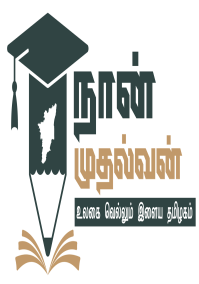
****  **SMART PARKING**

**PROJECT REPORT**

**Submitted by**

**PREM KUMAR . S 811221106024**

**VEERA KUMAR . R 811221106035**

**VICTORRAJ . S 811221106037**

**VISHAL . V 811221106040**

**OM PRAKASH . R 811221106301**

**BACHELER OF ENGINEERING**

**IN**

**ELECTRONICS AND COMUNICATION ENGINEERING**

**INDRA GANESAN COLLEGE OF ENGINEERING,TRICHY**

**NAN MUTHALVAN**

INDEX

ABSTRACT - 2

LITERATURE REVIEW - 2

CIRCUIT DIAGRAM - 4

HARDWARE & SOFTWARE COMPONENTS - 5

PROGRAME - 6

OUTPUT - 11

CONCLUSION - 12

FUTURE SCOPE - 12

REFERENCE - 13

**1**

**ABSTRACT**

The idea of creating a smart city is now becoming possible with the emergence of the IOT. One of the key issues 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.

**LITERATURE REVIEW**

**1.“Smart Parking using IOT”**

Anusha,arshitha;2019

**ABSTRACT**

It allows drivers to reverse parking spaces in advance through a mobile application or web interface. To provide secured parking space for the users in metropolitan cities.

**LIMITATIONS**

The user needs to park his vehicle in the slot that is allocated to him.

**2**

**2.“Smart Parking System”**

Namith,Elakiya;2019

**ABSTRACT:**

The goal is to automate and reduce the time spend manually searching for a parking space.To reduce time and increase efficiency of the current parking management system.

**LIMITATIONS:**

The user should be sensible enough to know that he should vacate the parking space at his specified departure time.The system fails if the sensors stop working

**3.“Smart parking using IOT”**

Supreeth,ragavendra prasath;2019

**ABSTRACT:**

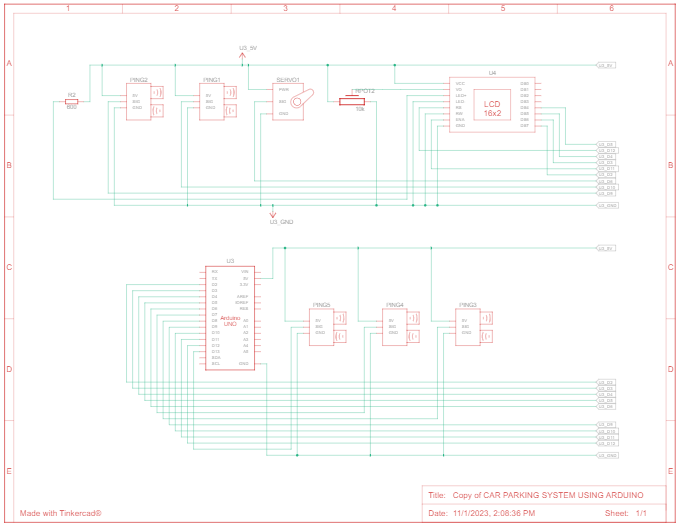
This prototype aims to optimize parking space utilization, reduce human error, and enhance overall parking efficiency. To offer save and secure parking slots with in limited area.

**LIMITATIONS:**

Availability of the space could be found only after the car enters the parking lot.

**3**

**CIRCUIT DIAGRAM**



**4**

**HARDWARE COMPONENT:**



**SOFTWARE COMPONENTS:**

* **TINKERCAD**

Tinkercad is a free web app for 3D design, electronics, and coding, trusted by over 50 million people around the world. Build STEM confidence by bringing project-based learning to the classr

**5**

**PROGRAM:**

#include <Servo.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define t1 10

#define t2 9

#define t3 8

#define t4 7

#define t5 13

Servo myservo;

int distanceThreshold = 100;

int parkingAvailable = 3;

int barrierState = 0;

void setup() {

lcd.begin(16,2);

lcd.setCursor(0,0);

Serial.begin (9600);

myservo.attach(6);

myservo.write(0);

}

long readDistance(int triggerPin, int echoPin)

