International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

PARKING SPACE DETECTION USING INTERNET OF THINGS

Priyadharshini. V, Likhith Yadav. V, Dr. Y. Srinivas

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BACHELOR OF TECHNOLOGY
GITAM
(Deemed to be University)
VISAKHAPATNAM

ABSTRACT

The range of people that use their very own motors is usually rising. Car parking is one of the significant problems. Finding an available parking spot in a large area is burdensome and time-consuming. The evaluation of the Internet of Things has improved efficiency and minimized human effort in numerous fields. Our project aims to utilize the IOT and reduce the stress of searching for parking spaces. By analyzing and investigating parking space information measured by the sensors, dependable detection of enough parking spaces can be found. This document discusses the principle troubles in detecting parking spaces, illustrating the importance and cutting-edge research popularity of parking space detection techniques. Finally, it summarizes the parking area detection methods, which offer a clean path for future studies.

Chapter 1

INTRODUCTION

In recent years the number of people who prefer their vehicle to public transport is increasing rapidly. Modernization plays a significant role in the high car-possessing trend. People often choose a car because of its flexibility and security. The number of cars on the road grows daily, leading to new problems such as high traffic jams, pollution of the air and more carbon footprints. We do not get to control people from choosing their mode of transportation. However, we can reduce its contradictions by taking practical steps. One of the major problems car owners face is finding a parking spot. Let it be a mall, movie theatre, park, or any place finding a space to park the vehicle is tiresome. A recent study says drivers take almost 17 hours annually to find a parking space. Consider the fuel they burned and the stress people would have endured finding a parking spot. It is necessary to take action on the above said.

1.1 DIFFERENT PARKING SYSTEMS

To make it simpler, we can utilize the different parking systems. Similar to having different parking ways, like angular and perpendicular parking, numerous parking management systems are available to avoid this situation. One can build software to maintain the log of the incoming and outgoing cars and calculate the total space available for parking. Installing a number plate recognition camera will help to make it automatic. An online portal may be used to pre-book the slots avoiding overlap.

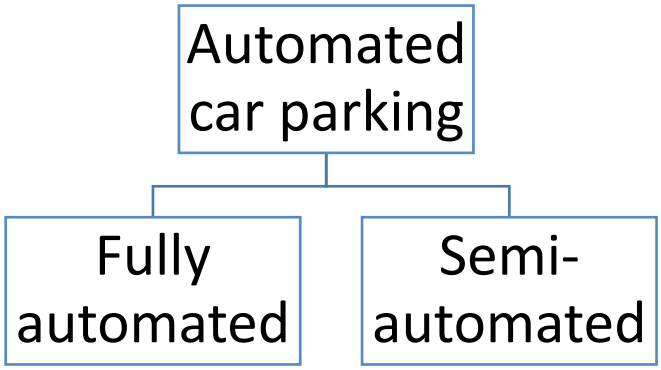

Figure 1.1 classification of Automated parking system

With machine learning and IoT advancement, People have developed different parking systems based on their requirements. Automated car parking has received a grand welcome from the parking lot owners. Automated car parking is a mechanical system that stacks vehicles vertically, thereby limiting space and time. It is classified into two forms, as shown in figure 2.1. The difference between the two is that automated car parking requires an attendant, while semi-automated parking does not require one. Different parking systems fall into automated and semi-automated car parking. Puzzle parking system, as the name implies, the cars are arranged like a puzzle park vertically and horizontally(Figure 1.2). The ground level is said to be the entry and exit.

