Creating Parking System with Arduino Software

Chennu Ganesh Tiru Venkata Manikanta Sai, Chennu Aryan Karthikeya

1. Introduction

This Project Holds up in doing up with the smart parking system with Arduino UNO and the effective way of using this parking system more efficiently and flexible with the existing parking system. If this parking system comes into existence it will make the end users more flexible and comfortable when compared with the existing parking system. It makes the end users more flexible now-a-days customers want the things to be more secure but they want to do them so much easier. So, this project would be one to provide the best parking system experience ever before. If this project were implemented in the real-world scenario it would give the end-users a lot of advantages like fuel consumption, manpower which would result in making lots of companies install this system for their parking slots. Now-a-days the biggest problem in lots of cities like Madras, Bangalore, Hyderabad, Mumbai is they don't have the parking slots to make there parking and that makes the customers to become angry and takes lots of time to search their free space and park at their right place and this even results in the wastage of fuel during their roaming in search of the parking slots. So, in this project I came out with a small innovative idea to overcome this parking problem, especially available in Indian places. Let's take a look at the Intelligent Parking System with Arduino UNO in this documentation in detail.

1.1 General Introduction

Traffic congestion caused by vehicles is an alarming problem at a global scale and it has been growing exponentially. Car parking problem is a major contributor and has been, still a major problem with increasing vehicle size in the luxurious segment and confined parking spaces in urban cities. Searching for a parking space is a routine (and often frustrating) activity for many people in cities around the world.

This search burns about one million barrels of the world's oil every day. As the global population continues to urbanize, without a well-planned, convenience-driven retreat from the car these problems will worsen.

The existing doesn't have the proper security system and the customers need to park their cars should search and then park but whereas in this project it's clear that they will notified while they are out of the parking slot the maximum cars that could be able to park and if the parking slot is full they may wait still the car comes out or else they can move to any other parking slots next to them. So, this system makes the user move from outside of the parking slot which is better than coming inside and searching and going out. This parking system will provide the best security and it will give the best parking experience and a flexible way for the customers to park their cars or vehicles in this intelligent parking system.

1.2 Motivation:

This Smart parking system with Arduino could make the end users a more flexible time consuming and fuel consuming system.

So, this parking system is most important and a very useful thing to be implemented in a real world scenario to make our existing system into a developed one. The end users nowa-days like only the things which makes the users easier to use and the complexity is less when compared to the ordinary parking system.

Car parking problem is a major contributor and has been, still a major problem with increasing vehicle size in the luxurious segment and confined parking spaces in urban cities. Searching for a parking space is a routine (and often frustrating) activity for many people in cities around the world.

1.3 Aim & Objective:

The main aim and the objective of this project is to provide the customers or the users with the best quality, secured, easier, flexible product that is the intelligent parking system. And also for the companies they no need to have the securities for the parking slots they may have only the machines that would be controlled by sitting from a specific place. As technology is growing day-by-day but still few things are the same as before. So, in order to decrease the manpower and to make the customers.

This parking system is generally programed and designed in way in the form of the FOR loop which would be resulting in the repetition of the process still some 'N' number of times declared in the system as I am using the 8 Segment anode display as my display I am taking my n as 9 so that it uses a loop and so whenever the parking slot is filled with 9 cars it would not open the gateway that is the servo motor in terms of hardware and if it is less than the limit 9 that is 'N' then the gateway opens up and whenever

there is an entry inside the gateway the display gets incremented by the flag value over there on the system and vice-versa that is when the car comes out of the parking slots it gets decremented by how many comes out.

1.4 System Requirements:

1.4.1 Hardware Requirements:

-Arduino UNO

-Servo Motor

-Anode Display

-Connectors (Male to Male, Female to Female, Male to Female)

1.4.2 Software Requirements:

-Arduino Software To be installed from https://www.arduino.com/

-USB Drivers to be installed in the System

2.1 Introduction of Problem & Its Related Concepts:

The existing doesn't have the proper security system and the customers need to park their cars should search and then park but whereas in this project it's clear that they will notified while they are out of the parking slot the maximum cars that could be able to park and if the parking slot is full they may wait still the car comes out or else they can move to any other parking slots next to them. So, this system makes the user move from outside of the parking slot which is better than coming inside and searching and going out. This parking system will provide the best security and it will give the best parking experience and a flexible way for the customers to park their cars or vehicles in this intelligent parking system.

