# Visvesvaraya Technological University, Belagavi

A Mini Project report on

# SMART PARKING SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree

**Bachelor of Engineering**

**In**

**Electronics and Communication Engineering**


Submitted by

| | |
|---|---|
| **AKSHATHA N** | (4MH18EC004) |
| **MANJUNATH K L** | (4MH18EC061) |
| **PALLAVI M R** | (4MH18EC076) |
| **SOUNDARYA S** | (4MH18EC100) |

Under the guidance of

**Mr. SHIVAPRASAD K**

Assistant Professor

Dept. of E&CE,

MIT MYSORE

**Department of Electronics & Communication Engineering**

**Maharaja Institute of Technology Mysore**


**2020-21**

Visvesvaraya Technological University, Belagavi

# Maharaja Institute of Technology Mysore

## Department of Electronics and Communication Engineering

# *Certificate*

Certified that the Mini project work entitled **"SMART PARKING SYSTEM"** is a Bonafide work carried out by **AKSHATHA N (4MH18EC004), MANJUNATH K L (4MH18EC061), PALLAVI M R (4MH18EC076), SOUNDARYA S (4MH18EC100),** and this report of Mini project is submitted in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of **Visvesvaraya Technological University, Belagavi** during the year **2020-2021**.

It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the project report and has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

| Signature of the Guide | Signature of the HOD | Signature of the Principal |
|---|---|---|
| **Mr. Shivaprasad K** | **Dr.Ravichandra** | **Dr.Naresh kumar** |
| Asst. Professor | Assoc. Prof. & HOD | Principal |
| Dept. of E&CE | Dept. of E&CE | MIT Mysore |
| MIT Mysore | MIT Mysore | |

# Abstract

With increasing automobiles in parking space, we could face problems like unplanned parking, lack of discipline, wasting time and fuel while looking for free space around the parking ground. These problems could be solved by applying smart technology. Here Arduino microcontroller is used for controlling car parking system. The proposed system will detect an available parking slot and number of vehicles in parking slot and saves time, saves fuel and offer monitoring car parking system with low cost.

# Table of Contents

# LIST of Figures

# List of Tables

# Chapter 1

# PREAMBLE

## 1.1 INTRODUCTION

Due to the surge in urbanization, the usage of automobiles has increased which in turn has led to traffic and parking difficulties. The most wide spread solution used currently, is to increase manpower to handle such traffic. Even in malls, trade centers and business parks, parking of vehicles has become an issue. This is related to parking problems which one of them is the difficulty of knowing the condition of vacant space in the wide parking lot so that the driver spends his time just to find a parking place and tend to more difficult along with the increasing number of vehicle ownership. Problems related to parking can be solved if the driver can be informed beforehand about the availability of parking space. As the result, the concept of smart parking system increased.

Various approaches and research have been done to overcome from parking problems. Since the early 1970s, smart parking has been implemented throughout Europe, the UK, and Japan. The initial system is displayed in the driver's parking information such as availability status and/or the amount of space available. More complex smart parking incorporates more advanced technology to serve customers with different needs. A recent survey conducted by the International Parking Institute reflects an increasing number of innovative ideas related to parking systems. Currently, there are certain parking systems that are able to provide real-time information about available parking spaces. Such systems require efficient sensors to be placed in parking lots to monitor parking spaces and rapid data processing units to gain practical insights of data collected from various sources.

.

