebase is a fault-free backend-as-a-service (BaaS) provided by Google, offering real-time database management, authentication, cloud storage, and hosting services. In this