This system uses IOT, which is the Internet of Things, which means providing the best things in this world to be digitalized and which makes the customers happier and more efficient than any other products. We live in the world of digitized things in our day- today life and this parking system would also provide the best thing of digitalization and flexibility to the customers and consumers who live on this technology. And this parking system would also decrease the amount of fuels that is used and this would control the air pollution all over the country especially in India some of the famous cities like Chennai, Bangalore

, Mumbai etc., and this system could result in providing the people using this parking system with best parking system and with best security system than before systems and this would be a very advantageous system for the people using vehicles on their day-today life and this would make them happy because this makes them to decrease the usage of amount and the manpower when compared with the old parking system that is the ancient parking system with lots of disadvantages in them like the security issues and they don't have hope and the security persons are not to be honest to their parked cars in few infrastructures so, this intelligent parking system would be one of the best remedy for them to park their vehicles safe with best security systems which would be totally computer controlled.

2.2 Why to make smart parking instead of existing one?

According to my research, Smart Parking could result in making the traffic system more efficient for the end users and make them use this smart parking system. Resource- Google Docs

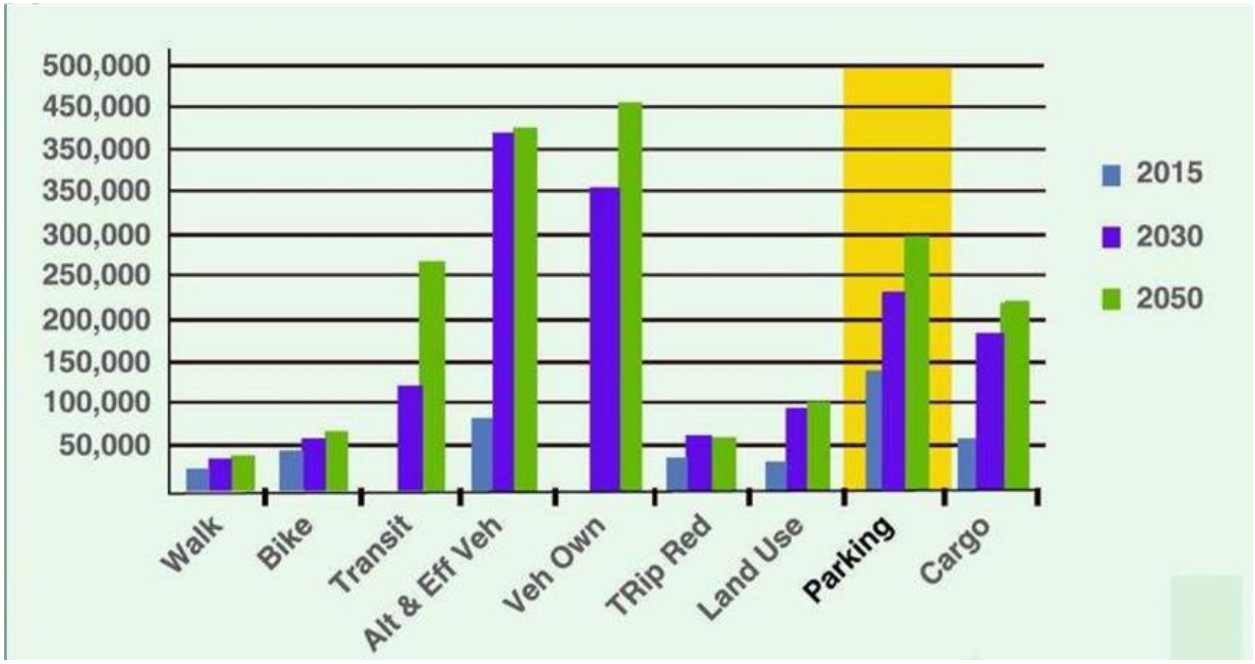

Figure 1 Graph of Comparison

The above graph is the comparison of the assumption by the experts how the future would be like. The graph has the comparison of the things that is travel through which they use like the first one shows that

how many of the people uses the system of walking and the next one shows that how many uses Bikes and the next one shows that how many uses transit and the next shows that how many uses the alternative vehicles and efficient vehicles and the next one shows that how many has their own vehicles and the next one shows that tractors and using for the lands and agriculture and the next one shows that how many park their cars and the next one shows that how many of them uses the Cargo loads.

