

## **Smart Parking System Using Arduino**

### **A Project Report FALL SEM 2022-23**

#### **ECS1001**

(Engineering Clinics - Arduino using Embedded C - Project)

#### **Submitted by group id-220137**

Gaurav Sharma	(20BCE7443)
Kommasani Lakshmi sai sri	(20BCE7470)
Anandam vishal Aryan	(20BCR7116)
Jyesta laasya Priya	(20BCN7124)
Veerendra Singamsetty.	(20BCR7117)
Athuluri sai ram	(20BCD7116)

Under the guidance of :

**Prof.Sunil kumar singh**



**VIT-AP UNIVERSITY**

**AMRAVATI**

**ANDHRA PRADESH**

**2022**

## Abstract

This project explains the design of **Arduino based smart car parking system**. Authorization of car number plate in the database is the basic rule used to park a vehicle in a parking place. Authorization database will be given to each user, which carries the vehicle number or other details. If the **faculty/student** is authorized and space is available in the parking area, then the parking gate opens, and the user is allowed to park the vehicle in parking place according to their category and availability of parking slots else the user is not allowed to enter the parking even if the user is authorized .

It manages the time, also provides security to a vehicle and an unauthorized user is not allowed to enter into a parking place whereas they are given separate parking slots. It helps to park vehicle in empty slot parking also as it will display which slot has free space.

## Index

Sl.No	Sections	Page number
1	Introdcution	5
2	Backgroud	6
3	Problem Defination	6
4	Objectives	6
5	Procedure/Methodology	7-8
6	Result and Discussion	9
7	Conclusion and Future Scope	9
8	References	10
9	Appendix	10-16

**List of Figures**

Sl.No	Figure	Page Number
1	Circuit Diagram	7
2	Prototype	8

## Introduction

In the universities/colleges, finding place to park the car is one of disturbing problem for faculties, students and visitors. Especially when you visit campus during admissions or fests or daily times etc. The owner of the respective vehicle waste time and fuel looking for a slot to park their cars even within the parking area itself. This will affect the vehicle owner's mood as well as pollution of the surrounding environment during the search for the parking slot to park. In this project we developed and designed a smart parking system that can solve those problems efficiently. Furthermore, the developed system provides a new identification mechanism of the parking slots based on the owner of the vehicle like faculty, students to their respective parking areas. In the last few years there have been many studies that aim to reduce the car parking problems and to make it more easier and human less.

Prof. Sunil Kumar Singh, has proposed smart car parking system using Arduino Uno. Effective smart parking technologies are developed to overcome the existing problems by using of sensor network and providing real time data analysis from the sensor. The system uses an Arduino based application to let the vehicle look for an empty parking slot within their respective parking areas detecting the car number and given car number database information in Arduino. The system defines the time that the owner needs to park his car at the parking. It uses a plate recognition technology at the entrance barrier of the parking area and compares the given database of the vehicle, to allow authorized driver enter the parking area. The system detects the empty slots within the parking by using IR sensors placed at the slots of the parking and helps the vehicle owner to find parking in unfamiliar place. The present available technology in parking systems has limitations to give a parking access to authorized vehicles and the current available system does not provide information to the driver about available parking slots.

To overcome these issues, this project attempts to develop a smart parking system with following features:

- An access mechanism only for authorized vehicles to park their cars.
- The guidance mechanism for vehicles about available parking slots.
- Guides faculties, students to their respective available parking slot.
- Accurately sense and predict spot/vehicle occupancy in real-time.
- Optimize Parking Space Usage.
- Helps in the free flow of traffic at the campus.
- Smart Parking plays an important role in creating better urban environment by reducing the emission of CO<sub>2</sub> and other pollutants.

## Background

Based on the prevailing science, there is a solution to every hassle that we're going through in our everyday lives.

Smart parking system is an efficient way to accurately sense and predict spot/vehicle occupancy in real-time. It will provide guidance faculties, students and visitors to their respective available parking slot. The main purpose of this project is to optimize parking space usage and help with the free flow of traffic in the city.

## Problem Definition

The current parking system has inadequate parking spaces, causing activities to be stifled and faculty, students and visitors to be dissatisfied. They must wait a long time at the entrance gate during busy hours before finding an empty parking space. As a result, users will waste time and energy looking for a spare area.

Last but not least, the main problem with the parking system is the information about the available parking places.

Smart parking system can be used to manage and reduce parking search traffic on the campus.