## 1.2 BLOCK DIAGRAM

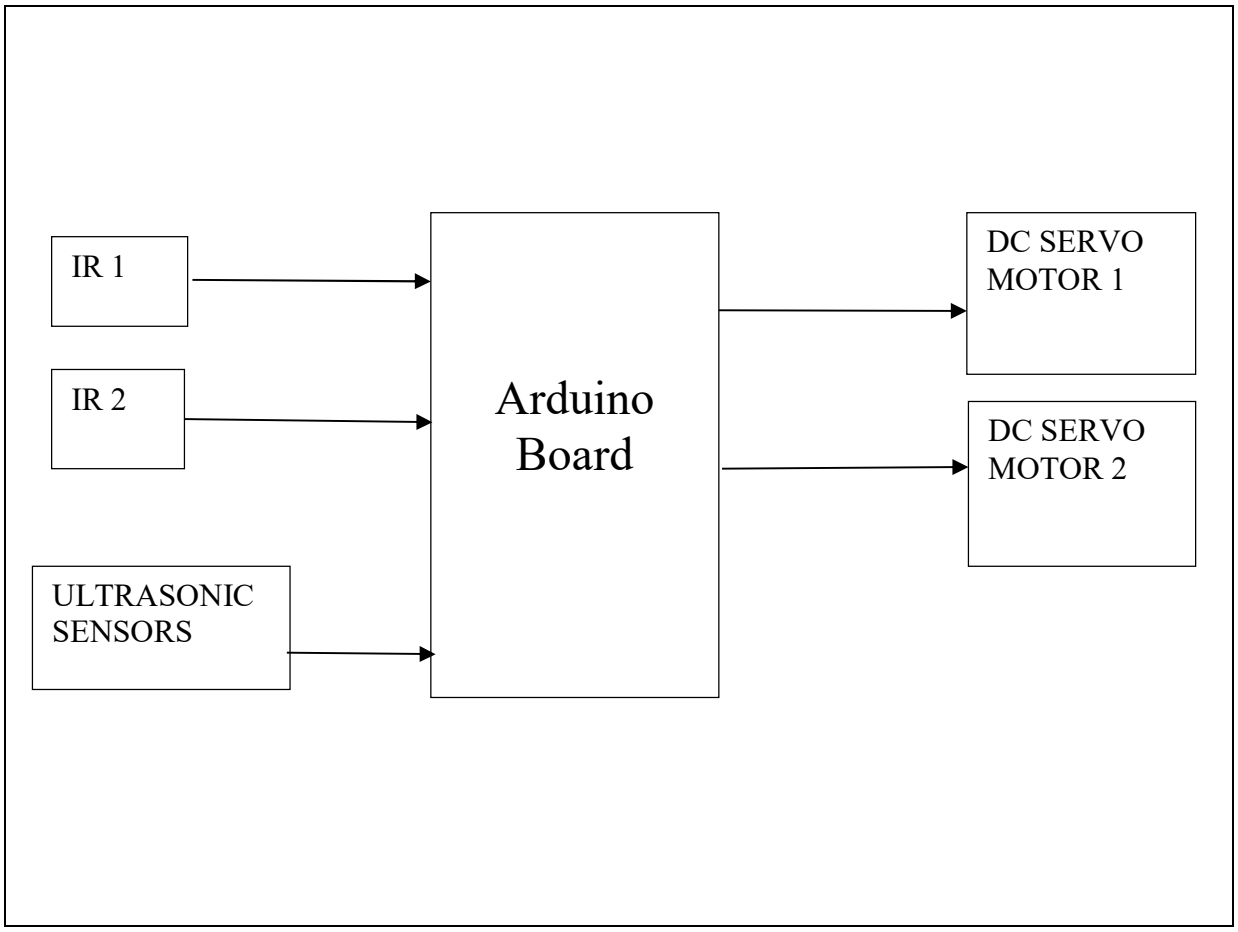


Fig. 1.1: Smart Parking System Diagram

The block diagram consist of 2 IR Sensors, 2 servo motors, ultrasonic sensors & 1 Node mcu microcontroller board. The IR sensor 1 is connected in the input side along with DC servomotor 1 which is used to sense the vehicle at entry gate and IR sensor 2 is connected in the output side along with DC servomotor which is used to sense the vehicle at exit gate. Ultrasonic sensors is used to detect the slots and LCD display used for display slots.

## 1.3 OBJECTIVES AND SCOPE OF THE PROJECT

The objectives of our project are:

1. To provide efficient parking slots for people.
2. To identify the presence of car and number of cars in parking slots.

The scope of the project are:

Technological advances have seen no implementation in parking management systems till date. Our true motivation came from a discussion where we were talking about the future prospects of IoT and cloud computing. We plan to create an effective smart parking system that significantly reduces traffic congestion and also help to find a parking slots for the people.

## 1.4. LITERATURE SURVEY

Paper [1] proposed Present day's car parking has become a significant issue in urban areas. The causes of traffic waste of time and money. Methods: This prototype is developed using a sensor circuit, RFID and IoT. This system helps the user to find parking slots easily

Paper [2] suggests the Internet of Things (IoT) played a vital role in connecting the surrounding environmental things to the network and made it easy to access those un-internet things from any remote location.

Paper [3] is issued in the International Journal of Computer Application in which it suggests to find out the available vacant parking space in the central city area is most in parking space. If the particular parking slot is empty, then

that parking slot can be reserved for parking by any car driver using QR code verification before reach to that specific place.

## 1.5 REPORT ORGANISATION

**Chapter 1:** In this chapter, a brief introduction of smart parking system is described with a generalized schematic diagram. Also it describes the scope of the designed system, objectives and literature survey.

**Chapter 2:** This chapter focusses on complete detail of all the hardware and software components used along with the specifications of each and the pin diagrams are also included and methodology.

**Chapter 3:** This chapter describes the result and conclusion of the project.

# Chapter 2

# Materials and Methods

## 2.1 Hardware Requirements

## Arduino Board

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning:

**Microcontroller:** This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

**External Power Supply:** This power supply is used to power the Arduino development board with a regulated voltage ranging from 9 – 12 volts.

**USB plug:** This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.

**Internal Programmer:** The developed software code can be uploaded to the microcontroller via USB port, without an external programmer.

**Reset button:** This button is present on the board and can be used to resets the Arduino microcontroller.

**Analog Pins:** There are some analog input pins ranging from A0 – A7 *(typical)*. These pins are used for the analog input / output. The no. of analog pins also varies from board to board.