Majority of this is Parking is most thing and as of the research it shows in 2050 the future would be every home would be having at least one car in each home.

2.3 Proposed Method:

As of the above problems and by having these things in mind I have implemented this hardware project in order to provide a secured parking system and flexible parking system for the customers and the companies to reduce manpower for them which has become the most other thing in India.

We are facing the problems of the low parking spaces all around the country and also there is no proper security because of the salary incrementation for the security persons and there is lot of consumption of fuel which is also being a big problem as there is less amount of fuel available there is more amount price increment and by implementing my project we can have these things to be control to make our country better and safer place to live and even the air pollution could be controlled at most 10% of the existing air pollution especially in cities .

As we all know that India is one of the countries with lots of population of people and it is also one of the countries with lots of cars available all over the country. Though India is poor at its marketing and the Sensex, India is the country with lots of imported cars & even the middle-class people have cars. And that's the main reason why there is a lot of problem with parking in India. That is why we need a smart system to make this parking system more secure and flexible and to make the customers spend less amount of money for the sake of parking their vehicles in the parking slots. So, that is why this intelligent parking system would help them to provide the best parking with less manpower and spending up money for the sake of parking their cars.

3.1 Brief Introduction

The Main Feature:

The parking system has the advantage of working on the basis of the hardware with this Arduino UNO. This System makes the parking system more flexible like when there is a loop of n cars to go in only n could go in and the cars exceeding than this percentage it will make the gateway not to be opened. The same thing continues for the next that is the outer gateway and the count gets displayed on the Anode Display

This parking system is generally programed and designed in way in the form of the FOR loop which would be resulting in the repetition of the process still some 'N' number of times declared in the system as I am using the 8 Segment anode display as my display I am taking my n as 9 so that it uses a loop and so whenever the parking slot is filled with 9 cars it would not open the gateway that is the servo motor in terms of hardware and if it is less than the limit 9 that is 'N' then the gateway opens up and whenever there is an entry inside the gateway the display gets incremented by the flag value over there on the system and vice-versa that is when the car comes out of the parking slots it gets decremented by how many comes out.

We are facing the problems of the low parking spaces all around the country and also there is no proper security because of the salary incrementation for the security persons and there is lot of consumption of fuel which is also being a big problem as there is less amount of fuel available there is more amount price increment and by implementing my project we can have these things to be control to make our country better and safer place to live and even the air pollution could be controlled at most 10% of the existing air pollution especially in cities .

Intelligent Parking System would enable the following

Accurately predict and sense spot/vehicle occupancy in real-time. Guides residents and visitors to available parking. Optimize Parking Space Usage. Simplifies the parking experience and adds value for parking stakeholders, such as drivers and merchants. Help traffic in the city flow more freely leveraging IoT technology. Enables intelligent decisions using data, including real–time status applications and historical analytics reports. Smart Parking plays a major role in creating a better urban environment by

reducing the emission of CO2 and other pollutants. Smart Parking enables better and real time monitoring and managing of available parking space, resulting in significant revenue generation. Provides tools to optimize workforce management

As we all know that India is one of the countries with lots of population of people and it is also one of the countries with lots of cars available all over the country. Though India is poor at its marketing and the Sensex, India is the country with lots of imported cars & even the middle-class people has cars. And that's the main reason why there is a lot of problem with parking in India. That is why we need a smart system to make this parking system more secure and flexible and to make the customers spend less money for the sake of parking their vehicles in the parking slots. So, that is why this intelligent parking system would help them to provide the best parking with less manpower and spending up of money for the sake of parking their cars.

Majority of this is Parking is most thing and as of the research it shows in 2050 the future would be every home would be having at least one car in each home.

