Faculty of Engineering and Computing

IOT ASSIGNMENT

IOT Assignment Individual Part

B1146

David Gong Hongshen

Bachelor of Computer Science (Intelligent Systems) (Honours)

September 2022

Table of Contents

IOT Assignment 1

IOT Assignment Individual Part 1

Table of Contents iii

List of Figures v

Chapter 1: Task 4 - Sensing Process 1

1.1 IR - Infrared (IR) HW-201 1

1.2 RFID - RC522 2

1.3 Ultrasonic Distance Sensor - HC-SR04 4

1.4 Summary 5

1.5 Code of Arduino Uno 1

1.6 Code interpretation for Arduino Uno 2

1.7 Code for ESP8266 Nodemcu 3

1.8 Code interpretation for ESP8266 Nodemcu 7

Chapter 2: Task 6 9

2.1 Introduction 9

2.2 Social Issues 11

2.2.1 Accessibility and Convenience 11

2.2.2 Traffic Congestion Reduction 11

2.2.3 Privacy Concerns 12

2.2.4 Security Risks 12

2.2.5 Impact on Local Businesses and Residents 13

2.3 Economic Issues 14

2.3.1 Infrastructure Costs 14

2.3.2 Maintenance and Operation Expenses 14

2.3.3 Potential for Revenue Generation 15

2.3.4 Employment Opportunities and Job Displacement 15

2.4 Ethical Issues 16

2.4.1 Data Collection and Usage 16
2.4.2 Equitable Access to Parking Resources 16
2.4.3 Environmental Impact 17
2.4.4 Legal and Regulatory Compliance 17
2.5 conclusion 18
References 19

List of Figures

figure 1 Infrared (IR) HW-201 1

figure 2 RFID - RC522 2

figure 3 Ultrasonic Distance Sensor - HC-SR04 4

figure 4 Overview 1

Chapter 1: **Task 4 - Sensing Process**

1.1 IR - Infrared (IR) HW-201

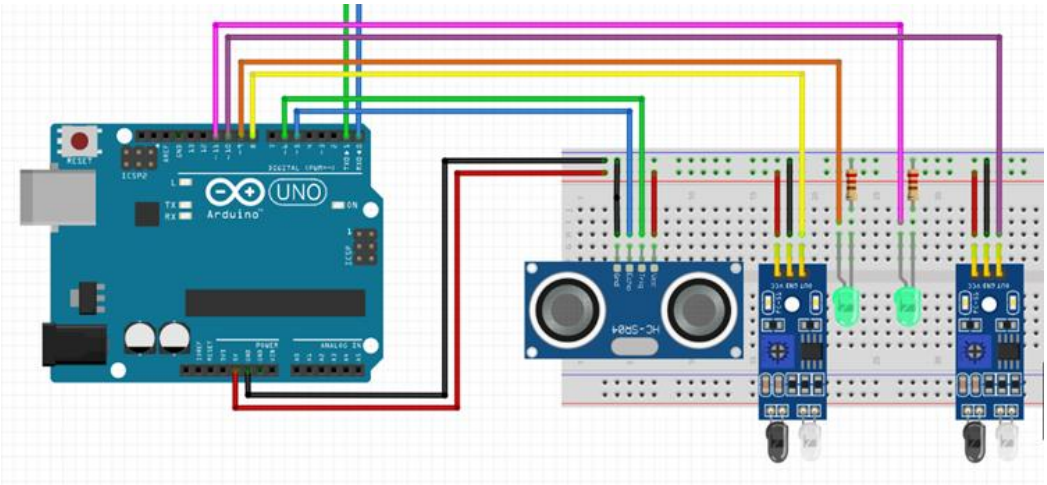

figure 1 Infrared (IR) HW-201

The IR sensors (such as the HW-201) can be used in a parking system to detect the presence of a vehicle in a parking space. Here's how they can be applied to a solution:

1. Install the IR sensors at each parking space. The sensor can be placed on the ground or mounted on a pole above the parking space.

2. The sensor emits an infrared beam, which is reflected back to the sensor when a vehicle is present in the space.

3. The sensor detects the reflected beam and sends a signal to the system's microcontroller, indicating that the parking space is occupied.

4. The system can then update its database or display information to show that the parking space is no longer available.

5. When the vehicle leaves the parking space, the absence of the reflected beam is detected by the sensor, and the system is notified that the parking space is now available.

Using IR sensors in a parking system has several advantages. Firstly, they are reliable and can detect the presence of a vehicle even in low light conditions. Secondly, they are relatively low-cost compared to other sensor technologies such as ultrasonic sensors. Additionally, IR sensors have a long lifespan and require minimal maintenance. Overall, using IR sensors in a parking system can improve the efficiency and accuracy of parking space management.

