**Smart Parking System Project Concept**

Smart car parking systems using IOT, which include sensors and microcontrollers, can be available in each parking slot. Smart parking system using IOT has smart phones and other sensors added into an interconnected system to determine parking space or level and provide real-time feedback. It is accomplished by using cameras, counters on the doors or gates of parking lots, sensors embedded in the paved area of ​​individual parking lots, among other locations, depending on the deployment.

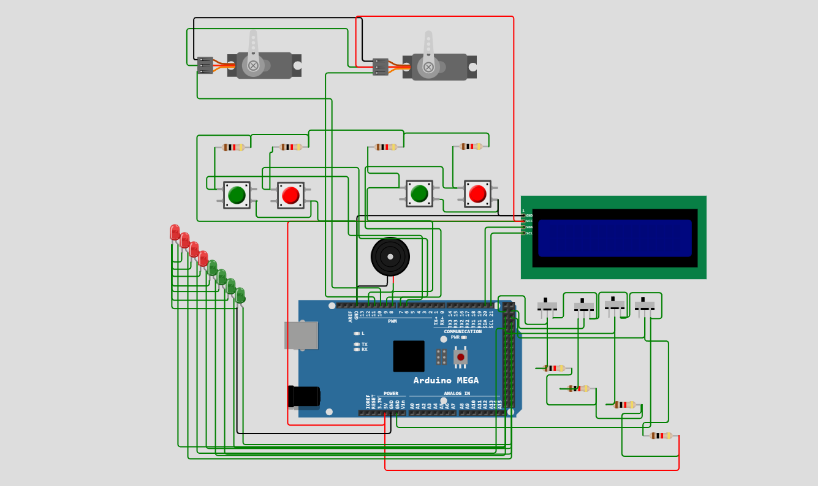
**Component Required**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Components** | **Quantity** |
| 1. | Arduino or Microcontroller | 1 |
| 2. | ICD 1602 | 1 |
| 3. | Servo Motors | 2 |
| 4. | Register | 8 |
| 5. | Buzzer | 1 |
| 6. | Push Button | 4 |
| 7. | LED Bulb | 8 |
| 8. | Slide Switch | 4 |
| 9. | Cables and Connectors | As Needed |

**Tools Required**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Tools** | **Examples** |
| 1. | Arduino IDE | Arduino IDE |
| 2. | CAD Software (Optional) | Autodesk Eagle. |
| 3. | Version Control System | GitHub, GitHub Desktop. |
| 4. | Project Management Software | ThingSpeak |
| 5. | Simulation Software (Optional) | Wokwi |
| 6. | Data Analysis Tools | Excel. |
| 7. | Communication Tools | Microsoft Teams, Discord, Zoom |

**Circuit Diagram of Smart Parking**



**Working principle**

The working principle of a typical car parking system involves several components and technologies:

Entry and Exit Gates: The system usually has entry and exit gates controlled by sensors or attendants. These gates regulate the flow of vehicles in and out of the parking facility.

Sensors: Sensors, such as ultrasonic or infrared sensors, are used to detect the presence of vehicles in parking spaces. These sensors send signals to a central control system.

Central Control System: This system processes the data from the sensors and manages the availability of parking spaces. It directs incoming vehicles to open spots and keeps track of occupied spaces.

Remote Management: parking systems are remotely monitored and managed through software, enabling operators to track occupancy, revenue, and perform maintenance tasks.

**Arduino Code**

#include <Servo.h>

#include <LiquidCrystal\_I2C.h>

uint8\_t indoor1 = 6;

uint8\_t indoor2 = 8;

uint8\_t outdoor1 = 7;

uint8\_t outdoor2 = 9;

uint8\_t buzzer = 12;

uint8\_t sncar[]={22,23,24,25};

uint8\_t outputgreen[]={26,27,28,29};

uint8\_t outputred []={30,31,32,33};

int count1 = 0;

boolean enablecount = false;

byte datax = 0b0000;

byte indexcar = 0x00;

LiquidCrystal\_I2C lcd(0x27,16,2);

Servo myservo1;

Servo myservo2;

void setup() {

Serial.begin(9600);

Serial.println("program start");

for(uint8\_t index = 0; index<4; index+=1){

pinMode(sncar[index],INPUT);

pinMode(outputgreen[index],OUTPUT);

pinMode(outputred[index],OUTPUT);

digitalWrite(sncar[index],LOW);

digitalWrite(outputgreen[index],LOW);

digitalWrite(outputred[index],LOW);

}

pinMode(indoor1,INPUT);

pinMode(indoor2,INPUT);

pinMode(outdoor1,INPUT);

pinMode(outdoor2,INPUT);

pinMode(buzzer,OUTPUT);

myservo1.attach(10);

myservo2.attach(11);

lcd.init(); // initialize the lcd

lcd.backlight();

lcd.clear();

delay(100);

}

void loop() {

/\* static unsigned long timer1 = millis();

if((millis()-timer1)>=1000){

timer1 = millis();

Serial.print("timeq1 = ");

Serial.print(timer1/1000);

Serial.println(" sec");

}\*/

if(enablecount==true){

// Serial.println("yeet");

count1+=1;

if(count1>=100){

myservo1.write(90);

enablecount = false;

Serial.println("ok");

count1=0;

}

}

for(byte i =0; i<4;i+=1){

bitWrite(datax,i,digitalRead(sncar[i]));

}

for(byte i =0; i<4;i+=1){

bitWrite(datax,i,digitalRead(sncar[i]));

//Serial.print(bitRead(datax,i),BIN);

indexcar = indexcar+!bitRead(datax,i);

if(i==3){

//Serial.println("");

lcd.setCursor(0,0);

lcd.print("Total : "+String(indexcar)+" car");

lcd.setCursor(0,1);

if(indexcar==4){

lcd.print(String(!bitRead(datax,0))+" "+String(!bitRead(datax,1))+" "+String(!bitRead(datax,2))+" "+String(!bitRead(datax,3))+" "+"FULL CAR");

}

else{

lcd.print(String(!bitRead(datax,0))+" "+String(!bitRead(datax,1))+" "+String(!bitRead(datax,2))+" "+String(!bitRead(datax,3))+" "+"Emptycar");

}

indexcar = 0;

}

}

if(digitalRead(indoor1)==0 && datax!= 0b0000){

myservo1.write(180);

count1=0;

}

if(digitalRead(outdoor1)==0 || datax == 0b0000 ){

enablecount = true;

}

if(digitalRead(indoor2)==0){

myservo2.write(180);

}

if(digitalRead(outdoor2)==0){

myservo2.write(90);

}

for(uint8\_t index = 0; index<4; index+=1){

digitalWrite(outputgreen[index],!digitalRead(sncar[index]));

digitalWrite(outputred[index],digitalRead(sncar[index]));

}

delay(10);

}