**Digital I/O Pins:** There are some digital input pins also ranging from 2 to *16 (typical)*. These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.

**Power and GND Pins:** There are pins on the development board that provide 3.3, 5 volts and ground through them

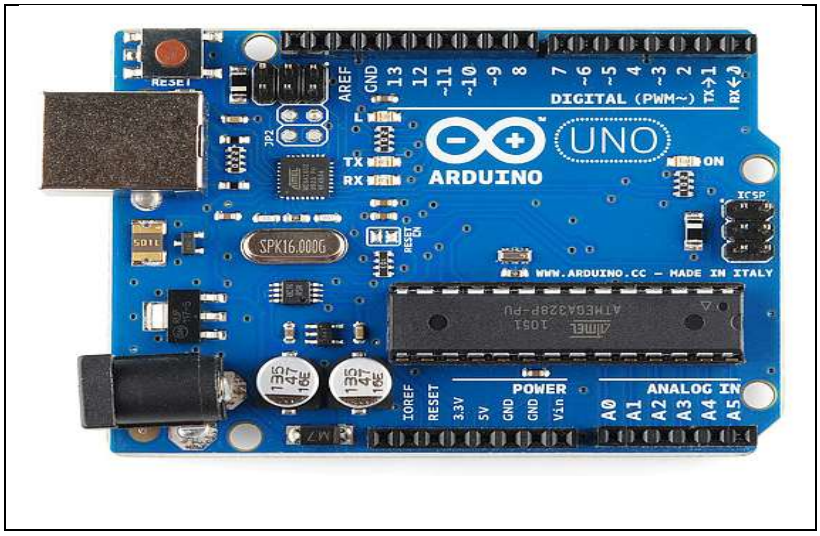


Fig.2.1.1 Arduino UNO Board

## Pin Out Configuration:

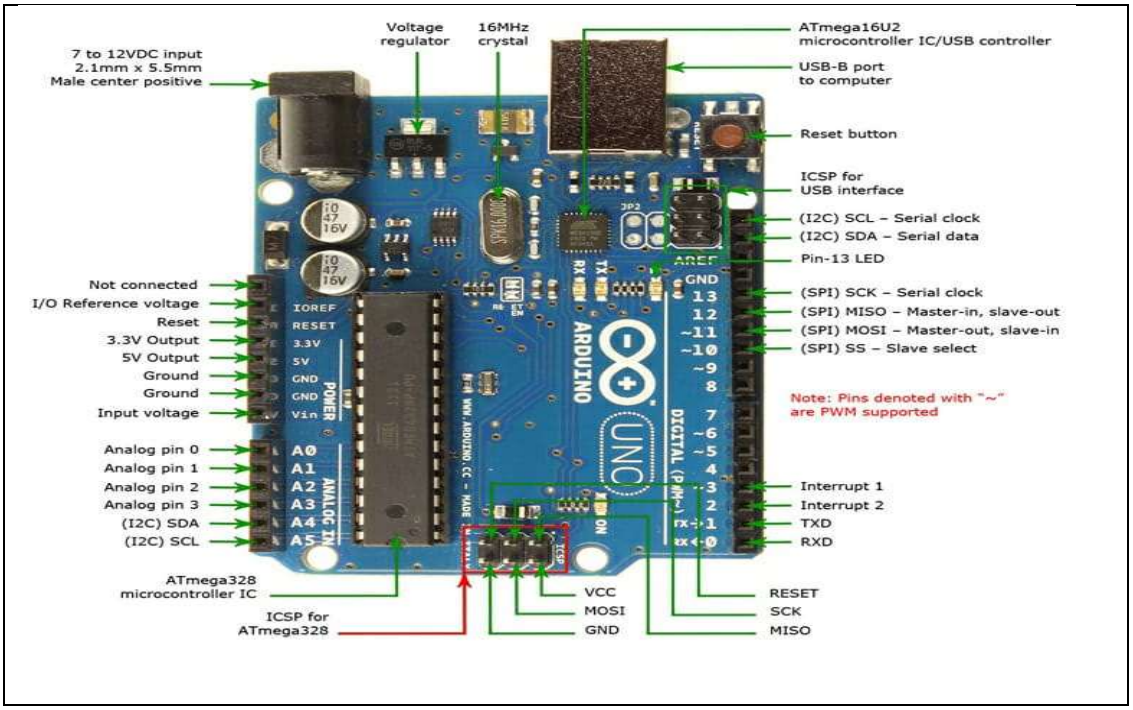


Fig 2.1.2 Pin out diagram

## 2.1.3 INPUT AND OUTPUT

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green)
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

## 2.2 DC SERVO MOTOR

A servomotor is designed to move to a given angular position. A typical servo motor has three connections. Two of them are the positive and 0V supply lines. The third connection carries the control signal pulses from the control circuit (the Arduino in this case).Servo motors may be classified, according to the torque it can withstand, as mini, standard and giant servos. Usually mini and standard size servo motors can be controlled by Arduino directly with no need to external driver.