1.2 RFID - RC522

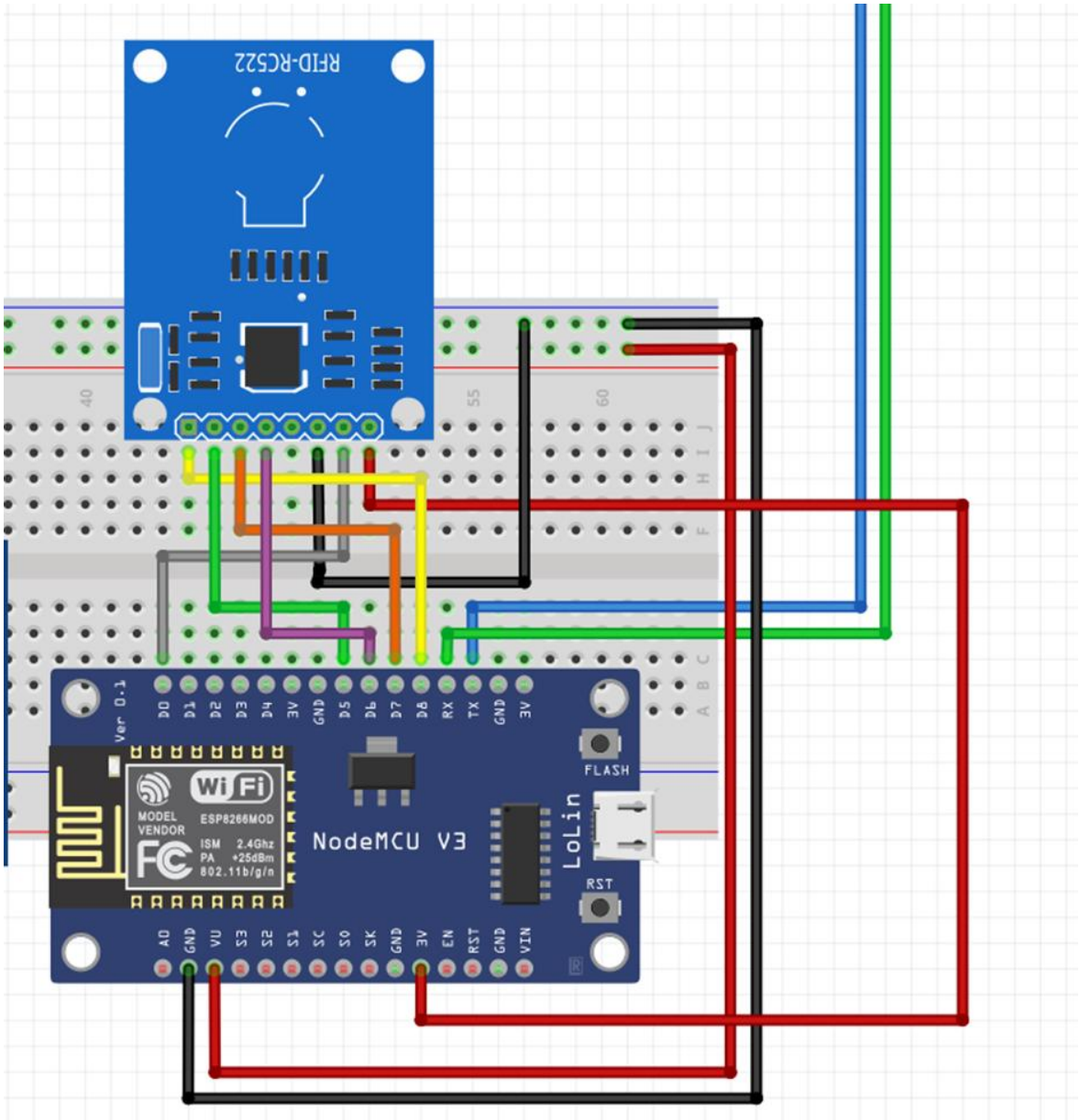

figure 2 RFID - RC522

The RFID-RC522 sensor can be used in a parking system to identify and track individual vehicles. Here's how they can be applied to a solution:

1. Install the RFID-RC522 sensor at the entrance and exit gates of the parking lot.

2. Each vehicle is equipped with an RFID tag, which contains a unique identification number.

3. When a vehicle approaches the entrance gate, the RFID sensor reads the identification number from the RFID tag.

4. The system checks the identification number against its database to verify if the user has a valid parking permit.

5. If the user has a valid permit, the entrance gate opens, and the vehicle is allowed to enter the parking lot.

6. The system records the time and location of the vehicle in the parking lot.

7. When the vehicle leaves the parking lot, it passes through the exit gate, and the RFID sensor reads the identification number again.

8. The system uses the identification number and parking information to calculate the parking fee.

9. The user pays the parking fee, and the exit gate opens, allowing the vehicle to exit the parking lot.

Using RFID-RC522 sensors in a parking system has several advantages. Firstly, they provide a secure and convenient way to identify and track vehicles. Secondly, they can reduce the chances of parking fraud, such as using a fake or stolen parking permit. Additionally, they can automate the parking fee calculation and payment process, saving

time for both the user and the parking lot management. Overall, using RFID-RC522 sensors in a parking system can improve the efficiency and security of parking space management.