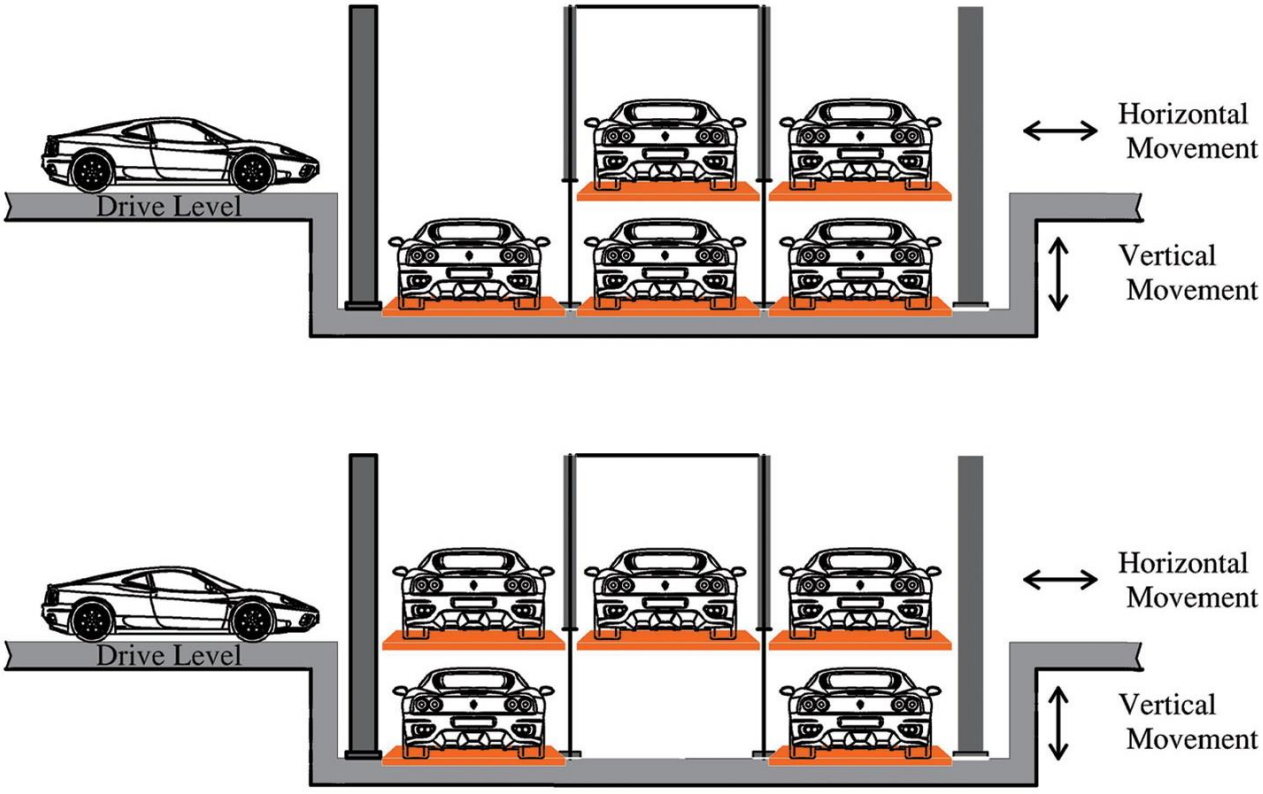

Figure 1.1.2 puzzle car parking

A Radio Frequency Identification parking system uses radio waves to track and identify objects. A tag will be attached to the vehicle's windshield, which a camera will read at the entry and exit points. RFID is more secure and prevents unauthorized users from entering the parking area.