Absence of a robust billing platform leading to possible revenue leakages. Interoperability between devices/lack of standards. Although other countries have solutions deployed, Smart parking does not really provide much solution to two wheelers as yet in India.

Various Security issues and threats to the installed on-site parking meter. The IoT enabled Parking System shall support mechanisms to correlate charging data/records from different IoT Application Service Providers. The IoT enabled Parking System shall support triggering M2M Devices to report on-demand information regarding collected data from other M2M Devices. Smart parking providers will need to establish reliable application programming interfaces (APIs) that enable service partners to provide consumers with access to smart parking services on-line through a variety of channels, including the web, mobile phone apps, connected personal navigation devices and car telematics services

3.2 Requirement Analysis

3.2.1 Hardware & Software Functional Requirement

The functional requirements are the general requirements needed for the functioning of the product under the execution of the products and the software that has been used to make the system more accurate and makes the hardware components to work properly and easier with requirements. Now, let's see the requirements required for the execution of this project.

3.2.1.1 Hardware Perspective & Its Features

-Arduino UNO:

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone.

Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

-Servo Motor:

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

-7 Segment Anode Display:

A seven-segment display (SSD), or seven-segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays.

Seven-segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information.

It consists of 7 segments in them and that's is why it is called as the 7 Segment Anode

Display which could get the output only for specific numbers and alphabets containing only 7 Segments in them.

-Jumper Wires:

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

3.2.1.2 Software Perspective & Its Features

--Arduino IDE (Integrated Development Environment):

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

-Required Tools and Driver Software's:

The tools required to complete this execution with Arduino IDE are

-USB Driver

-JDK (Java Development Kit)

4.1 Design

4.1.1 Design Overview:

This intelligent parking system has the design of connections of wires that is the Jumper Wires through which the data is sent to the hardware components connected. The main component used in this project is the Arduino UNO which is a microcontroller which is controlled by the program entered in the Arduino IDE which is one of the applications developed by the Arduino company to make their microcontrollers to have a control. In this project, we are using the Anode display and the servo motor. The servo motor is a device generally which will be able to rotate for about 90 deg. Only but whereas there is another motor called stepper motor which will rotate up to 180deg. As we need only the gateway so I am using this servo motor which rotates only up to 90deg. Which would be useful like the gateway for the parking slot and next the anode display is a 7 Segment display which is used to display the numbers by making the bits to be on and off. Since I am using only up to 9 cars to be parked inside my parking slot that's why this 7-segment anode display is used. Now, let's see the architecture through which it has been implemented.

The functional requirements are the general requirements needed for the functioning of the product under the execution of the products and the software that has been used to make the system more accurate and makes the hardware components to work properly and easier with requirements.

4.1.2 Design Architecture:

4.1.2.1 Overall Architecture & Its Connections:

The design of any of the Arduino based architecture would generally depend up on the program given through the java coding through the Arduino IDE(Integrated Development Environment) this architectural connections of the pins are generally based on the connectivity of the pins declared on the software and whenever the compilation is done check whether the board is properly connected or not and also check out with the drivers which could result in some errors if it is not been found on the system through which the compilation take place and after that compilation is successfully over after the execution without any errors then the next thing to be followed up on is that the uploading of the compiled program into the Arduino board it may be of any boards UNO,MEGA etc., The next process to be followed after the compilation of the program is that the uploading of the program into the Arduino

which gives the specific pins with some data or power as declared in the compiled program and if the hardware components declared on the program and the pins are been connected properly then when the program is being uploaded it will be updated on with the board and with pins and the hardware works properly.