1.3 Ultrasonic Distance Sensor - HC-SR04

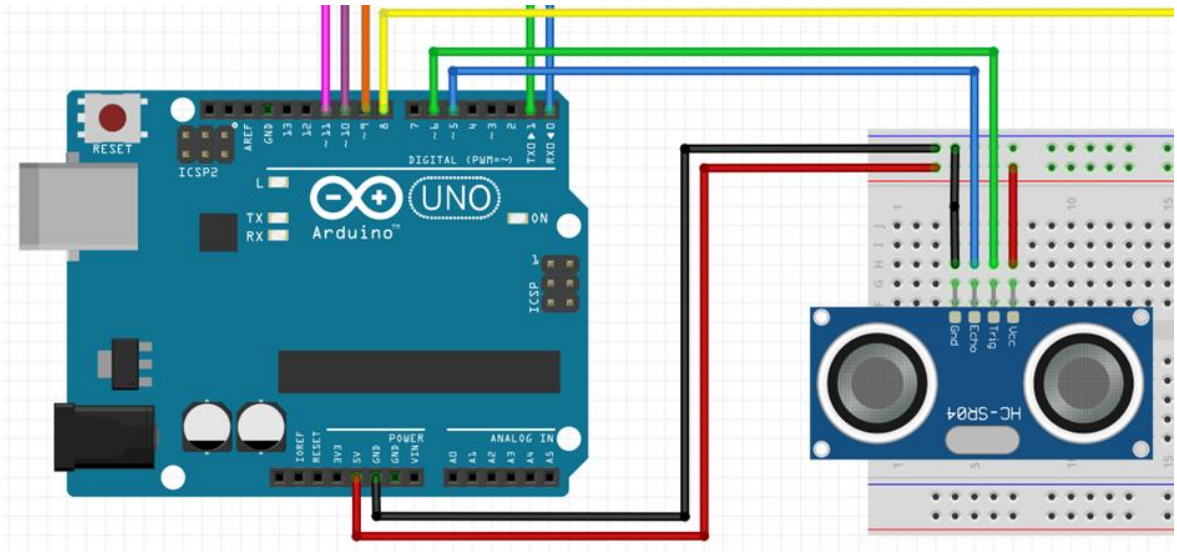

figure 3 Ultrasonic Distance Sensor - HC-SR04

The Ultrasonic Distance Sensor - HC-SR04 can be used in a parking system to measure the distance between a vehicle and a parking space. Here's how they can be applied to a solution:

1. Install the HC-SR04 sensor at each parking space, preferably mounted on a pole above the parking space.

2. The sensor emits an ultrasonic sound wave, which travels towards the ground.

3. The sound wave is reflected back to the sensor when it encounters an object, such as a vehicle.

4. The sensor measures the time taken for the sound wave to travel to the object and back, which is used to calculate the distance between the object and the sensor.

5. The system uses the distance measurement to determine if a vehicle is parked in the space or not.

6. When the vehicle leaves the parking space, the absence of the reflected sound wave is detected by the sensor, and the system is notified that the parking space is now available.

Using HC-SR04 sensors in a parking system has several advantages. Firstly, they are relatively low-cost and easy to install compared to other sensor technologies such as cameras or infrared sensors. Secondly, they are accurate and reliable in measuring distances, even in outdoor environments. Additionally, they can be used in combination with other sensors such as cameras or RFID sensors to provide a more comprehensive parking space management system. Overall, using HC-SR04 sensors in a parking system can improve the efficiency and accuracy of parking space management.

1.4 Summary

The three sensors that can be used in a parking system are the Infrared (IR) sensor (such as HW-201), the RFID-RC522 sensor, and the Ultrasonic Distance Sensor (HC-SR04).

The IR sensor can be used to detect the presence of a vehicle in a parking space by emitting and detecting infrared radiation. The RFID-RC522 sensor can be used to identify and track individual vehicles by reading a unique identification number from an RFID tag attached to the vehicle. The HC-SR04 sensor can be used to measure the distance between a vehicle and a parking space by emitting and receiving ultrasonic sound waves.

To implement these sensors, an ESP8266 NodeMCU can be used to connect to an Arduino Uno, which can be used as the main controller for the parking system. The ESP8266 NodeMCU can provide Wi-Fi connectivity and data transmission capabilities, while the

Arduino Uno can process the sensor data and control the entrance and exit gates of the parking lot. The system can also be programmed to update a database, display parking space availability information, and calculate parking fees.

Overall, using a combination of these sensors with an ESP8266 NodeMCU and an Arduino Uno can provide an efficient and effective parking system with improved accuracy and convenience for users.