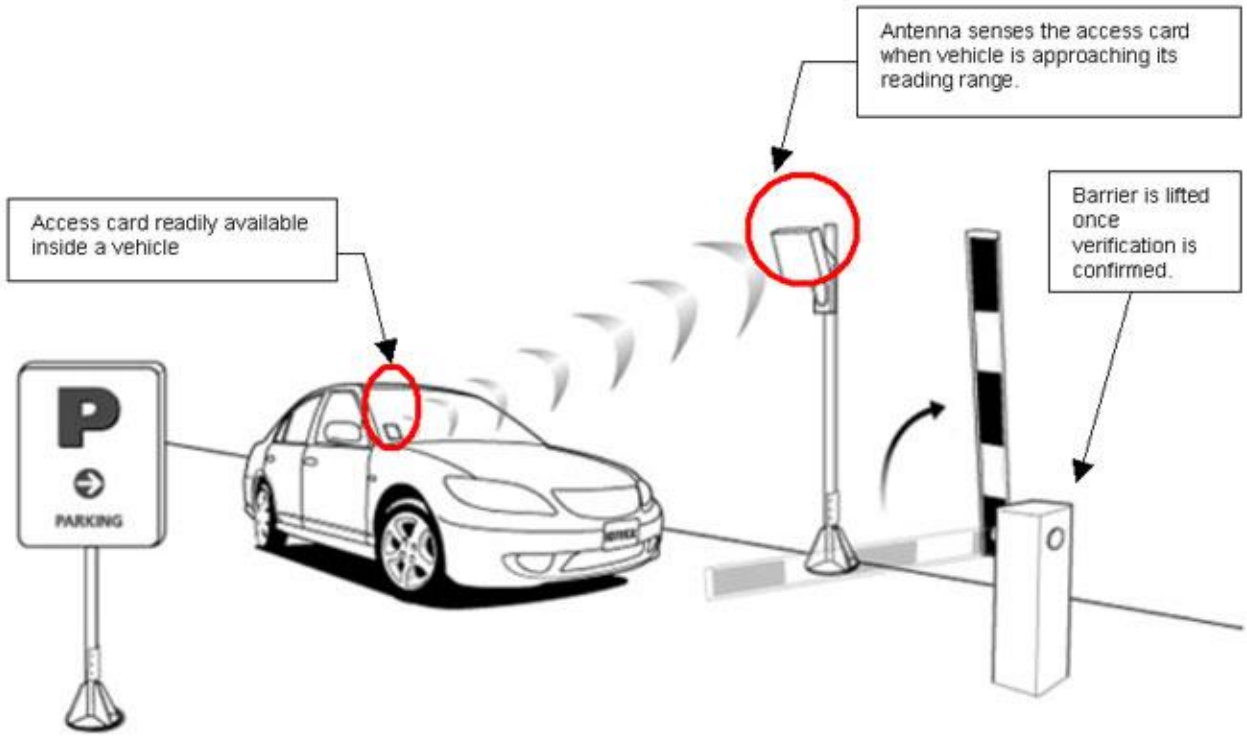

Figure 1.1.3 RFID Parking system.

A rotary parking system can fill six cars in place of two. The cars are rotated clockwise in the bay to load and unload them. The rotary parking system is customizable and can be accumulated to different requirements.