## Objectives

The aim of this project, as the name suggests, is that the parking system designed for vehicles to provide them a parking space in an efficient manner.

During the execution of the project of the Smart parking system, we have had come up with detailed and transparent aims and objectives.

1. The main motivation of this project is to manage the time of faculty and student's vehicles searching for parking.
2. Reduce the wastage of time and energy finding free space across the parking surface when they need to park their car which requires a good amount of fuel. By providing information about the space availability before entering the parking.
3. This automated system will only allow entry to authorized vehicles and no entry will be provided if all the parking slots are full even if the vehicle is authorized.
4. Therefore reducing the traffic in front of parking slot and help maintain college environment.

## Procedure

## Materials Required

- Arduino uno
- IR sensors(x10)
- 16x2 LCD display
- 20x4 LCD display
- Servo-Motor
- Breadboard ,
- Jumper wires.

## Circuit Design

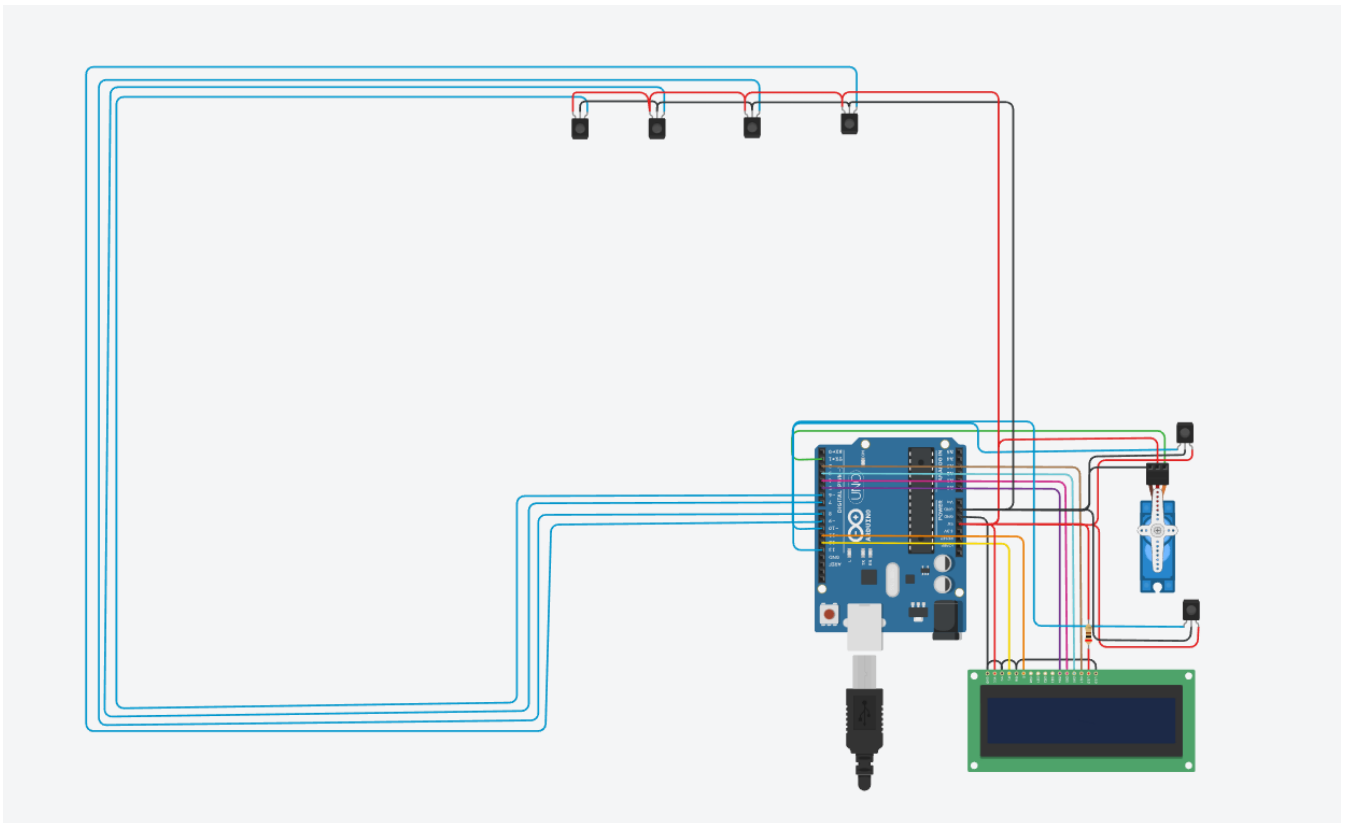


Figure 1.circuit design

## Methodology

### 1. License Plate Image Acquisition

At the entrance, when the car comes the license plate's image acquisition is done with the help of system(laptop) web-cam .After acquisition and processing the image it the algorithm indentifies whether it is a faculties/students/visitors car. Hence displaying the detials on LCD (16x2) and providing proper dirctions to vehicles.