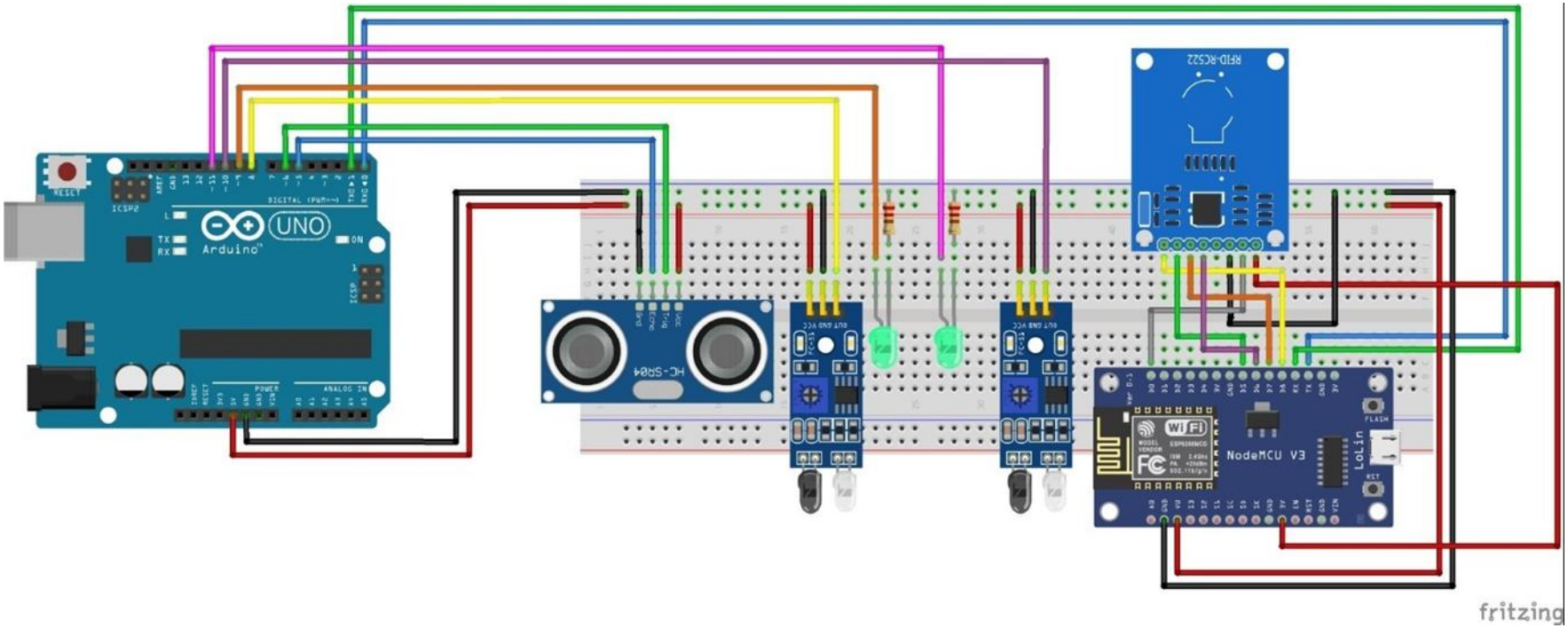

figure 4 Overview

1.5 Code of Arduino Uno

```
const int ir1 = 8;
const int led1 = 9;

const int ir2 = 10;
const int led2 = 11;

//hc04

const int trigPin = 6; //trigPin D3
const int echoPin = 5; //echoPin D1
// defines variables
long duration;
int led1Val,led2Val;
int cm = 0;
void setup() {
  pinMode(ir1,INPUT);
  pinMode(led1, OUTPUT);
  pinMode(ir2,INPUT);
  pinMode(led2, OUTPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(115200);
}
void loop() {
  if (digitalRead(ir1)==LOW){ //object is detected
    //Serial.println("Led1 LOW");
    digitalWrite(led1,LOW);
    led1Val = 0;
  }
  else{ //no object detected
  //  Serial.println("Led1 HIGH");
    digitalWrite(led1,HIGH);
    led1Val = 1;
  }


  if (digitalRead(ir2)==LOW){ //object is detected
   // Serial.println("Led2 LOW");
    digitalWrite(led2,LOW);
    led2Val = 0;
  }
  else{ //no object detected
   // Serial.println("Led2 HIGH");
    digitalWrite(led2,HIGH);
    led2Val = 1;
  }
  hc04();
  char s[200];
  sprintf(s, "{\"led1\":%d,\"led2\":%d,\"cm\":%d}", led1Val,led2Val,cm);
  Serial.println(s);
  delay(20000);
}


void hc04(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in
microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  cm = duration * 0.034 / 2;
}
```

1.6 Code interpretation for Arduino Uno

This code is written for an Arduino microcontroller and appears to be used for an object detection system with two infrared sensors (IR) and one ultrasonic distance sensor (HC-SR04). The code controls two LEDs based on the object detection status. Here's a step-by-step interpretation of the code:

1. Define constants for IR sensor and LED pin numbers for two sets (ir1, led1, ir2, led2) and pin numbers for the ultrasonic sensor's trigger and echo pins (trigPin, echoPin).

2. Declare global variables for duration, led1Val, led2Val, and cm (centimeters).

3. In the setup() function, configure the pinMode for IR sensors (INPUT), LEDs (OUTPUT), trigPin (OUTPUT), and echoPin (INPUT). Begin a serial communication with a baud rate of 115200.

4. In the loop() function, check the states of the IR sensors:

   (1) If ir1 is LOW (object detected), turn off LED1 and set led1Val to 0.

   (2) If ir1 is HIGH (no object detected), turn on LED1 and set led1Val to 1.

   (3) If ir2 is LOW (object detected), turn off LED2 and set led2Val to 0.

   (4) If ir2 is HIGH (no object detected), turn on LED2 and set led2Val to 1.

5. Call the hc04() function to measure the distance using the ultrasonic sensor.

6. Format a string 's' containing the JSON representation of the led1Val, led2Val, and cm variables.

7. Print the JSON string 's' to the serial monitor.

8. Wait for 20 seconds (20000 milliseconds) before running the loop again.

9. The hc04() function measures the distance using the ultrasonic sensor by sending a trigger pulse, waiting for the echo pulse, calculating the duration of the echo pulse, and converting the duration to distance in centimeters.

