Smart Parking System Using IoT

Date	**21-10-2023**
Team ID	**670**
Team Name	**Proj_223439_Team_2**
Project Name	**Smart Parking System Using IoT**

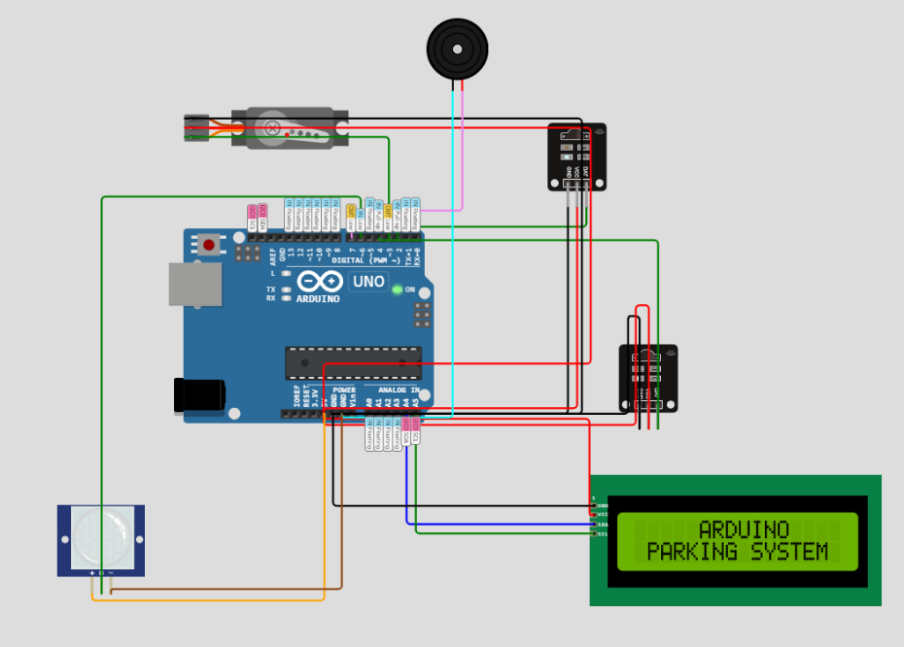
Source Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27, 16, 2);

#include <Servo.h>
Servo myservo1;

int IR1 = 2;
int IR2 = 4;
int SmokeDetectorPin = 6;
int BuzzerPin = 7;
int Slot = 4;

bool flag1 = false;
bool flag2 = false;

unsigned long lastLcdUpdate = 0;
unsigned long lcdUpdateInterval = 1000;

void setup () {
  lcd. begin (16, 2);
  lcd. backlight ();
  pinMode (IR1, INPUT);
  pinMode (IR2, INPUT);
  pinMode (SmokeDetectorPin, INPUT);
  pinMode (BuzzerPin, OUTPUT);

  myservo1.attach(3);
  myservo1.write(100);

  lcd. setCursor (0, 0);
  lcd.print ("    ARDUINO    ");
  lcd. setCursor (0, 1);
  lcd.print (" PARKING SYSTEM ");
  delay (2000);
  lcd. clear ();
```

```
  Serial.begin(9600);
}

void loop () {
  if (digitalRead (IR1) == LOW &&! flag1) {
    if (Slot > 0) {
      flag1 = true;
      if (! flag2) {
        myservo1.write(0);
        Slot--;
      }
    } else {
      displayMessage ("    SORRY :(    ", “Parking Full “);
    }
  }

  if (digitalRead (IR2) == LOW &&! flag2) {
    flag2 = true;
    if (! flag1) {
      myservo1.write(0);
      Slot++;
    }
  }

  if (flag1 && flag2) {
    delay (1000);
    myservo1.write(100);
    Serial.println("Servo returned to initial position.");
    flag1 = false;
    flag2 = false;
  }

  if (millis () - lastLcdUpdate >= lcdUpdateInterval) {
    updateLcdDisplay ();
    lastLcdUpdate = millis ();
  }
}

void updateLcdDisplay () {
  if (digitalRead (SmokeDetectorPin) == HIGH) {
    displayMessage ("   WARNING!   ", " Smoke Detected ");
    digitalWrite (BuzzerPin, HIGH);
  } else {
    displayMessage ("    WELCOME!    ", "Slot Left: " + String (Slot));
    digitalWrite (BuzzerPin, LOW);
  }
}
void displayMessage (const char *line1, const String &line2) {
```

```
  lcd. clear ();
  lcd. setCursor (0, 0);
  lcd.print(line1);
  lcd. setCursor (0, 1);
  lcd.print(line2);
}
```

Circuit Diagram:



Conclusion:

Smart parking systems, utilizing ESP32, STM32, and Arduino simulator technologies, offer an innovative approach to smart parking system. These systems leverage IoT and microcontroller capabilities to efficiently monitor and manage parking spaces in real time. Using sensors and data communication, they provide users with instant information on available parking spots, reducing search times and traffic congestion. The combination of ESP32, STM32, and Arduino simulator technologies enables cost-effective development and extensive testing of smart parking solutions. The systems can be easily scaled to accommodate more parking spaces. Integration with mobile apps and digital platforms enhances user convenience and optimizes the parking experience. The use of simulators during development ensures a robust, bug-free system before deployment, saving time and resources. These systems contribute to environmental sustainability by reducing emissions, as drivers spend less time searching for parking.