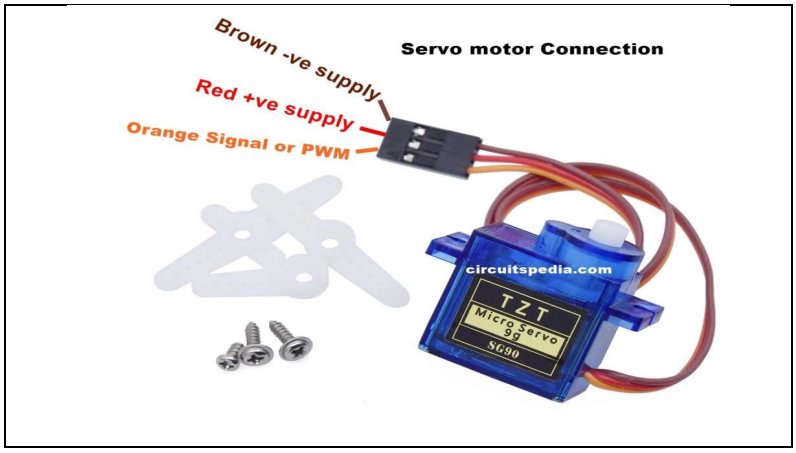


Fig 2.2.1: DC servo motor

**Working of servo motor:**

Servo motors control position and speed very precisely. Now a potentiometer can sense the mechanical position of the shaft. Hence it couples with the motor shaft through gears. The current position of the shaft is converted into electrical signal by potentiometer, and is compared with the command input signal. In modern servo motors, electronic encoders or sensors sense the position of the shaft.

**Controlling of servo motor:**

Usually a servomotor turns 90 degree in either direction hence maximum movement can be 180 degree. However a normal servo motor cannot rotate any further to a build in mechanical stop. We take three wires are out of a servo .Positive, ground and control wire. A servo motor is control by sending a pulse width modulated (PWM) signal through the control wire. A pulse is sent every 20 milliseconds. Width of the pulses determine the position of the shaft.

## 2.3 ULTRASONIC SENSOR

Ultrasonic sensors use sound to determine the distance between the sensor and the closest object in its path. The HR-SR04 ultrasonic sensor uses sonar emission technique to determine distance with an object just like bats or dolphins do. Ultrasonic sensors are essentially sound sensors, but they operate at a frequency above human hearing. It offers excellent range detection without contact but with high accuracy of stable readings to use the package in an easy manner. Its operation is not affected by sunlight or black material like Sharp rangefinders but acoustically soft materials like cloth can be difficult to detect. It comes with a complete ultrasonic transmitter and receiver module.

$$d = v * t \quad \text{---------- 1}$$

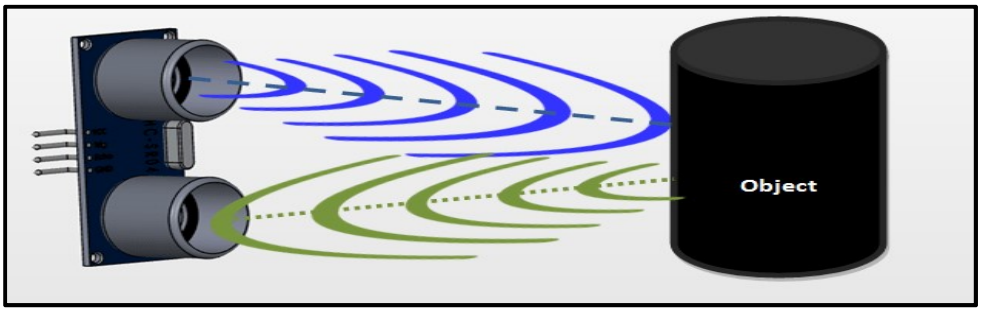


Fig. 2.3.1 Ultrasonic sensors detection

The speed of sound can be calculated based on the variety of atmospheric conditions, including temperature, humidity and pressure. Actually calculating the distance will be shown later on in this document. It should be noted that ultrasonic sensors have a cone of detection, the angle of this cone varies with distance. The ability of a sensor to detect an object also depends on the objects orientation to the sensor. If an object doesn't present a flat surface to the sensor then it is possible the sound wave will bounce off the object in a way that it does not return to the sensor.

The HCSR04 Specifications are

- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15º
- Ranging Distance: 2-400 cm
- Resolution: 0.3 cm
- Measuring Angle: 30º
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm
- Weight: approx. 10 g

### 2.3.1 TIMING CHART AND PIN EXPLANATIONS

| Pin Number | Pin Name | Description |
|---|---|---|
| 1 | V cc | The Vcc pin powers the sensor, typically with +5V |

| | | |
|---|---|---|
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

**TABLE 2.3.1: PIN EXPLANATION**

The HC-SR04 has four pins, VCC, GND, TRIG and ECHO as mentioned intable2.4.These pins all have diffrent functions. The VCC and GND pins are th e simplest, they power the HC-SR04.