1.7 Code for ESP8266 Nodemcu

```
#include <ArduinoJson.h>
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>

#define SS_PIN 15  //D8
#define RST_PIN 16 //D0

// Define ThingSpeak credentials
unsigned long channelID = 2109747;
const char* writeAPIKey = "810Y1B680Q80M34N";

MFRC522 rfid(SS_PIN, RST_PIN); //实例化类
MFRC522::MIFARE_Key key;
//D8 SS
//D5 SCK
//D7 MOSI
//D6 MISO
//D0 RST

Servo servo;

// WiFi parameters to be configured
const char* ssid = "FIRSTCITY"; // Write here your router's username
```

```
const char* password = "fcuc1234"; // Write here your router's passward


//const int sg = 2; //sg90 2 D4

String UART_String="";
int angle = 0;
int led1 = 0;
int led2 = 0;
int cm = 0;
long duration;
int i;
String rfidNumber="0";
// Init array that will store new NUID
byte nuidPICC[4];

 //using Wifi class
WiFiClient client;

void setup() {

  Serial.begin(115200);
  // Connect to WiFi
  WiFi.begin(ssid, password);
  // while wifi not connected yet, print '.'
  // then after it connected, get out of the loop
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  //print a new line, then print WiFi connected and the IP address
  Serial.println("WiFi connected");
  // Print the IP address
  Serial.println(WiFi.localIP());

  //RFID
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522
  Serial.println();
  Serial.print(F("Reader :"));
  rfid.PCD_DumpVersionToSerial();
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  Serial.println();
  Serial.println(F("This code scan the MIFARE Classic NUID."));
  Serial.print(F("Using the following key:"));
  printHex(key.keyByte, MFRC522::MF_KEY_SIZE);



  // Initialize ThingSpeak
  ThingSpeak.begin(client);
}

void loop() {
  rfidNumber = "0";
  rfidFunction();
  // hc04();
```

```
 if (Serial.available()) {
  String data = Serial.readStringUntil('\n');
  Serial.print("Received data: ");
  Serial.println(data);
 StaticJsonDocument<128> jsonDoc; // Create a JSON document
    DeserializationError error = deserializeJson(jsonDoc, data); //
Deserialize the JSON string into the JSON document
  if (error) {
   Serial.print("Error deserializing JSON: ");
   Serial.println(error.c_str());
   return;
  }
  led1 = jsonDoc["led1"]; // Get the value of led1
  led2 = jsonDoc["led2"]; // Get the value of led2
  cm = jsonDoc["cm"]; // Get the value of led2
  Serial.print("led1 value: ");
  Serial.println(led1);
  Serial.print("led2 value: ");
  Serial.println(led2);
  Serial.print("cm value: ");
  Serial.println(cm);
 }


// Send sensor value to ThingSpeak
 ThingSpeak.setField(1,led1);
 ThingSpeak.setField(2,cm);
 ThingSpeak.setField(3,rfidNumber);
 ThingSpeak.setField(4,led2);
 int status = ThingSpeak.writeFields(channelID,writeAPIKey);
 if (status == 200) {
  Serial.println("Sensor value sent to ThingSpeak!");
 }else {
    Serial.println("Problem updating channel. HTTP error code " +
String(status));
 }
 delay(20000);
 Serial.println("----------------------------------------");
}

void rfidFunction(){
  //Reset the loop if no new card present on the sensor/reader. This
saves the entire process when idle.
 if ( ! rfid.PICC_IsNewCardPresent())
  return;

 // Verify if the NUID has been readed
 if ( ! rfid.PICC_ReadCardSerial())
  return;

 Serial.print(F("PICC type: "));
 MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
 Serial.println(rfid.PICC_GetTypeName(piccType));

 // Check is the PICC of Classic MIFARE type
 if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
   piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
```

```
    piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE Classic."));
    return;
  }

  if (rfid.uid.uidByte[0] != nuidPICC[0] ||
    rfid.uid.uidByte[1] != nuidPICC[1] ||
    rfid.uid.uidByte[2] != nuidPICC[2] ||
    rfid.uid.uidByte[3] != nuidPICC[3] ) {
    Serial.println(F("A new card has been detected."));

    // Store NUID into nuidPICC array
    for (byte i = 0; i < 4; i++) {
     nuidPICC[i] = rfid.uid.uidByte[i];
    }

    Serial.println(F("The NUID tag is:"));
    Serial.print(F("In hex: "));

    Serial.print(hexToStr(rfid.uid.uidByte, rfid.uid.size));
    //   printHex(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
    Serial.print(F("In dec: "));
    rfidNumber = decToStr(rfid.uid.uidByte, rfid.uid.size);
    // printDec(rfid.uid.uidByte, rfid.uid.size);
    Serial.println(rfidNumber);

   }
   else
   {
    Serial.println(F("The NUID tag is:"));
    Serial.print(F("In hex: "));
    Serial.print(hexToStr(rfid.uid.uidByte, rfid.uid.size));
    //printHex(rfid.uid.uidByte, rfid.uid.size);

    Serial.println();
    Serial.print(F("In dec: "));
    rfidNumber = decToStr(rfid.uid.uidByte, rfid.uid.size);
    Serial.println(rfidNumber);
    Serial.println(F("Card read previously."));
   }
   // Halt PICC
   rfid.PICC_HaltA();

   // Stop encryption on PCD
   rfid.PCD_StopCrypto1();
  }

/**
  Helper routine to dump a byte array as hex values to Serial.
*/
void printHex(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
   Serial.print(buffer[i] < 0x10 ? " 0" : " ");
   Serial.print(buffer[i], HEX);
  }
}
```

```
/**
  Helper routine to dump a byte array as dec values to Serial.
*/
void printDec(byte *buffer, byte bufferSize) {
 for (byte i = 0; i < bufferSize; i++) {
  Serial.print(buffer[i] < 0x10 ? " 0" : " ");
  Serial.print(buffer[i], DEC);
 }
}

//convert hex to str
String hexToStr(byte *buffer, byte bufferSize) {
 String str = "";
 for (byte i = 0; i < bufferSize; i++) {
  str += String(buffer[i], HEX);
 }
 return str;
}

String decToStr(byte *buffer, byte bufferSize) {
 String str = "";
 for (byte i = 0; i < bufferSize; i++) {
  str += String(buffer[i], DEC);
 }
 return str;
}
```

1.8 Code interpretation for ESP8266 Nodemcu

This code is for an IoT project that involves an ESP8266 microcontroller, an RFID reader (MFRC522), a servo motor, and an ultrasonic sensor. The code connects to a Wi-Fi network, reads RFID tags, receives sensor data through a serial connection, and sends the collected data to ThingSpeak, a cloud service for IoT analytics.