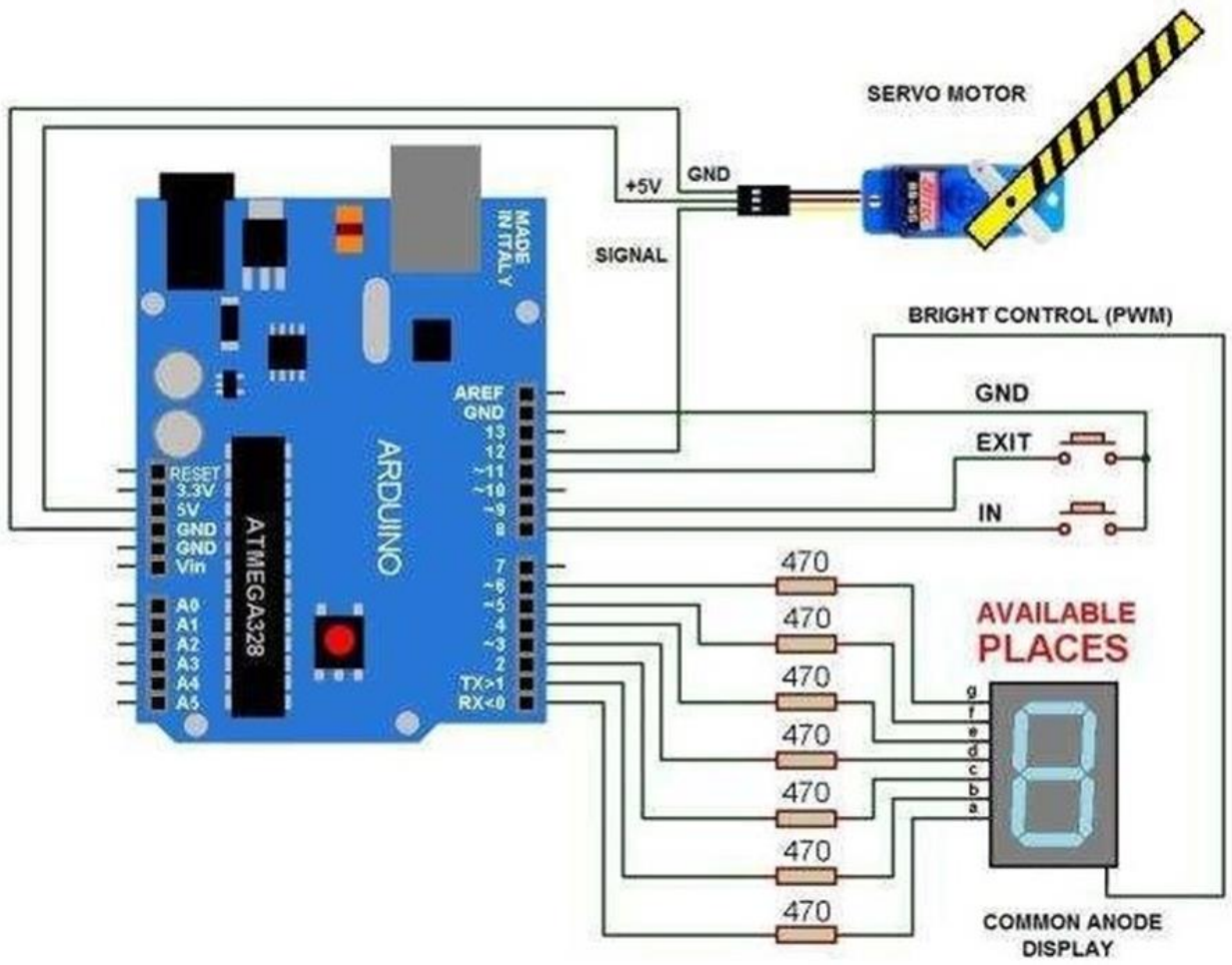

Figure 2 – Architecture

From the above diagram architecture, it has been shown how the connections are established and according to the program I have implemented the connections have been done with the following pins and its connections.

According to my architectural view I have used 14 PINS from the Arduino Uno and it has been connected to the connections like anode display, servo motor and the in and out gates.

When the above following pins are being connected with the following hardware components which could make the system to get the output properly according to the program it has to be updated. When the specified pins are being declared in the program and it is being executed and sent on to the process

that is to the Arduino IDE and after the execution it moves upon to the next step that is the program to be uploaded on the board and the pins should be connected properly to be get the output respectively.