These pins need to be attached to a +5 volt source and ground respectively. The re is a single control pin: the TRIG pin. The TRIG pin is responsible for sendin g the ultrasonic burst. This pin should be set to HIGH for 10 µs, at which point t he HC-SR04 will send out an eight cycle sonic burst at 40 kHz. After a sonic bu rst has been sent the ECHO pin will go HIGH. The ECHO pin is the data pin -- it is used in taking distance measurements. After an ultrasonic burst is sent the p in will go HIGH, it will stay high until an ultrasonic burst is detected back, at w hich point it will go LOW.

The HC-SR04 is shown in figure 2.3.1, and they can be triggered to send out an ultrasonic burst by setting the TRIG pin to HIGH. Once the burst is sent the ECHO pin will automatically go HIGH. This pin will remain HIGH until the burst hits the sensor again. You can calculate the distance to the object by keeping track of how long the ECHO pin stays HIGH. The time ECHO stays

HIGH is the burst spent travelling. Using this measurement in equation 1 along with the speed of sound will yield the distance travelled.

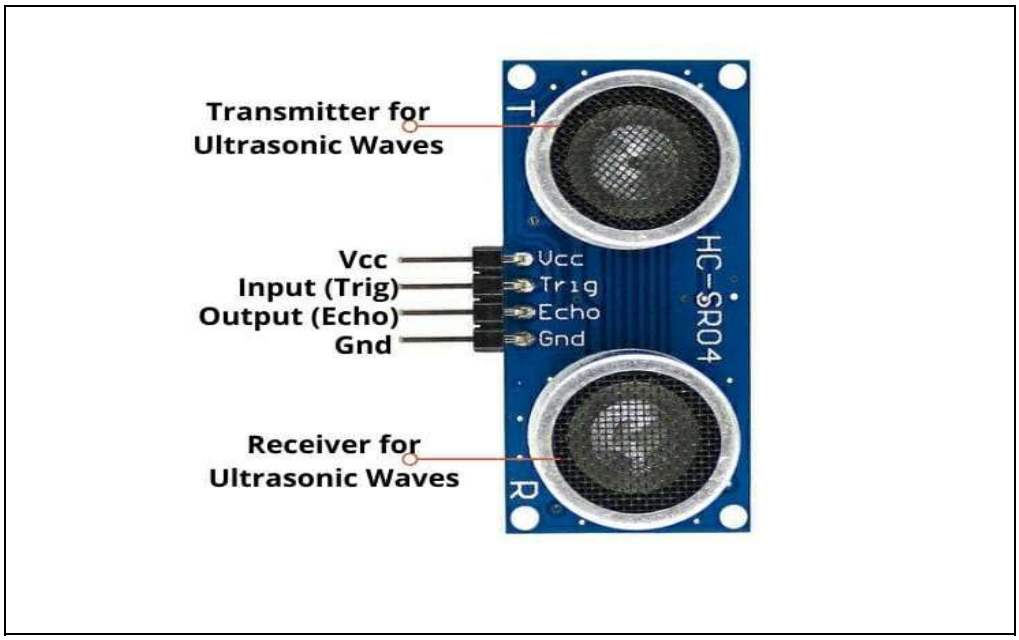


Fig. 2.3.2: ULTRASONIC SENSOR BOARD

## 2.4 IR Sensors

Two IR sensors are used at entry and exit gate to detect the presence of car and automatically open or close the gate.IR Sensor is used to detect any object by sending and receiving the IR rays.

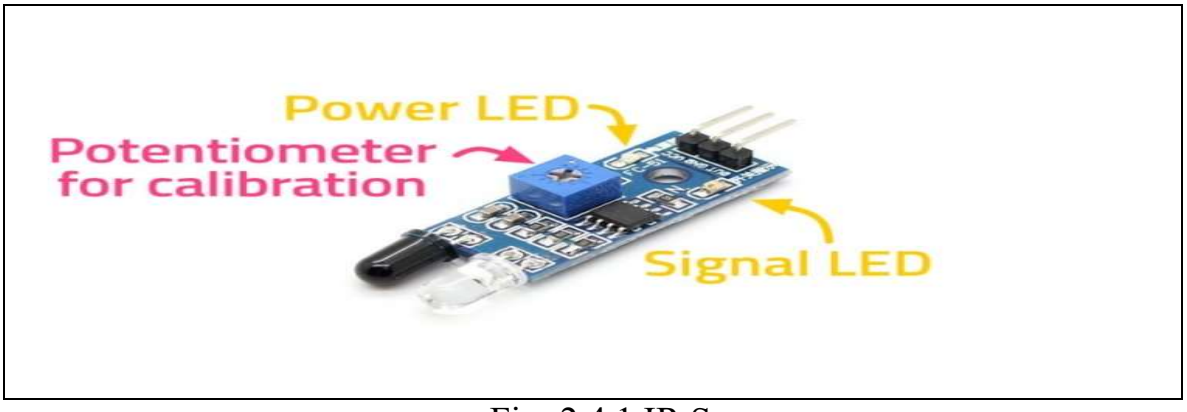


Fig: 2.4.1 IR Sensors

Two servos will act as entry and exit gate and they rotate to open or close the gate. An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was

accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest.

