**Smart Parking System**

In today's increasingly crowded urban environments, efficient and intelligent parking solutions are more critical than ever. To address this challenge, a Parking System using Arduino Uno offers a cost-effective and innovative approach.

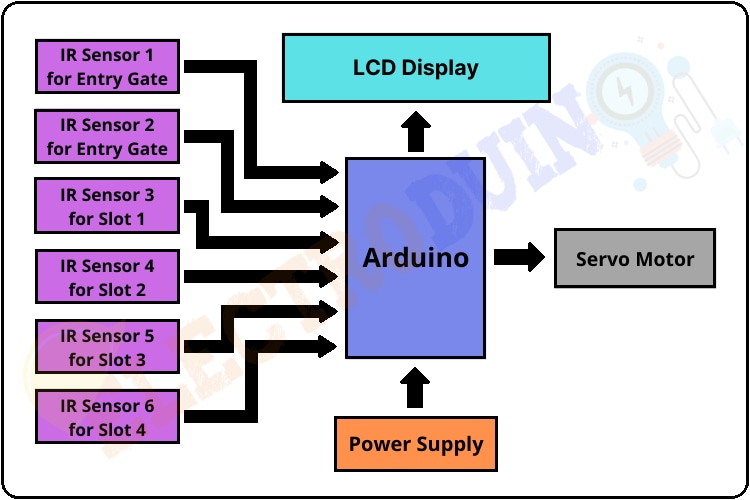
This system leverages the power of Arduino Uno, a versatile microcontroller platform, to create a smart and automated parking management solution.

The Arduino Uno is a popular choice for such projects due to its ease of use, affordability, and a vibrant community of developers. By combining it with various sensors and actuators, we can design a Parking System that not only simplifies parking for users but also enhances overall traffic management**.**

Key components of this Parking System typically include ultrasonic sensors to detect vehicle presence, a user interface for drivers, and a centralized control unit that manages parking spaces, communicates with users, and collects data for analysis.

The system can provide real-time information on available parking spots, simplify the entry and exit process, and even enable cashless payment option