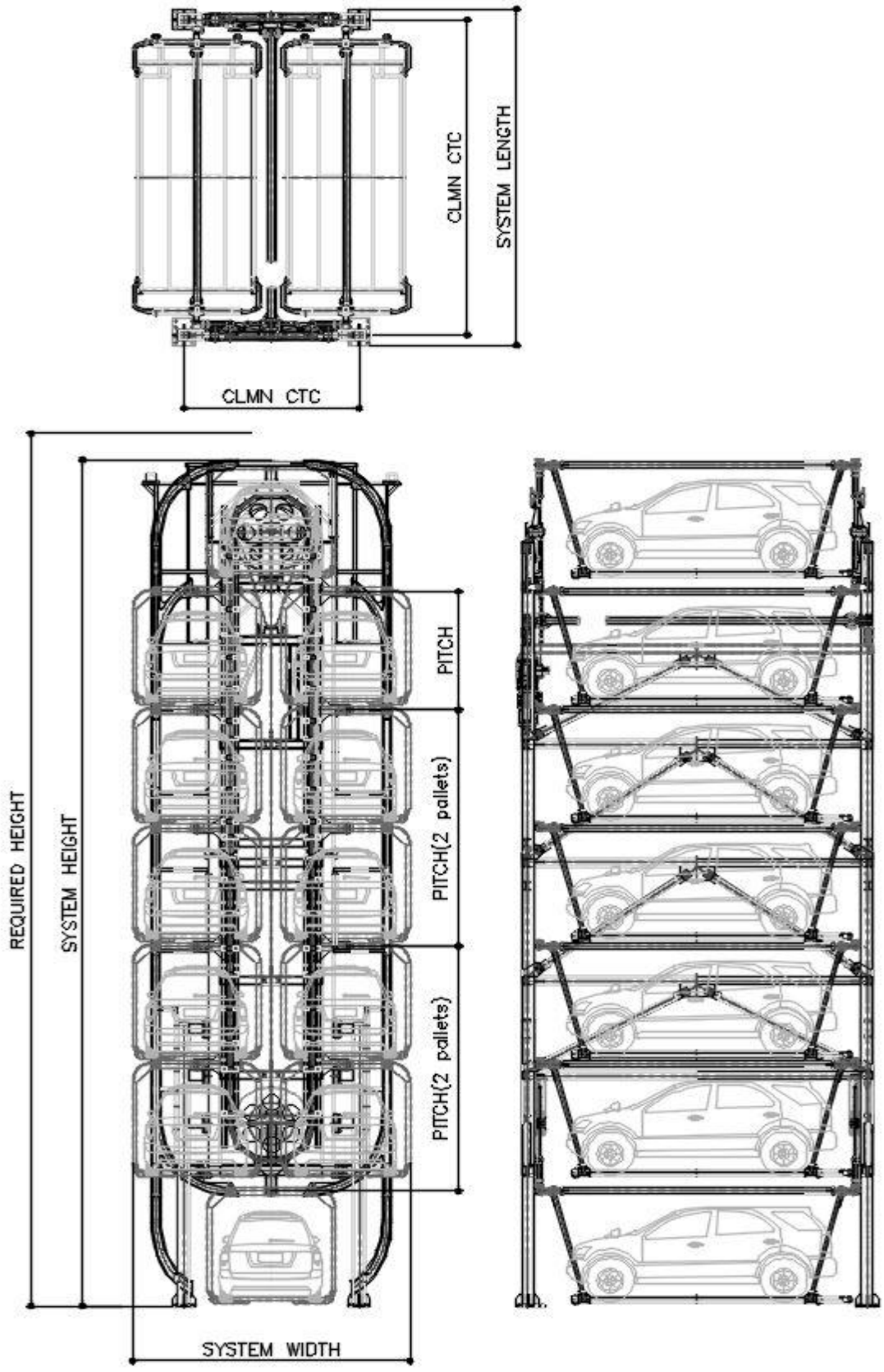

Figure 1.1.4 Rotary parking system

Parking Space can also be detected using CNvison and training models with data sets. Training the model by identifying cars first and then with empty slots. The model is developed in such a way that it can identify different vehicles. Now, intersecting both, we get a machine-learning model that can detect parking space using datasets. The support vector machine algorithm indicated a detection rate of 94%. The k-nearest algorithm indicated a detection rate of 89%. The proposed stage has different stages, from collecting videos, sample training, and testing new datasets. The detection rate for the laser range finder was 76.9%.

Figure 1.1.5 Parking detection using machine learning

These are a few examples of advanced parking systems followed around the world. Large shopping malls, Giant business centers, auditoriums, and stadiums can significantly use it. When we consider medium-sized businesses, schools, hospitals, and universities, have larger parking spaces, the problem will be finding them. Of course, we can use the above-said systems, but it will be expensive and takes time to set up, which might disturb their regular environment and causes inconvenience. Before construction, they have to plan to install these systems. Installing it on an existing building or area may cause unwanted interruptions.

Chapter - 2

LITERATURE REVIEW

Al Qaidhi [1] The idea of using automated parking systems was proposed in the past to address the issue of limited parking space, particularly in crowded areas where parking spots are scarce. These systems typically feature multiple levels of parking that are stacked vertically, allowing for more vehicles to be parked within a smaller area, similar to what is observed in multi-story car parks. Their paper compares the intelligent parking system with the traditional one, a more user-friendly, cloud-based environment. It also explained the scenario in UAE and Japan. It recommends a four-layer parking system. They are sensing, networking, middleware, and application layer. It also proposes a framework for the parking system.