1. Include necessary libraries, such as ArduinoJson, Servo, SPI, MFRC522, ESP8266WiFi, and ThingSpeak.

2. Define pins, ThingSpeak credentials, and Wi-Fi credentials.

3. Instantiate objects for the RFID reader (MFRC522) and the Servo motor.

4. Declare global variables for sensor values, RFID card numbers, Wi-Fi client, and other required data.

5. In the setup() function, initialize the serial communication, Wi-Fi connection, SPI bus for RFID, RFID reader, and ThingSpeak client.

6. In the loop() function, perform the following steps:

   (1) Call the rfidFunction() to read RFID tags.

   (2) If serial data is available, read and parse the JSON string containing sensor data (led1, led2, cm).

   (3) Update ThingSpeak fields with the sensor values and RFID card number.

   (4) Send the updated data to ThingSpeak.

   (5) Wait for 20 seconds before repeating the loop.

7. The rfidFunction() reads the RFID tags, detects new cards, and prints the card numbers in both hexadecimal and decimal formats.

8. Helper functions printHex(), printDec(), hexToStr(), and decToStr() are used for printing and converting byte arrays to hexadecimal and decimal strings.

Chapter 2: **Task 6**

2.1 Introduction

The Internet of Things (IoT) has revolutionized various aspects of our lives, from smart homes to intelligent transportation systems. One of the emerging applications of IoT is in parking systems, aimed at making the process of finding and managing parking spaces more efficient and convenient. As cities grow and urban populations increase, the demand for parking spaces has surged, leading to traffic congestion, pollution, and stress for drivers. To address these challenges, IoT-enabled smart parking systems have been introduced, which utilize a network of sensors, devices, and software to optimize parking space allocation and management.

IoT solutions for parking systems can vary in their implementation, but generally involve a combination of sensors, cameras, and software to detect the availability of parking spaces, guide drivers to available spots, and manage payments and reservations. In addition, these systems can be integrated with other city infrastructure and services, such as traffic management and public transportation, to create a more seamless urban experience. As a result, smart parking systems have the potential to significantly improve the quality of life in urban areas by reducing traffic congestion, cutting down on emissions, and optimizing land use [1].

Despite the numerous benefits associated with IoT-based parking systems, there are also several social, economic, and ethical issues that arise with their implementation. These issues are important to consider, as they can have a significant impact on the acceptance and success of smart parking systems, as well as their long-term implications for urban development.

Social issues encompass a range of concerns, from accessibility and convenience for all users to privacy and security risks. For instance, while IoT-enabled parking systems may make the process of finding parking more efficient, they also raise questions about data privacy, as personal information may be collected and used by these systems. Furthermore, there are concerns about the potential for IoT systems to be hacked or otherwise compromised, which could result in negative consequences for users and cities alike [2].

Economic issues involve the costs and benefits associated with implementing IoT solutions for parking systems. While these systems have the potential to generate revenue through more efficient parking management, they also require significant investments in infrastructure, maintenance, and operation. Additionally, the transition to smart parking systems may lead to job displacement, as traditional parking attendants and other related roles may no longer be needed [3].

Ethical issues in IoT-based parking systems revolve around data collection and usage, equitable access to parking resources, environmental impact, and legal and regulatory compliance. For example, there are concerns about the potential for discrimination and exclusion if smart parking systems favor certain users or neighborhoods over others, or if they prioritize certain types of vehicles (e.g., electric cars) [4].

In this discussion, we will explore the social, economic, and ethical issues linked to emerging IoT solutions for parking systems, in order to better understand their implications and potential solutions. By examining these issues, we can gain insight into the challenges and opportunities associated with the implementation of IoT-based parking systems, and ultimately, contribute to the development of more equitable, sustainable, and efficient urban environments.

2.2 Social Issues

2.2.1 Accessibility and Convenience

IoT-enabled parking systems have the potential to greatly improve accessibility and convenience for drivers. By providing real-time information on parking availability, these systems can help drivers find open parking spaces more quickly and easily, reducing the time and effort spent searching for a spot. Additionally, smart parking systems can offer features such as online reservations and mobile payment options, further streamlining the parking experience [5].

However, it is important to ensure that these benefits are accessible to all users, including those with disabilities or limited access to technology. For example, parking systems should be designed with features like audio guidance and wheelchair accessibility to accommodate users with varying needs. Furthermore, cities should consider the potential digital divide that may arise from the implementation of IoT-based parking systems, and work to provide alternative options for those without access to smartphones or the internet [6].

2.2.2 Traffic Congestion Reduction

One of the main benefits of smart parking systems is their potential to reduce traffic congestion. Studies have shown that a significant portion of urban traffic is caused by drivers searching for parking spaces, which contributes to increased emissions, fuel consumption, and frustration [7]. By providing real-time information on parking availability and guiding drivers to open spots, IoT-enabled parking systems can help to alleviate this problem, leading to smoother traffic flow and improved quality of life for urban residents.

However, there may be unintended consequences to this reduction in traffic congestion. For example, if parking becomes more convenient and readily available, it may encourage more people to drive and rely on personal vehicles, rather than opting for public transportation or alternative modes of transportation like walking or biking. This could, in turn, contribute to increased traffic and pollution in the long run [8].

2.2.3 Privacy Concerns

As with many IoT applications, privacy is a significant concern when it comes to smart parking systems. These systems often rely on data collection from sensors, cameras, and user devices to function, which can include information about individual drivers, their vehicles, and their parking habits. While this data is necessary for the operation of the system, there is a risk that it could be used in ways that violate users' privacy, such as for targeted advertising or surveillance [9].