**Block Diagram:**

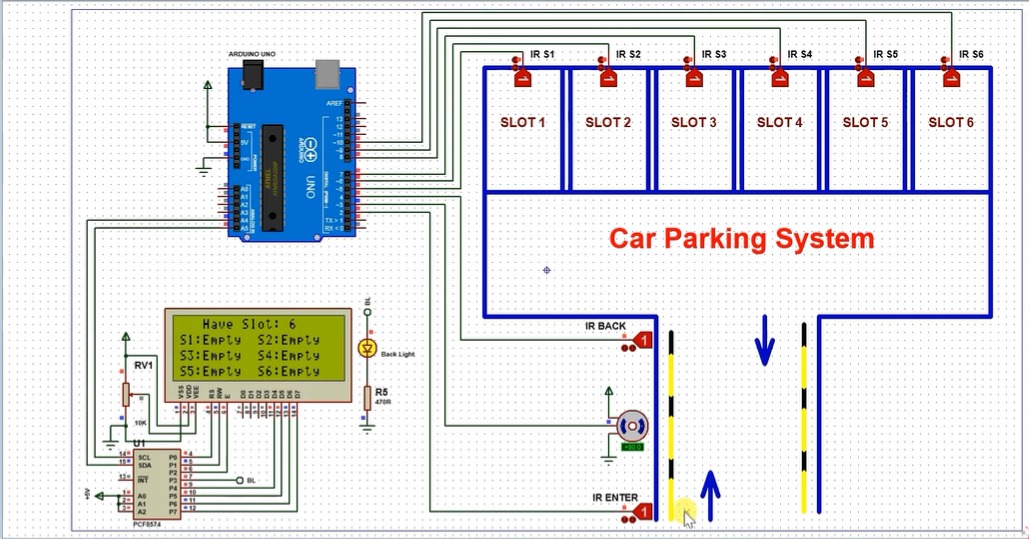


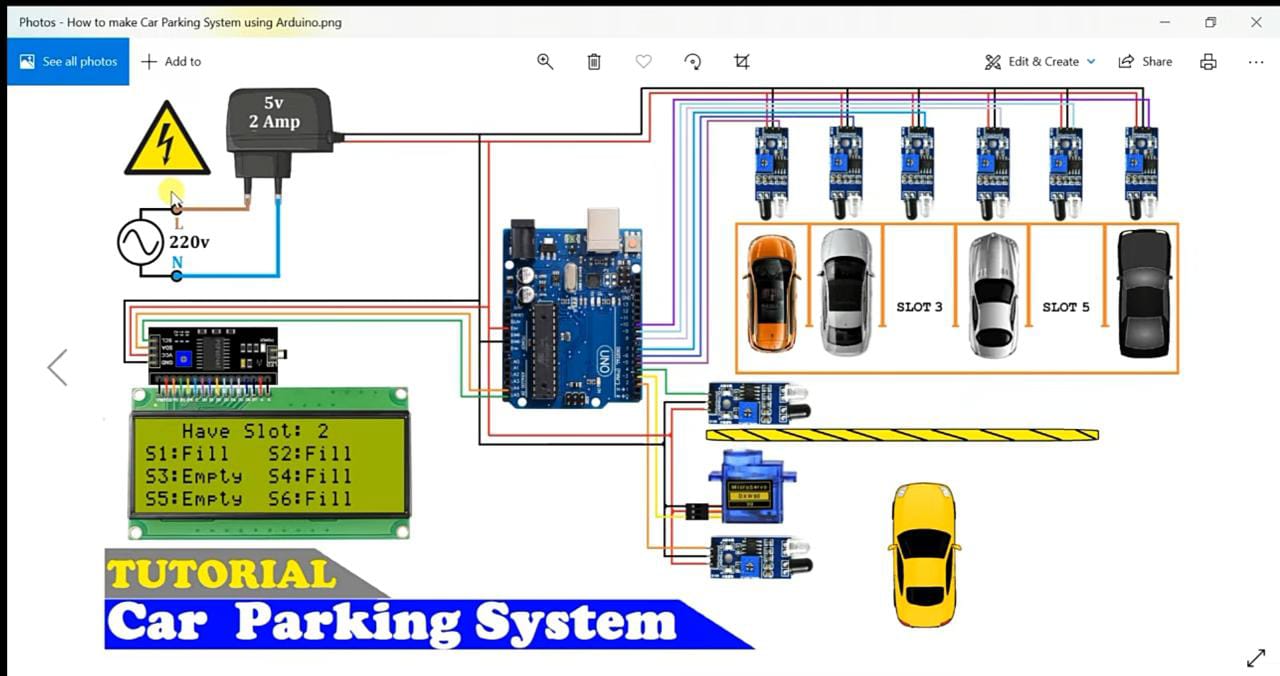
**Project Ojective:**

The objective of a Smart Parking System using Arduino Uno is to create an intelligent, automated, and user-friendly solution for managing parking spaces. The project aims to address several key goals and objectives, including:

* **Efficient Space Management**: Optimize the utilization of available parking spaces by providing real-time information on spot availability, reducing the time spent searching for parking, and ensuring a more orderly and space-efficient parking environment.
* **Enhanced User Experience**: Improve the parking experience for drivers by offering a convenient, user-friendly interface for finding andreserving parking spots, as well as simplifying the entry and exit processes.
* **Reduction in Traffic Congestion**: Minimize traffic congestion and pollution caused by vehicles circling in search of parking spaces. This project aims to enhance traffic flow and reduce emissions by directing drivers to available parking spots.
* **Cost-Efficiency**: Implement a cost-effective solution that can be easily deployed in various locations, helpingbusinesses and municipalities make the most of their parking infrastructure.
* **Safety and Security**: Ensure the safety of vehicles and drivers by implementing features such as collision avoidance and monitoring of parking areas to deter theft and vandalism.
* **Data Collection and Analysis**: Gather data on parking space usage, user behavior, and traffic patterns for analysis. This data can be used to make informed decisions about expanding parking facilities, pricing structures, and traffic management strategies.
* **Scalability**: Design a system that can be easily scaled and adapted to accommodate various parking locations, including open lots, multi-level garages, and on-street parking.
* **Integration with IoT and Smart City Initiatives**: Enable the Smart Parking System to integrate with broader smart city initiatives, connecting it to traffic management systems, public transportation, and other urban infrastructure for more comprehensive urban planning.
* **Sustainability**: Contribute to sustainability goals by reducing fuel consumption and emissions through more efficient parking, and by promoting alternative transportation options when parking is limited.