### 2. Functionality of parking system

- Whenever the vehicle arrives near the enterance ,it is detected by IR sensors , the sensor send signals to servo-motor to open the enterance gate. Similar prodedure is used for closing the gate .
- The parking slot cotains 6 parking slots and 2 charging slots for electric vehicles charging each slot has one IR-sensor to provide information about the slot availability.The slot availability is displayed with the help of 20x4 LCD .
- The LCD display shows actual space/slots availabilty in the parking area .In case of no parking slot availability , the entrance gate will not and a message about the same will be displayed on LCD.

This is how the whole process works automatically without any human effort.

### 3. Prototype

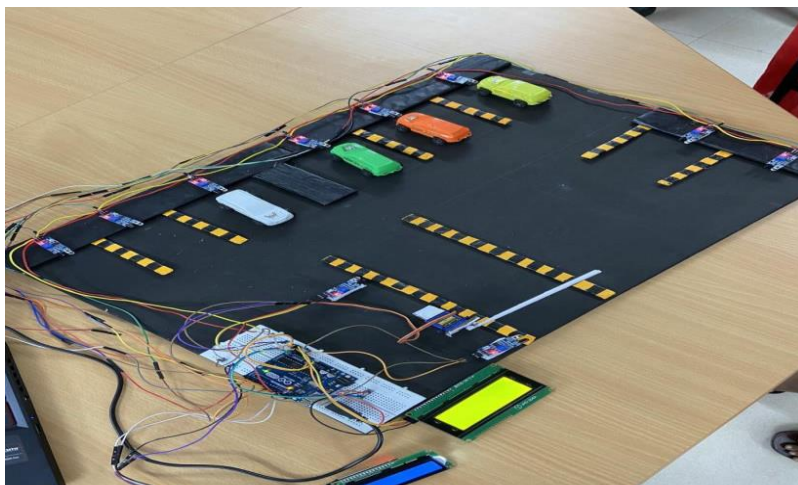


Figure 2: Smart Parking System



## Result And Discussion

- It guarantees snappy and computerized parking and simple recovery of vehicles.
- Up to 6 cars can be effectively and securely parked in the outlined model.
- 2 electric charging port is provided for the people who came to charge their vehicle.
- This parking works with minimum interaction required from humans, from entry to parking slot allocation is automated.
- Low support levels are required by the framework.
- Sensors utilized have high affectability and are anything but difficult to deal with.
- Cordial reorientation of cars for driving in and out, optimised to control traffic near parking places.

## Conclusion And Future Scope:

Smart parking system is a systematic and efficient way to provide parking slots to authorized users.

The project detects the empty slots and helps the drivers (students/faculty) to find parking space. The average waiting time of users for parking their vehicles is effectively reduced in this system. Our preliminary test results show that the performance of the Arduino UNO based system can effectively satisfy the needs and requirements of existing car parking hassles thereby minimizing the time consumed to find vacant parking lot. This smart parking system provides better performance. When a car enters the parking area, the camera will scan the number plate and recognizes who they are and the LCD display shows directions to the empty slot if slot is free, if slot is not free the LCD display shows the slot is full to faculty/students.

Our smart Parking system involves the use of low-cost sensors, real-time data and applications that allow users to monitor available and unavailable parking spots and shows the direction of empty parking slots. The goal is to automate and decrease time spent manually searching for the optimal parking floor, spot and even lot. User (student/faculty/visitors) can find the best spot available, saving time, resources, effort. This smart parking system will have more safety, because parking lot employees and security guards contain real-time lot data that can help prevent parking violations and suspicious activity. License plate recognition cameras can gather pertinent footage. And this system reduces the traffic flow increases as fewer cars are required to drive around in search of an open parking space.