{

pinMode(triggerPin, OUTPUT);

digitalWrite(triggerPin, LOW);

delayMicroseconds(2);

digitalWrite(triggerPin, HIGH);

delayMicroseconds(10);

digitalWrite(triggerPin, LOW);

pinMode(echoPin, INPUT);

return pulseIn(echoPin, HIGH);

}

**6**

void loop()

{

float d1 = 0.01723 \* readDistance(t1, t1);

float d2 = 0.01723 \* readDistance(t2, t2);

float d3 = 0.01723 \* readDistance(t3, t3);

float d4 = 0.01723 \* readDistance(t4, t4);

float d5 = 0.01723 \* readDistance(t5, t5);

//get all sensor readings

Serial.println("d1 = " + String(d1) + "cm");

Serial.println("d2 = " + String(d2) + "cm");

Serial.println("d3 = " + String(d3) + "cm");

Serial.println("d4 = " + String(d4) + "cm");

Serial.println("d5 = " + String(d5) + "cm");

//display all sensor distance on serial monitor

/\* For below command

barierState is used to ensure that the barrier either is closed or open for enter or exit

So while barrierState is:

0 barrier is closed

-1 barrier is open for enter

1 barrier is open for exit

-2 barrier is closed after the vehicle passed the barrier gate (enter)

2 barrier is closed after the vehicle passed the barrier gate (exit)

after it reached to -2 or 2 value, it will reset back to 0 when there is no vehicle detected

by two sensor.

For parkingAvailable, we set to 3 as we have 3 parking lot only

so it will automatically update while the barrier is open.

Therefore, the barrier will not be open while there is no more parking lot available.

\*/

**7**

if (barrierState == 0)

{

if (d4<100 && d5>=100 && parkingAvailable>0)

{

parkingAvailable -= 1;

barrierState = -1;

myservo.write(90);

}

if (d4>=100 && d5<100 && parkingAvailable<3)

{

parkingAvailable += 1;

barrierState = 1;

myservo.write(90);

}

}

else if (barrierState == -1)

{

if (d4>=100 && d5<100)

{

barrierState = -2;

myservo.write(0);

}

}

else if (barrierState == 1)

{

if (d5>=100 && d4<100)

{

barrierState = 2;

myservo.write(0);

}

}

**8**

else if (barrierState == -2)

{

if (d5>=100)

{

barrierState = 0;

}

}

else if (barrierState == 2)

{

if (d4>=100)

{

barrierState = 0;

}

}

/\*

The command below is used to print out the information on the LCD Screen

For the first row of the LCD Screen, it will show the number of parking lot available

While it full, it will show Parking Full

For the second row of the LCD Screen, it will show which parking lot is still empty

This allow users to know which parking lot is still available without wasting time

to find an empty parking lot.

\*/

lcd.setCursor(0,0);

if (parkingAvailable == 0)

{

lcd.print("Parking Full ");

}

else

{

lcd.print("Parking left ");

**9**

lcd.print(parkingAvailable);

}

if (d1>100 & d2>100 & d3>100)

{

lcd.setCursor(0,1);

lcd.print("Slot 1 2 3 Free");

delay(500);

}

else if((d1>100 & d2>100)|(d2>100 & d3>100)|(d3>100 & d1>100))

{

lcd.setCursor(0,1);

if(d1>100 & d2>100)

lcd.print("Slot 1 & 2 Free");

else if(d1>100 & d3>100)

lcd.print("Slot 1 & 3 Free");

else

lcd.print("Slot 2 & 3 Free");

}

else if(d1<100 & d2<100 & d3<100)

{

lcd.setCursor(0,1);

lcd.print("Parking Full ");

}

else if((d1<100 & d2<100)|(d2<100 & d3<100)|(d3<100 & d1<100))

{

lcd.setCursor(0,1);

if(d1>100)

lcd.print("Slot 1 is Free ");

else if (d2>100)

lcd.print("Slot 2 is Free ");