**Simulation output:**





**Code:**

#include <Servo.h> //includes the servo library

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

Servo myservo;

#define ir\_enter 2

#define ir\_back 4

#define ir\_car1 5

#define ir\_car2 6

#define ir\_car3 7

#define ir\_car4 8

#define ir\_car5 9

#define ir\_car6 10

int S1=0, S2=0, S3=0, S4=0, S5=0, S6=0;

int flag1=0, flag2=0;

int slot = 6;

void setup(){

Serial.begin(9600);

pinMode(ir\_car1, INPUT);

pinMode(ir\_car2, INPUT);

pinMode(ir\_car3, INPUT);

pinMode(ir\_car4, INPUT);

pinMode(ir\_car5, INPUT);

pinMode(ir\_car6, INPUT);

pinMode(ir\_enter, INPUT);

pinMode(ir\_back, INPUT);

myservo.attach(3);

myservo.write(90);

lcd.begin(20, 4);

lcd.setCursor (0,1);

lcd.print(" Car parking ");

lcd.setCursor (0,2);

lcd.print(" System ");

delay (2000);

lcd.clear();

Read\_Sensor();

int total = S1+S2+S3+S4+S5+S6;

slot = slot-total;

}

void loop(){

Read\_Sensor();

lcd.setCursor (0,0);

lcd.print(" Have Slot: ");

lcd.print(slot);

lcd.print(" ");

lcd.setCursor (0,1);

if(S1==1){lcd.print("S1:Fill ");}

else{lcd.print("S1:Empty");}

lcd.setCursor (10,1);

if(S2==1){lcd.print("S2:Fill ");}

else{lcd.print("S2:Empty");}

lcd.setCursor (0,2);

if(S3==1){lcd.print("S3:Fill ");}

else{lcd.print("S3:Empty");}

lcd.setCursor (10,2);

if(S4==1){lcd.print("S4:Fill ");}

else{lcd.print("S4:Empty");}

lcd.setCursor (0,3);

if(S5==1){lcd.print("S5:Fill ");}

else{lcd.print("S5:Empty");}

lcd.setCursor (10,3);

if(S6==1){lcd.print("S6:Fill ");}

else{lcd.print("S6:Empty");}

if(digitalRead (ir\_enter) == 0 && flag1==0){

if(slot>0){flag1=1;

if(flag2==0){myservo.write(180); slot = slot-1;}

}else{

lcd.setCursor (0,0);

lcd.print(" Sorry Parking Full ");

delay(1500);

}

}

if(digitalRead (ir\_back) == 0 && flag2==0){flag2=1;

if(flag1==0){myservo.write(180); slot = slot+1;}

}

if(flag1==1 && flag2==1){

delay (1000);

myservo.write(90);

flag1=0, flag2=0;

}

delay(1);

}

void Read\_Sensor(){

S1=0, S2=0, S3=0, S4=0, S5=0, S6=0;

if(digitalRead(ir\_car1) == 0){S1=1;}

if(digitalRead(ir\_car2) == 0){S2=1;}

if(digitalRead(ir\_car3) == 0){S3=1;}

if(digitalRead(ir\_car4) == 0){S4=1;}

if(digitalRead(ir\_car5) == 0){S5=1;}

if(digitalRead(ir\_car6) == 0){S6=1;}

}

**PREPARED BY,**

1. **N.ABINAYA**
2. **T.PAVITHRA**
3. **E.ABARNA LAKSHMI**
4. **K.SHABARNA**
5. **A.DIVYA**