4.1.2.2. Anode Display Connections:

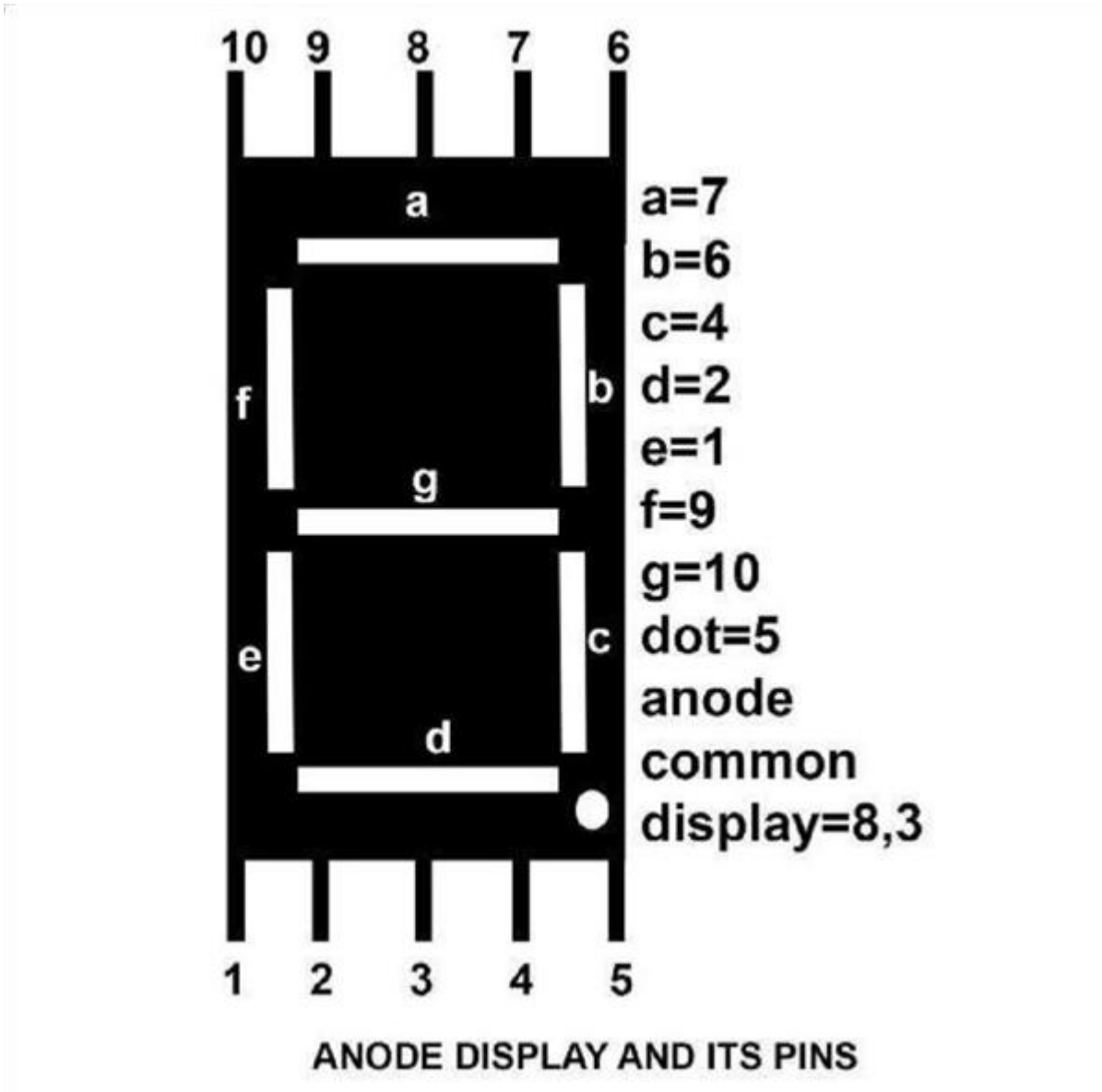

Figure 3 Anode Display & Pins

The Above diagram shows that there are generally 10PINS available with the 8 BIT anode display and each pin has its own function in the 8 Bit Segment respectively.