There are two types of infrared sensors: active and passive. Active infrared sensors both emit and detect infrared radiation. Active IR sensors have two parts: a light emitting diode (LED) and a receiver. When an object comes close to the sensor, the infrared light from the LED reflects off of the object and is detected by the receiver. Active IR sensors act as proximity sensors, and they are commonly used in obstacle detection systems (such as in robots). IR detectors are little microchips with a photocell that are tuned to listen to infrared light.

## 2.5 Software Requirements:

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have intelligent code completion. The software requirements of the project make uses of Arduino IDE 1.8.2 version. 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.

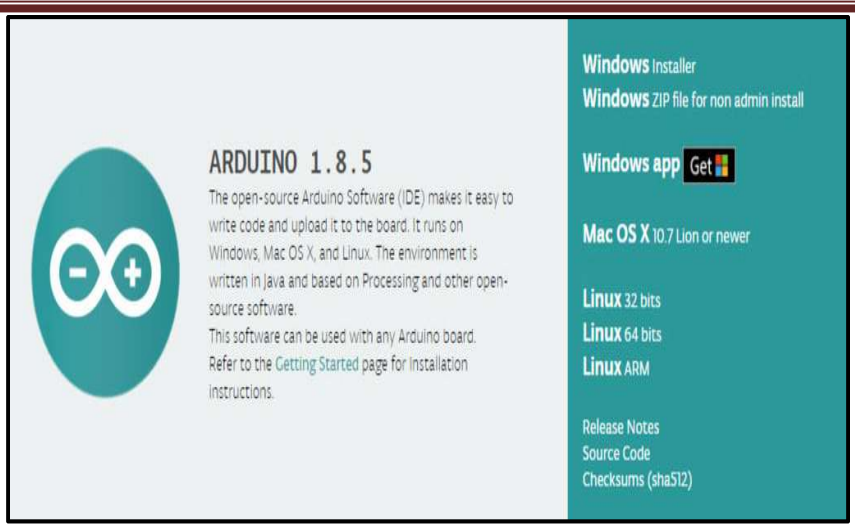


Fig.2.5.1: ARDUINO IDE 1.8.5 SOFTWARE

The Arduino Integrated Development Environment or Arduino software contains a text editor for writing code. A message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and genuine hardware to upload programs and communicate with them.

**TABLE 2.5.1.ARDUINO IDE TOOLBAR**

| | |
|---|---|
|  | Verify<br>Checks your code for errors compiling it. |
|  | Upload<br>Compiles your code and uploads it to the configured board. See uploading below for details. |
|  | New<br>Creates a new sketch |
|  | Open<br>Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content. |
|  | Save |

| | |
|---|---|
| | Saves your sketch |
|  | Serial Monitor<br>Opens the serial monitor. |

## WRITING SKETCHES

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension in out. The editor has a feature of cutting/passing and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. Figure 4.1 shows the window when we open the Arduino IDE software. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow to verify and upload programs, create, open, and save sketches, and open the serial monitor. The toolbar buttons allow to verify and upload programs, create, open and save the sketches and open the serial monitor. Additional commands are found within the five menus: File, Edit, Sketch, Tools, and Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

