**SMART PARKING**



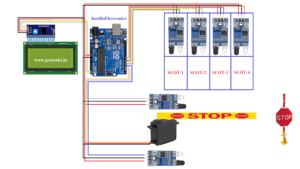
ABSTRACT

* The idea of creating a smart city is now becoming possible with the emergence of the IOT. One of the key issue that smart cities relayed to are car parking facilities and traffic management system
* Parking sport is always difficult for drivers. It tends to become harder with the increasing number of private car users.
* Main aim of this system is to avoid the problems which are occur during the parking of the vehicle.
* RFID based car parking uses a microcontroller along with sensing circuits monitoring entry and exit of cars.
* An H-bridge arrangement operates the entry and exit boom motors operating clockwise and anticlockwise for opening and closing. Upon every entry of a car the parking availability gets reduced by one number while every exit the number increases.
* The use of the IOT is done by connecting sensor devices scattered in each parking lot to the internet network, and then, data from each sensor will be stored on the server.
* So that it can facilitate the communication between the sensor device and the server to store parking lot data into a database that users will access to obtain information about the parking lot.
* A prototype for a parking mall is developed and tested. The functionality of the system is also tested and the results are encouraging.

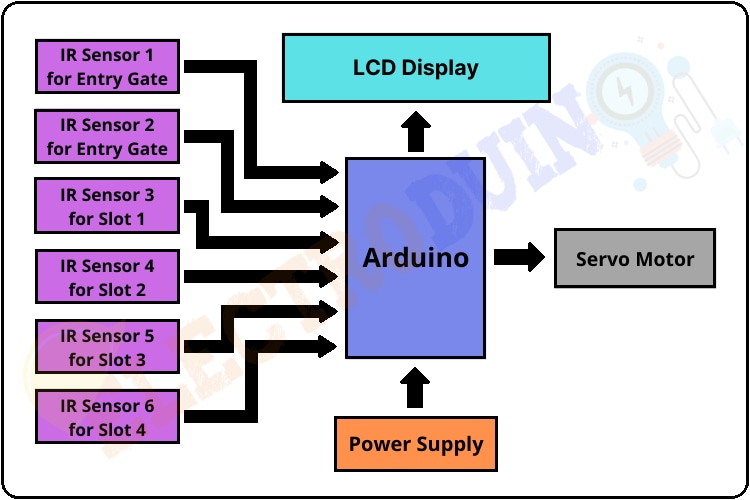
**Components Needed**

1. *Arduino Uno*
2. *IR Sensor*
3. *20×4 LCD Display with I2C*
4. *MG995 Servo Motor*
5. *Wire*
6. *Zero PCB*
7. *9V power supply*

**CIRCUIT DIAGRAM**



***BLOCK DIAGRAM***



**PROGRAME**

//Prateek

//https://justdoelectronics.com

//https://www.youtube.com/@JustDoElectronics

#include <Servo.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4);

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

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

int flag1 = 0, flag2 = 0;

int slot = 4;

void setup() {

Serial.begin(9600);

pinMode(ir\_car1, INPUT);

pinMode(ir\_car2, INPUT);

pinMode(ir\_car3, INPUT);

pinMode(ir\_car4, INPUT);

pinMode(ir\_enter, INPUT);

pinMode(ir\_back, INPUT);

myservo.attach(3);

myservo.write(90);

lcd.init();

lcd.backlight();

lcd.setCursor(0, 1);

lcd.print(" Hi Welcome To ");

lcd.setCursor(0, 2);

lcd.print(" JustDoElectronics");

delay(5000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" Today's Project ");

lcd.setCursor(0, 1);

lcd.print(" Car Parking ");

lcd.setCursor(0, 2);

lcd.print(" System ");

delay(5000);

lcd.clear();

Read\_Sensor();

int total = S1 + S2 + S3 + S4;

slot = slot - total;

}

void loop() {

Read\_Sensor();

lcd.setCursor(0, 0);

lcd.print(" Available Slot: ");

lcd.print(slot);

lcd.print(" ");

lcd.setCursor(0, 1);

if (S1 == 1) {

lcd.print("S1:Full ");

} else {

lcd.print("S1:Empty");

}

lcd.setCursor(11, 1);

if (S2 == 1) {

lcd.print("S2:Full ");

} else {

lcd.print("S2:Empty");

}

lcd.setCursor(0, 2);

if (S3 == 1) {

lcd.print("S3:Full ");

} else {

lcd.print("S3:Empty");

}

lcd.setCursor(11, 2);

if (S4 == 1) {

lcd.print("S4:Full ");

} else {

lcd.print("S4: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;

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;

}

}

***CONCLUSION***

*We are working with a python code to implement the above setup. The python code will be uploaded in the upcoming phases of the project.*

*THANK*

*YOU!!!*