Thorat[2], This research paper suggested using an RFID system. The person has to register using the mobile application, and the server will store the information. The user can pre-book the slot. The vehicle is attached to the RFID tags. When the vehicle is parked in a particular slot, the RFID verifies whether it is the respective user and alerts the breach. When the user leaves, the parking charge is sent to the user's mobile. The system minimizes the parking waiting time in a large-sized parking facility. It also helps in maximizing their venue generation for the parking facility owners. It would also help reduce the need for a workforce in the parking facility, significantly reducing the cost and errors in the process. Also, this method would minimize paper usage, ensuring a green system.

[3] Jae Kyu Suhr has developed a parking system that uses ultrasonic sensors to detect parking spaces, which are divided into four stages with parallel line pairs marking each slot. The system employs a low-level fusion of AVM images and sensors to locate free spaces and detect pillars, resulting in more convenient parking for users. However, the system faced challenges due to the need for cameras on every side of the vehicle, which made predicting parking space and tracking empty slots more time-consuming.

[4] ThanhNamPham has proposed an algorithm that enables users to find the parking space with the lowest cost and suggests a new parking spot if the current one is full. The algorithm calculates metrics based on distance and the total number of free spaces. Using RFID technology, the system can be implemented at a low cost and on a large scale. It helps prevent parking disputes and saves time by providing an efficient way

to find a parking space. Once the user logs into the system, they can select an appropriate parking space and receive a notification confirming their choice. The system then sets the status of the selected space to "pending," preventing other users from reserving it. If the system does not detect a parked car in that space after a certain period, it changes the status back to "available."

VishwanathY[5] proposed a parking system that detects parking slots using parallel lines arranged in a hierarchical tree representation. Each slot is connected to a GSM communication network via a sensor. An LED screen displays the number of available and vacant slots. In an emergency, a person can reserve a parking space from a specific distance using a mobile phone. This system provides significant benefits to vehicle drivers by saving time and allocating reserved parking spaces, effectively reducing their workload at a lower cost.

Chapter - 3

PROBLEM IDENTIFICATION AND OBJECTIVES

Real-time detection of available parking spaces and communication with users can significantly reduce wait times, increase scalability, and make parking lot navigation more efficient. Effective parking lot management that displays parking division information to drivers before entering the lot is crucial. However, implementing an automated car parking system in an existing setup can be costly, time-consuming, and challenging.

The main aim of this project is to make parking slot detection simpler and of less cost compared to RFID and other systems. When pre-booking the slot, if the user does not keep up with his allocated time, it causes inconvenience for other users. When the user does not show up after pre-booking the slot, it wastes space. It is essential to consider these facts when coming up with the project. Here are its main objectives

1. To develop a parking space detection system using the Internet of Things
2. A parking system which will not be affected by weather and can be accessed even during nighttimes
3. Identifying parking spaces with less delay
4. A parking system that is simple and which does not require the long process of registration, signing up, and receiving a One-time password every time.

By achieving these objectives, the developed parking system can help prevent long queues and reduce fuel consumption, reducing carbon footprint. It also reduces the need for staffing. The system would also reduce paper usage, providing a more eco-friendly environment.

Chapter - 4

SYSTEM METHODOLOGIES

Upon arrival at the parking area gate, the display shows the number of available slots, with a green LED indicating available slots and a red LED indicating no availability. Once a vehicle enters a parking slot, the display indicates that the slot is occupied. If there are no empty slots available, the system displays that all slots are full.

4.1 BLOCK DIAGRAM

A block diagram is a graphical representation of a system that uses labeled blocks to represent individual or multiple entities, items, or concepts. These blocks are connected by lines to illustrate their relationships and interactions with each other. Block diagrams are commonly used to design and develop new systems, as well as to analyze and improve existing ones. They provide a top-level view of the major components of a system, the key participants involved in its processes, and the critical relationships between them.

A block diagram is a useful tool for visually representing the functional view of a system. It uses blocks to represent system components, and lines to connect the blocks, showing their relationships. Block diagrams are commonly used in software design and engineering systems to depict the critical parts of a system and the flow of data in a process flowchart. This high-level overview allows for easy communication and understanding of the system's essential components and processes.