Now, let's make a view of how the connections has been established between the Arduino Uno and the Anode Display

Arduino Pin	Anode Pin
0	7(a)
1	6(b)
2	4(c)
3	2(d)
4	1(e)
5	9(f)
6	10(g)
~11	8/3 (Common Display)

Table 1 – Anode Display Connections

According To the above values the connections are established according the program updated on the Arduino UI in the system

When the connections established according to table 1 -Anode Display Connections the output will be displayed on the 7 Segment Anode Display

4.1.2.3. Servo Motor:

Servo Motor is a motor which only could rotate only about 180 deg. And we could control its actions that is the rotating degrees by having the package that is #include<servo.h>. So, by using this package we could be able to import the control of the servo motor respectively.

It has generally 3 PINS in them by color of the pins it can be divided into individuals

-Black/Brown-Negative

-Red-Positive

-Orange-Signal

Pins Connections and Its Establishment According to the program

Arduino Pin	**Servo Pin**
Gnd	Negative (Black Pin)
5V	Positive (Red Pin)
12	Signal (Orange Pin)

Table 2 – Servo Motor Connections

When the connections established according to table 2- Servo Motor Connections the output will be done according to the program.

4.1.2.4. In and Out Gate:

This IN & OUT gates act as the entry and exit check lines of the parking slots which would provide the entry check and exit check and will be counted and displayed on anode display and the gate open and closes that is the servo motor

Connections Of the in and out gates

Arduino Pin	**Specification**
Gnd	Negative
8	IN PIN
~9	EXIT PIN

Table 3 – In and out gateway connections

According to the above specifications if the accessories are connected successfully and if the program is executed and is uploaded properly with the Arduino UI It Starts working.

Now, let's see the coding part with the Arduino UI

5.1 Tools Used

5.1.1 General Introduction:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

In this hardware project, we are using the Arduino IDE (Integrated Development Environment) which is the general software used for all types of Arduino boards from UNO to MEGA. This software's has a very easy and flexible platform to make the coding easily

and to check them by compiling once and after that uploading on to the Arduino board

takes place. This Arduino software generally uses java based coding and also uses some basic programming like C, C++., So, we are using this IDE to make the coding more easy and flexible and to find the errors easily.

5.1.2 Required Tools and Driver Software's:

Arduino IDE (Integrated Development Environment)

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

The tools required to complete this execution with Arduino IDE are

-USB Driver

-JDK (Java Development Kit)

5.2 Program Implementation

5.2.1 Program Description:

The program available in the below implementation column consists of the packages for importing the servo motor on to the Arduino IDE and the servo motor has been connected to the pin number 12 and the anode display control is being controlled through the pin number 11 and the exit pin has been connected with the pin number 9 and the input that is the in gate pin in the pin number 8 and for the next process that is the servo motor is controlled by making the bar low as 177 that is closing the gate and bar up as 95 that is the closing up of the gate and next is the capacity of the cars to be parked inside the

parking slots is declared as 8 and the intensity of the anode display has been given in order to control its intensity in terms of percentage that is of 80% and as the anode display has seven segments it is being declared from segA – segG(0-6) and then in order to make the numbers from 0-9 to be viewed in the display an array has been declared 10 with the decimal notation as 0's and 1's that is glowing and not glowing.

A formula has been given in order to give the output on the anode display

(255*intensity/100) it writes the output on the analog screen and at last displays the output on the anode display and the bar gets up and down according to the program description of it.

This is the program description and how the program runs when executed on the Arduino IDE. That is the software which allows us to do the hardware programs and import them with the program we want to execute and this could provide us to make the parking system.

This acts as the algorithm of the project I have been implementing, which is the Intelligent Parking System with Arduino UNO.

5.2.2 Program Implementation With Arduino IDE:

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo #define ServoM 12
        //Connected to the servo motor.
#define Bright 11 //servo library disable PWM on pins 9 and 10. #define Exit 9
        //Pin connected to the EXIT button.
#define In 8 //Pin connected to the IN button. #define BarLow 177
        //Low position of the barrier. #define BarUp 95 //Up position of the barrier.
#define CAPACITY 8 //Capacity of the parking lot. #define INTEN
        80 //Display intensity % //Pins conections to segments (cathodes).
#define segA 0
#define segB 1