## REFERENCES:

<https://www.sciencedirect.com/science/article/pii/S2405844021011531>

[https://www.researchgate.net/publication/318486476\\_Smart\\_Car\\_Parking\\_System\\_using\\_Arduino\\_UNO](https://www.researchgate.net/publication/318486476_Smart_Car_Parking_System_using_Arduino_UNO)

[https://www.sciencedirect.com/science/article/pii/S1877705811017474?ref=pdf\\_download&fr=RR-2&rr=771e77e91a9f9371](https://www.sciencedirect.com/science/article/pii/S1877705811017474?ref=pdf_download&fr=RR-2&rr=771e77e91a9f9371)

## Appendix

### Ardiono Code :

```
Sendin/Wire.h
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <Servo.h> //includes the servo library
4
5
6  // Set the LCD address to 0x27 for a 16 chars and 2 line display
7  LiquidCrystal_I2C lcd (0x27, 20, 4);
8  LiquidCrystal_I2C lcd1(0x26, 16, 2);
9
10 Servo myservo;
11
12 char mtlb_data;
13 int ir_enter = 13;
14 int ir_back = 12;
15
16 int ir_car1 =11;
17 int ir_car2 =10;
18 int ir_car3 =9;
19 int ir_car4 =8;
20 int ir_car5 =7;
21 int ir_car6 =6;
22 int c2 =5;
23 int c1 =4;
24 int slot=7;
25 int flag1=0;
26 int flag2=0;
27 int S1=0,S2=0,S3=0,S4=0,S5=0,S6=0,C1=0,C2=0;
28 int row = 0;
29 int col = 0;
30
31 void setup(){
32     pinMode(ir_car1, INPUT);
33     pinMode(ir_car2, INPUT);
34     pinMode(ir_car3, INPUT);
```

```
38 pinMode(c1, INPUT);
39 pinMode(c2, INPUT);
40 pinMode(ir_enter, INPUT);
41 pinMode(ir_back, INPUT);
42
43 myservo.attach(2);
44 myservo.write(90);
45 lcd.begin();
46 lcd1.begin();
47 lcd.setCursor (5,1);
48 lcd.print("Car parking ");
49 lcd.setCursor (8,2);
50 lcd.print("System");
51 delay (2000);
52 lcd.clear();
53 Read_Sensor();
54 int total =S1+S2+S3+S4+S5+S6;
55 slot=slot-total;
56
57 Serial.begin(9600);
58 // Wire.setClock(10000);
59
60
61
62 void loop() {
63
64
65 Read_Sensor();
66 lcd.setCursor (2,0);
67 if(S1==1){lcd.print("S1:FULL ");}
68 | | else{lcd.print("S1:FREE");}
69
70 lcd.setCursor (12,0);
71 if(S2==1){lcd.print("S2:FULL");}
72 | | else{lcd.print("S2:FREE");}
73 lcd.setCursor (2,1);
74 if(S3==1){lcd.print("S3:FULL ");}
75 | | else{lcd.print("S3:FREE");}
76
77 lcd.setCursor (12,1);
78 if(S4==1){lcd.print("S4:FULL");}
79 | | else{lcd.print("S4:FREE");}
80 lcd.setCursor (2,2);
81 if(S5==1){lcd.print("S5:FULL ");}
82 | | else{lcd.print("S5:FREE");}
83
84 lcd.setCursor (12,2);
85 if(S6==1){lcd.print("S6:FULL");}
86 | | else{lcd.print("S6:FREE");}
87
88 lcd.setCursor (2,3);
89 if(C1==1){lcd.print("C1:FULL ");}
90 | | else{lcd.print("C1:FREE");}
91
92 lcd.setCursor (12,3);
93 if(C2==1){lcd.print("C2:FULL");}
94 | | else{lcd.print("C2:FREE");}
```

```
95
96     if(digitalRead (ir_enter) == 0 && flag1==0){
97         if (slot>0){
98             flag1=1;
99             if(flag2==0){
100                 myservo.write(180);
101                 slot=slot-1;
102             }
103         }else{
104             lcd.clear();
105             lcd.print("No Parking Slot");
106
107             delay(1500);
108             lcd.clear();
109         }
110     }
111
112     if(digitalRead (ir_back) == 0 && flag2==0){flag2=1;
113     if(flag1==0){myservo.write(180);
114     slot=slot+1;
115     }
116     }
117
118     if(flag1==1 && flag2==1){
119         delay (1000);
120         myservo.write(90);
121         flag1=0, flag2=0;
122     }
123
124     if(Serial.available() > 0) //1st if open
125     {
126         while(Serial.available())
127         {
128             mtlb_data = Serial.read();
129             if(mtlb_data == '#')
130             {
131                 lcd1.clear();
132                 welcome();
133                 lcd1.setCursor(1,1);
134                 lcd1.print(' ');
135                 lcd1.setCursor(0,1);
136             }
137             else
138             {
139
140                 col++;
141                 Serial.print(mtlb_data);
142
143
```

```
141         Serial.print(mtlb_data);
142         lcd1.print(mtlb_data);
143     }
144 }
145
146
147     lcd1.setCursor(--col,1);
148     lcd1.print(' ');
149     lcd1.setCursor(col,1);
150 }
151
152 }
153 void Read_Sensor(){
154     S1=0,S2=0,S3=0,S4=0,S5=0,S6=0,S6=0,C1=0,C2=0;
155
156     if(digitalRead(ir_car1) == 0){S1=1;}
157     if(digitalRead(ir_car2) == 0){S2=1;}
158     if(digitalRead(ir_car3) == 0){S3=1;}
159     if(digitalRead(ir_car4) == 0){S4=1;}
160     if(digitalRead(ir_car5) == 0){S5=1;}
161     if(digitalRead(ir_car6) == 0){S6=1;}
162     if(digitalRead(c1) == 0){C1=1;}
163     if(digitalRead(c2) == 0){C2=1;}
164
165
166 }
167
168 void welcome()
169 {
170     col = 0; row = 0;
171
172     lcd1.setCursor(col,++row);
173 }
```