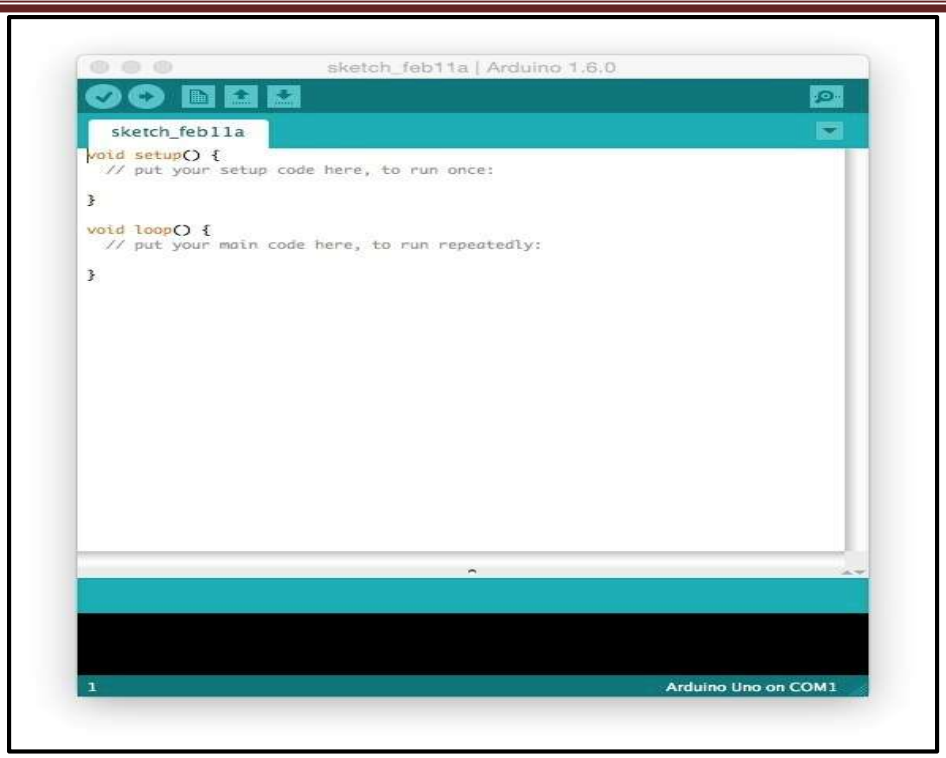


Fig. 2.5.2: ARDUINO IDE SCREEN

The sketch involves following options:

**[1]Verify/Compile**

This step checks the sketch for errors compiling it. It will report memory usage for code and variables in the console area.

**[2] Upload**

Compiles and loads the binary file onto the configured board through the configured Port.

**[3] Libraries**

We can add library to the sketch by inserting #include statements at the start of the code. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include

statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of the code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager.

## TOOLS

Tools that are needed while testing the codes are:

### [1] Serial Monitor

Serial monitor window initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

Displays serial data being sent from the Arduino or Genuine board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial. Begin in your sketch.

### [2] Board

Select the board as esp8266 while dumping the code to Wi-Fi module and select Arduino board while connected to Arduino UNO. The board selection sets the parameters like CPU speed and baud rate used when compiling and uploading sketches. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the boot loader.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, and Galileo and so on.

**[3]Port**

This menu contains all the serial devices (real or virtual) on your machine. It will automatically refresh every time you open the top-level tools menu.

**HELP**

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

**UPLOADING**

Before uploading sketch, we need to select the correct items from the Tools like Board and Port. The boards are described as COM1 or COM2 for a serial board and COM4, COM5 and COM or higher for a USB board. To find out this, we have to look for USB serial device in the ports section of the Windows Device Manager. Once selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. On most boards, contains the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error. When you upload a sketch, you're using the

Arduino boot loader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

## 2.6. Methodology:

First the code is loaded in to Arduino board then when car enter the entry gate then IR sensors 1 will sense car presence and opens the gate which is connected to DC servo motor 1 and at the exit gate there is another IR sensor 2 which will detect the car presence then opens the gate and display 1 in the screen when it get sense by car otherwise it displays zeroes and ultrasonic sensors

In this smart parking system project .The IR sensors along with DC motor are used in both entry and exit sides of the parking area for open & close the gate. The ultrasonic sensors are placed in the parking slots to detect the presence of vehicle in the slots .For example , If a vehicle parked in the slots then the LED of the ultrasonic sensors glows & the vacancies of the parking slots will shown in the display.

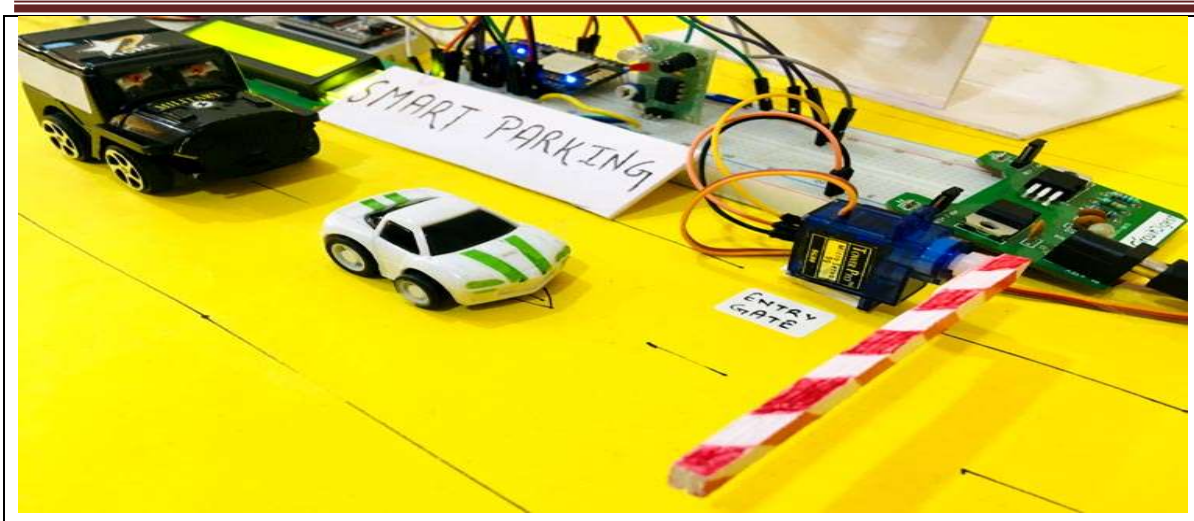


Fig. 2.6.1 Experimental setup.