#define segC 2

#define segD 3
```

```
#define segE 4

#define segF 5

#define segG 6


//Array with the segments to represent the decimal numbers (0-9). byte segments[10] = { // pgfedcba <-- segments
B00111111, // number 0 B00000110, // number 1 B01011011, // number 2 B01001111, // number 3 B01100110, // number 4 B01101101, // number 5 B01111101, // number 6 B00000111, // number 7 B01111111, // number 8 B11101111 // number 9
}; void setup () { myservo.attach(ServoM) ; // attaches the servo.
pinMode(Exit, INPUT); // set "EXIT" button pin to input pinMode(In, INPUT); // set "IN" button pin to input digitalWrite(Exit, HIGH); // Connect Pull-Up resistor. digitalWrite(In, HIGH); // Connect Pull-Up resistor.
pinMode(segA, OUTPUT); pinMode(segB, OUTPUT); pinMode(segC, OUTPUT); pinMode(segD, OUTPUT); pinMode(segE, OUTPUT); pinMode(segF, OUTPUT); pinMode(segG, OUTPUT); pinMode(Bright, OUTPUT); analogWrite(Bright, 255 * INTEN / 100); myservo.write(BarLow);
//Barrier in the low position
// delay(1000);
} int Available = 9; // Number of places available.


void loop() { Display(Available);
if (digitalRead(In) == 0)
{
if (Available != 0) { Available--; myservo.write(BarUp); delay(3000);
myservo.write(BarLow);
}
}
if (digitalRead(Exit) == 0)
{
```

```
if (Available != CAPACITY) { Available++;
myservo.write(BarUp);                    delay(3000); myservo.write(BarLow);
}

}
}
/*Put the segments according to the number.*/ void Display(int number)
{  byte segs = ~segments[number];          //"~" is used for commom anode.    digitalWrite (segA, bitRead (segs, 0));    digitalWrite (segB, bitRead (segs, 1));    digitalWrite (segC, bitRead (segs, 2));    digitalWrite (segD, bitRead (segs, 3));    digitalWrite (segE, bitRead (segs, 4)); digitalWrite (segF, bitRead (segs, 5));    digitalWrite (segG, bitRead (segs, 6));
}
```

Conclusion:

This intelligent parking system which is simple, economical and provides an effective solution to reduce carbon footprints in the atmosphere. It is well managed to access and map the status of parking slots from any remote location through the display outside the parking slots. Thus, it reduces the risk of finding the parking slots in any parking area and also it eliminates unnecessary traveling of vehicles across the filled parking slots in a city. So, it reduces time and it is cost effective also. By implementing this system we could be able to save nature and control air pollution and also we could be able to control manpower through this and there will be a great reduction of cost for both the customers and the merchants who make parking slots for their customers.

References:

[1] An Article of Smart Parking from Happiest Minds Technologies Pvt. Ltd.

[2] Science Research Group (2014) Intelligent Parking Management System Based on Image Processing. World Journal of Engineering and Technology, 2, 55-67.

[3] Al-Kharusi, H. (2014) Intelligent Car Parking Management System. Master of Engineering Thesis.

[4] Al-Kharusi, H. (2014) Intelligent Car Parking Management System. Thesis, Master of Engineering, Electronic and Computer System Engineering, Massey University, Palmerston North, New Zealand.

[5] Siemens, A.G. (2001) Systematic Parking Solutions by Industrial Solutions and Services, Intelligent Traffic Systems. Ref. No. E10003-A800-W6-X-7600 Systematic Parking 6052 WS 1.0 Edition 003, 2001-12-01.

http://www.siemens.de/traffic

[6] Wootton, J.R., Garcia-Ortiz, A. and Amin, S.M. (1995) Intelligent Transportation Systems: A Global Perspective. Mathematical and Computer Modelling, 22, 259-268.

[7] Faheem, S.A., Mahmud, G.M., Khan, M., Rahman and Zafar, H. (2013) A Survey of Intelligent Car Parking Systems. Elsevier, 11, 714-726.

[8] Zhang, X. and Wan, D. (2010) Economic Analysis of Regional Parking Guidance System Based on TIA. 2010 WASE International Conference on Information Engineering (ICIE), 14-15 August 2010, 401-404.

[9] Wiseman, Y. (2010) Take a Picture of Your Tire! Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010), Qingdao, 151-156.