The block diagram(figure 4.1.1) shows an Arduino, LCD, and Infrared sensors and the module used to connect that. The I2C module is used to make four pins utilizing the sixteen pins.

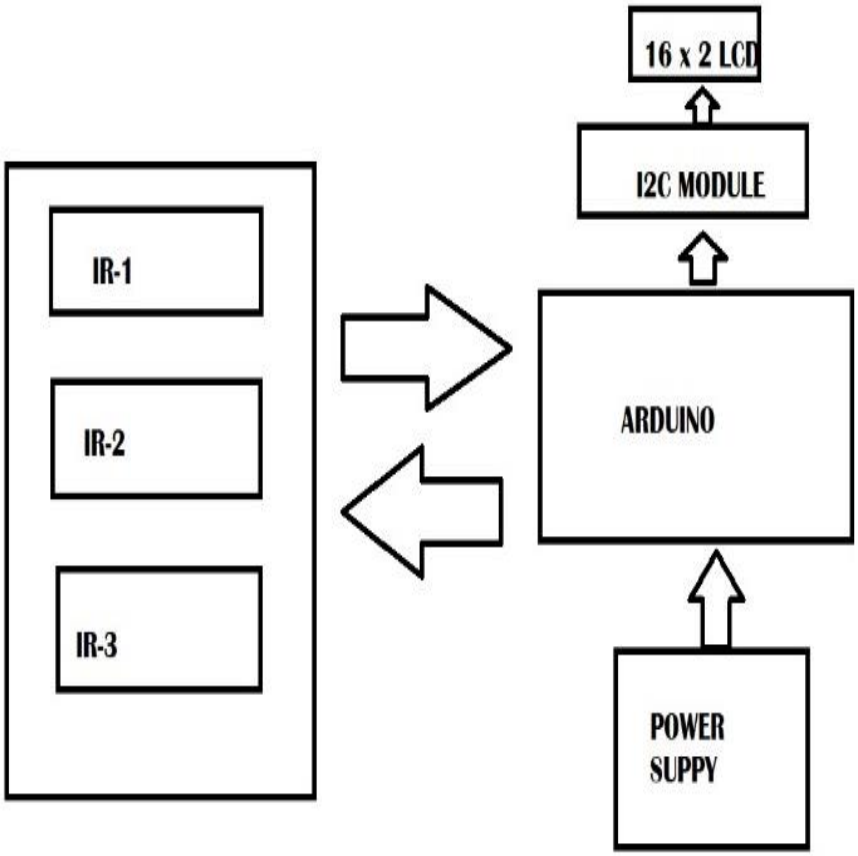

Figure 4.1.1 Parking system detection block diagram

4.2 UML DIAGRAM

UML stands for Unified Modeling Language, which is a modern approach to software modeling and documentation widely used in business process modeling. It uses graphical representations to depict software components, and there are various types of UML diagrams, serving different purposes, regardless of whether they are designed before or after implementation. The two broad categories of UML diagrams are Behavioral and Structural diagrams, with other types falling under these categories. While some UML diagrams focus

on analyzing and illustrating the structure of a system or process, others describe the system's behavior, actors, and building components.

A Use Case diagram is a useful tool for analyzing the high-level requirements of a system. These requirements are expressed through various use cases, which describe the functional requirements of the system using a verb to denote an action. Actors are another important component of the Use Case diagram, representing the various entities that interact with the system, including human users, organizations, and external or internal applications. The relationships between actors and use cases are depicted using straight arrows, providing a visual representation of how different actors interact with the system's functionality.