To mitigate these concerns, it is essential for IoT parking system developers and operators to implement robust data protection measures and adhere to relevant privacy regulations, such as the General Data Protection Regulation (GDPR) in the European Union. This includes practices like anonymizing data, obtaining user consent, and ensuring data is only used for its intended purpose [10].

2.2.4 Security Risks

Alongside privacy concerns, security risks are another significant social issue associated with IoT-based parking systems. The interconnected nature of these systems makes them vulnerable to cyberattacks and hacking, which could potentially disrupt parking operations, compromise user data, or even lead to unauthorized access to other city infrastructure [11]. This underscores the importance of ensuring that smart parking systems are developed and

maintained with robust security measures in place, such as strong encryption, regular software updates, and intrusion detection systems.

Additionally, it is crucial for cities and system operators to collaborate with cybersecurity experts and follow best practices for securing IoT devices and networks. This includes adhering to industry standards and guidelines, such as those developed by the National Institute of Standards and Technology (NIST) and the International Organization for Standardization (ISO) [12].

2.2.5 Impact on Local Businesses and Residents

The implementation of IoT-based parking systems can have both positive and negative effects on local businesses and residents. On one hand, more efficient parking management can lead to increased customer traffic for businesses, as potential customers are more likely to visit an area if parking is convenient and readily available. Furthermore, reduced traffic congestion and improved air quality can contribute to a more pleasant urban environment for residents [13].

On the other hand, there may be concerns about the potential gentrification and displacement of local communities due to the implementation of smart parking systems. For example, the increased desirability of areas with improved parking infrastructure may lead to rising property values and rent prices, which could force out long-time residents and small businesses. To address these concerns, it is essential for cities to work closely with communities and stakeholders to ensure that the benefits of IoT-based parking systems are equitably distributed and that potential negative impacts are mitigated [14].

2.3 Economic Issues

2.3.1 Infrastructure Costs

Implementing IoT solutions for parking systems can require significant investments in infrastructure, such as sensors, cameras, communication networks, and software. These upfront costs can be a barrier to adoption for some cities, particularly those with limited budgets or other competing priorities. However, it is important to consider the potential long-term savings and benefits that smart parking systems can offer, such as reduced traffic congestion, increased revenue from parking fees, and more efficient use of urban space [15].

When evaluating the costs and benefits of IoT-based parking systems, cities should also explore innovative financing and partnership models, such as public-private partnerships (PPPs) or collaborations with technology companies. These arrangements can help to distribute the costs and risks associated with implementing smart parking infrastructure, making it more feasible for cities of varying sizes and resources [16].

2.3.2 Maintenance and Operation Expenses

In addition to the initial infrastructure costs, IoT-based parking systems also entail ongoing maintenance and operation expenses. These costs can include software updates, sensor and camera maintenance, network management, and customer support. It is crucial for cities and system operators to plan for these ongoing expenses and to ensure that they are factored into the overall cost-benefit analysis of implementing smart parking solutions [17].

To optimize maintenance and operation expenses, cities can consider adopting data-driven approaches and leveraging predictive analytics to identify potential issues before they become critical. This can help to reduce the costs associated with reactive maintenance and system downtime, ultimately contributing to more efficient and cost-effective parking operations [18].

2.3.3 Potential for Revenue Generation

One of the key economic benefits of IoT-enabled parking systems is their potential to generate revenue for cities. By optimizing the allocation and pricing of parking spaces, these systems can help to increase the overall utilization of available parking and ensure that spaces are appropriately priced based on factors such as demand, location, and time of day. This can result in increased revenue from parking fees, as well as more efficient use of urban space [19].

Furthermore, smart parking systems can provide cities with valuable data on parking trends and user behavior, which can be used to inform future urban planning and transportation policies. For example, cities can use this data to identify areas where additional parking infrastructure is needed or where alternative transportation options should be promoted [20].

2.3.4 Employment Opportunities and Job Displacement

The implementation of IoT-based parking systems can have both positive and negative impacts on employment. On one hand, the development, installation, and maintenance of these systems can create new job opportunities in areas such as technology, engineering, and data analysis. Additionally, smart parking systems can indirectly contribute to job creation by supporting local businesses and promoting economic growth through reduced traffic congestion and improved accessibility [21].

On the other hand, the automation of parking management through IoT solutions may lead to job displacement for traditional parking attendants and other related roles. It is important for cities and system operators to consider the potential social and economic impacts of this job displacement and to develop strategies for supporting affected workers, such as

retraining programs or alternative employment opportunities within the smart parking industry [22].

2.4 Ethical Issues

2.4.1 Data Collection and Usage

As previously mentioned, IoT-based parking systems often rely on the collection and use of data to function effectively. While this data is essential for the operation of the system, there are ethical concerns surrounding its collection, storage, and usage. For example, there is the potential for data to be used in ways that violate users' privacy or to be shared with third parties without users' consent. Additionally, the aggregation and analysis of large amounts of data can raise questions about data ownership and control [23].

To address these ethical concerns, it is important for IoT parking system developers and operators to adhere to relevant data protection regulations and to implement best practices for data collection and usage. This includes obtaining user consent, anonymizing data where possible, and ensuring that data is only used for its intended purpose [24].

2.4.2 Equitable Access to Parking Resources

A key ethical consideration in the implementation of IoT-based parking systems is ensuring that access to parking resources is equitable and does not unfairly favor certain users or neighborhoods over others. For example, there is a risk that smart parking systems could prioritize spaces for users with higher willingness or ability to pay, potentially excluding lower-income drivers from accessing prime parking locations. Additionally, the deployment of IoT parking infrastructure may be concentrated in wealthier or more developed areas of the city, further exacerbating existing inequalities [25].