else

**10**

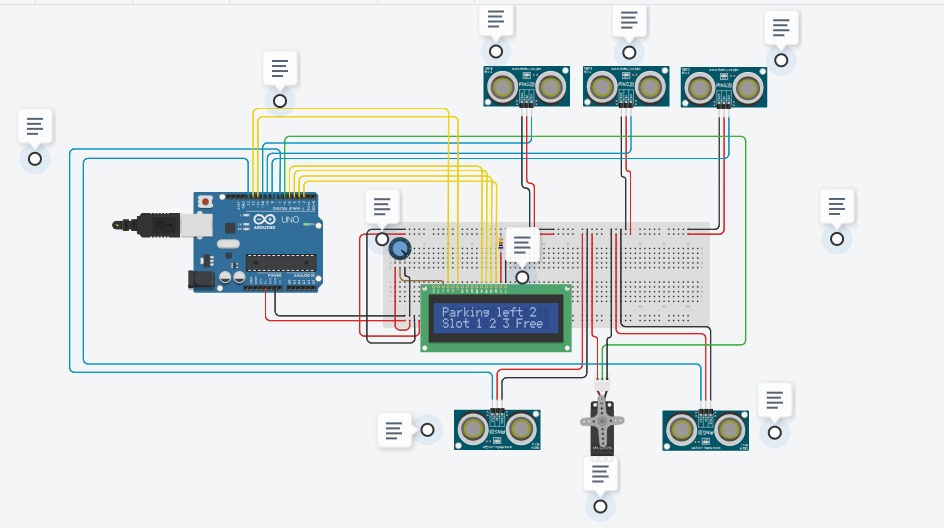
lcd.print("Slot 3 is Free ");

}

delay(100);

}

**OUTPUT:**



**11**

### **CONCLUSION**

By integrating an I2C 20×4 LCD display, an MG955 servo motor, an IR sensor, and an Arduino, we can create a car parking system that automates the process of parking and enhances the overall user experience. Such a system not only provides real-time information to drivers but also improves parking space management and reduces the need for manual intervention.

With further enhancements, this system can be expanded to handle multiple parking spaces, integrate with mobile applications, or incorporate advanced features like automatic payment systems.

**FUTURE SCOPE :**

The Smart parking system based on Slot booking is implemented, using the Android application. Using the slot allocation method we can book our own cheapest parking slot. It is an efficient one for solving parking problems, which overcomes the traffic congestion also provides automated billing process. This work could be further extended as a fully automated system using multilayer parking method. Safety measures such as tracing the vehicle number face recognition of the drivers so as to avoid theft & automatic billing process can also be designed. We plan to expand the tests on the real time environment where the users can have the “Smart Parking” system in their handheld devices.

**12**

**REFERENCE:**

1. IoT based Smart Parking System, IEEE ICICCS 2021, Yash Agarwal, Umang Shah, Puru Jain.

2. IoT-Based Shared Community Transportation System Using eBikes, IEEE 2021, A. R. Al-Ali, Raafat Aburukba.

3. An IoT based Smart Outdoor Parking System, ICACCS 2021, Gokul Krishna. S, Harsheetha.

4. A Smart, Efficient, and Reliable Parking Surveillance System With Edge Artificial Intelligence on IoT Devices, IEEE 2021, Ruimin Ke, Yinhai Wang.

5. IOT Based “No Parking Notifier System”, I-SMAC 2018, Sushanth.G, Dr.Sujatha.S .

6. IoT-Ready Energy Autonomous Parking Sensor Device, IEEE INTERNET OF THINGS JOURNAL,2021, Alfiero Leoni, Toni Perkovic.

7. IOT Based Smart Vehicle Parking System Using RFID, ICCCI -2021, Mohan P.Thakre, Nishant P.Matale .

8. IoT in Vehicle Presence Detection of Smart Parking System, IEEE Eurasia 2020, Yi-Yun Chu, Kai Hao Liu.

9. IoT based Smart Parking System: A Proposed Algorithm and Model, IEEE ICCSE 2020, Rahul, Padmaja Patel.

10. IoTRec: The IoT Recommender for Smart Parking System, IEEE TETC 2020, Yasir Saleem, Pablo Sotres.

11. A Millimeter Wave Dual-lens Antenna for IoT Based Smart Parking Radar System, IEEE JIOT 2020, Zhanghua Cai, Yantao Zhou.

12. IOT Based Car Parking, IEEE 2020, P.S.Patil, Dr.S.K.Padaganur.

13. IoT Parking Apps with Car Plate Recognition for Smart City using Node Red, ICICS 2020, Hannah Sofian.

14. An UAV assisted Multi-Sensor based Smart Parking System, IEEE 2020, Pritom Gogoi, Joy Dutta. 15. Smart Parking System Using MobileNet and Single Shot Detection Algorithm and IoT, ICECE 2020, Galib Ibne Haidar, Hasin Ishraq Reefat.

**13**