**Matlab Codes:**

```

1
2 while true
3     clc
4     close all;
5     clear;
6     p="";
7     load imgfildata;
8     cam=webcam
9     preview(cam)
10    pause(3)
11    picture=snapshot(cam)
12    T = readtable('2.csv');
13    A = table2array(T);
14    [~,cc]=size(picture);
15    picture=imresize(picture,[300 500]);
16    if size(picture,3)==3
17        picture=rgb2gray(picture);
18    end
19
20    threshold = graythresh(picture);
21    picture =~im2bw(picture,threshold);
22    picture = bwareaopen(picture,30);
23    imshow(picture)
24    if cc>2000
25        picture1=bwareaopen(picture,3500);
26    else
27        picture1=bwareaopen(picture,3000);
28    end
29    figure,imshow(picture1)
30    picture2=picture-picture1;
31    figure,imshow(picture2)
32    picture2=bwareaopen(picture2,200);
33    figure,imshow(picture2)
34    [L,Ne]=bwlablel(picture2);
35    propied=regionprops(L,'BoundingBox');
36    hold on
37    pause(1)
38    for n=1:size(propied,1)
39        rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
40    end
41    hold off
42
43    figure
44    final_output=[];
45    t=[];
46    for n=1:Ne
47        [r,c] = find(L==n);
48        n1=picture(min(r):max(r),min(c):max(c));
49        n1=imresize(n1,[42,24]);
50        imshow(n1)
51        pause(0.2)
52        x=[ ];
53    totalLetters=size(imgfile,2);
54
55    for k=1:totalLetters
56
57        y=corr2(imgfile{1,k},n1);
58        x=[x y];
59
60    end
61    t=[t max(x)];

```

```

62     if max(x)>.40
63         z=find(x==max(x));
64         out=cell2mat(imgfile(2,z));
65         final_output = [final_output out];
66         p=convertCharsToStrings(final_output);
67     end
68 end
69
70 val1=0;
71 val2=0;
72
73 for i = 1:size(A,1)
74
75     if strcmp(p,A(i,1))==1
76         val1=1;
77         faculty
78
79     end
80     if strcmp(p,A(i,2))==1
81         student
82         val2=1;
83
84     end
85 end
86 end
87
88

```

## Faculty.m

```

1  function faculty
2
3  global arduino
4
5  arduino=serialport('COM8',9600)
6  fopen(arduino)
7
8
9      fprintf(arduino,'# Faculty')
10     dataReceive = fscanf(arduino);
11     pause(3);
12     fprintf(arduino,'#');
13     fprintf(arduino,'# Turn Left')
14     pause(3);
15     fprintf(arduino,'#');
16
17
18 end

```

**Student.m**

```
1  function MATLAB_ARDUINO_LCD2
2
3  global arduino
4
5
6
7      fprintf(arduino,'#   Student')
8      dataReceive = fscanf(arduino);
9      pause(3);
10     fprintf(arduino,'#');
11     fprintf(arduino,'#   Turn Right')
12     pause(3);
13     fprintf(arduino,'#');
14
15 end
```