To promote equitable access to parking resources, it is important for cities to consider factors such as income, demographics, and existing transportation options when implementing IoT-based parking systems. This may involve adopting policies that prioritize spaces for certain user groups, such as disabled drivers or carpoolers, or implementing pricing schemes that account for users' ability to pay. Furthermore, cities should strive to deploy smart parking infrastructure in a balanced manner across neighborhoods, ensuring that all residents can benefit from these systems [26].

2.4.3 Environmental Impact

While IoT-based parking systems can contribute to a reduction in traffic congestion and emissions by making the parking process more efficient, there are also potential environmental trade-offs to consider. For instance, the increased convenience and availability of parking may encourage more people to drive, potentially leading to an overall increase in vehicle usage and associated environmental impacts [27].

To mitigate these potential negative environmental consequences, it is essential for cities to consider the broader context of their transportation systems and policies when implementing smart parking solutions. This may include promoting alternative transportation options, such as public transit, walking, and biking, or adopting policies that encourage the use of low-emission vehicles. Additionally, cities can leverage the data collected by IoT-based parking systems to inform their broader urban planning and environmental strategies [28].

2.4.4 Legal and Regulatory Compliance

The implementation of IoT-based parking systems raises various legal and regulatory issues, such as compliance with data protection laws, accessibility requirements, and zoning regulations. It is crucial for cities and system operators to ensure that these systems are developed and deployed in accordance with all applicable laws and regulations, both to protect users and to minimize the risk of legal challenges or penalties [29].

Collaboration between public and private stakeholders, as well as ongoing engagement with relevant regulatory bodies, can help to ensure that IoT parking solutions are compliant with existing laws and adaptable to any future changes in the regulatory landscape [30].

2.5 conclusion

In conclusion, IoT-based parking systems offer significant potential benefits in terms of improved efficiency, convenience, and sustainability. However, it is essential to carefully consider the social, economic, and ethical issues that arise with their implementation, in order to ensure that these systems contribute to more equitable, sustainable, and efficient urban environments.

References

[1] B. Evans, "Smart cities and the future of urban parking," Parking Today, 2017.

[2] A. Greenberg, "The high stakes of smart parking," Wired, 2016.

[3] M. Chui, et al., "Disruptive technologies: Advances that will transform life, business, and the global economy," McKinsey Global Institute, 2013.

[4] J. Holston, "Equity in the age of the smart city," Journal of Urban Technology, 2018.

[5] A. Khanna, et al., "Smart parking systems: A review," International Journal of Research in Engineering and Technology, 2014.

[6] A. Al-Hader, et al., "Smart city components for sustainable development," Journal of Information Technology Research, 2015.

[7] D. Shoup, "Cruising for parking," Transport Policy, 2006.

[8] M. Chester, et al., "Parking infrastructure: A constraint on or opportunity for urban redevelopment? A study of Los Angeles County parking supply and growth," Journal of the American Planning Association, 2010.

[9] A. Cavoukian, "Privacy by design: The definitive workshop," Identity in the Information Society, 2010.

[10] European Parliament and Council, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," Official Journal of the European Union, 2016.

[11] R. Roman, et al., "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, 2013.

[12] National Institute of Standards and Technology, "Framework for improving critical infrastructure cybersecurity," NIST, 2014.

[13] J. Holguín-Veras, et al., "Impacts of parking pricing on retail business in downtown: A case study," Journal of the Transportation Research Board, 2012.

[14] S. Zukin, "Gentrification, culture, and capital in the urban core," Annual Review of Sociology, 2017.

[15] N. Merat, et al., "Driver behavior in parking settings: An observational study," Transport Policy, 2016.

[16] E. Siemiatycki, "Public-private partnerships in Canada: Reflections on twenty years of practice," Canadian Public Administration, 2015.

[17] S. Shaheen, et al., "Smart parking management pilot program in San Francisco," Transportation Research Record, 2014.

[18] A. Gharaibeh, et al., "Smart cities: A survey on data management, security, and enabling technologies," IEEE Communications Surveys & Tutorials, 2017.

[19] M. Arnott, et al., "Parking economics, planning and policy," Transport Reviews, 2014.

[20] E. Welch, et al., "Open data, spatial enablement, and transport planning," Public Administration Review, 2015.

[21] J. Manyika, et al., "A future that works: Automation, employment, and productivity," McKinsey Global Institute, 2017.

[22] D. Acemoglu, et al., "Robots and jobs: Evidence from US labor markets," Journal of Political Economy, 2020.

[23] S. Mittal, et al., "Data management and security challenges in smart cities," International Journal of Engineering and Technology, 2016.

[24] T. Zarsky, "The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making," Science, Technology, & Human Values, 2016.

[25] K. Lucas, et al., "Transport poverty and its adverse social consequences," Proceedings of the Institution of Civil Engineers - Transport, 2016.

[26] E. Karner, et al., "Transportation planning and social justice," Journal of Planning Literature, 2020.

[27] M. Chester, et al., "Parking infrastructure: A constraint on or opportunity for urban redevelopment? A study of Los Angeles County parking supply and growth," Journal of the American Planning Association, 2010.

[28] D. Shoup, "The high cost of free parking," Journal of Planning Education and Research, 2005.

[29] M. Chourabi, et al., "Understanding smart cities: An integrative framework," Proceedings of the 45th Hawaii International Conference on System Sciences, 2012.

[30] S. Sadowski, et al., "Studying smart city governance: Analyzing the intersection of technology, policy, and administration," Public Administration Review, 2020.