**RESULT AND CONCLUSION:**

Smart Parking System on this project has been successfully made in accordance with the purpose of the smart parking system. One of the purposes is to know the condition of the parking lot through the IR sensors that is connected with the application in real-time wherever using applications based on Arduino board.

Through this prototype system we demonstrated that the proposed architecture can effectively satisfy the requirements of car park management system and we believe that wireless sensors can be promising technology to solve future parking hassles.

## Output:

In the entry gate the IR sensor 1 is connected along with DC servo motor when car enters the entry gate the IR sensor will sense the car and opens the gate in 90 degree after that the gate go back to its original position here ultrasonic sensors are used to detect the slots if the slots available it will display the slots

number in serial monitor once when car exit at the exit side then the slots number will increment.

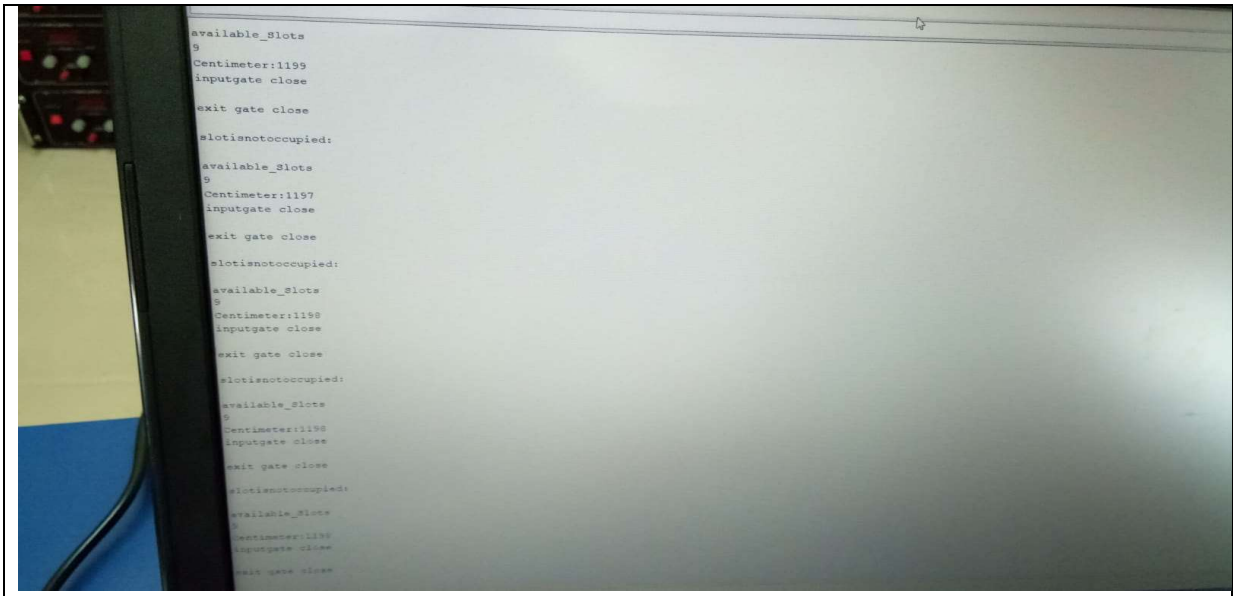


Fig.2.6.2 .output

## Conclusion:

The systems proposed by various authors help us effectively in reserving as well as eliminates the need for searching of a parking space in private parking lot. Many researchers have implemented systems which have dynamic arrangement scheme for satisfying the different needs of drivers and service providers, which is based on real-time parking information. Hence, we conclude that this is very useful for new researcher for innovation of new techniques to manage the problem faced by drivers on day to day basis. The table contains advantages and disadvantages of various systems implemented by researchers. In future work, we innovate this system which is not only used in a particular parking area available, but can be extended and also be implemented on various other platforms such as railway stations, airports, mall parking spaces. This will make the management of the parking spaces efficiently, by eliminating need of manual labor work.

## References:

[1] Lee C, Han Y, Jeon S, Seo D and Jung I 2016 “Smart parking system for Internet of Things”
IEEE Int. Conf. Consum. Electron. pp. 263–264.

[2] Khanna A and Anand R 2016 “IoT based smart parking system” Int. Conf. Internet Things Appl.
pp. 266–270.

[3] Basavaraju S R 2015 “Automatic Smart Parking System using Internet of Things (IoT)” Int. J.
Sci. Res. Publ. 5 (12) pp. 629–632.