Here we represented the car driver and the display in the use case diagram. As soon as the car driver enters the parking area, he looks at the display and learns where the spot is available for him to park the car. The slots are represented as S1, S2, S3, and S4. The availability of them is given straight after a colon. The sensors provide this information to the display through Arduino. A Led setup also glows based on whether the car is parked. If the car is parked, the red led glows, or the green led glows. Figure 4.2.1 is the use case diagram designed for this project

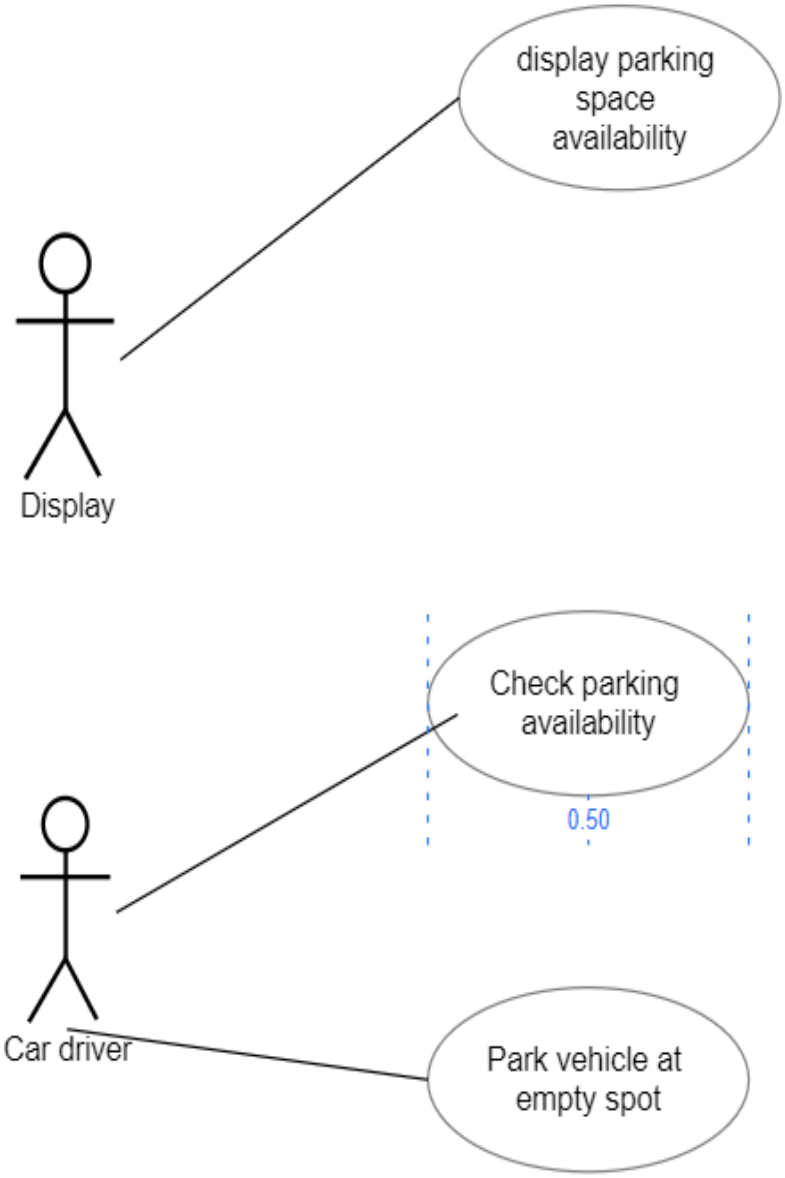

Figure 4.2.1 Parking system detection use case diagram

4.3 FLOW CHART:

A flow chart is a graphic representation that outlines a process using symbols to represent each step, along with a brief description of the action taken at each step. The symbols are connected by arrows to indicate the flow direction of the process. Each symbol in a flow chart represents a specific action or decision, with its unique meaning. Some of the most commonly used flow chart symbols include:

A flowchart is a graphical representation of a process that uses different symbols to represent each process step and briefly describe it. The symbols are connected by arrows indicating the direction of process flow. The most common flowchart symbols include the Terminator, which is an oval shape used to indicate the start or end of a process; the Process, which is a rectangular shape representing a typical process flow step; the Diamond, which is used to denote a branch in the process flow; the Connector, which is a small, labeled circular shape used to represent a break in the process flow; the Data, which is a parallelogram that represents the input or output of data in a process; and the Document, which is used to denote a report or document. The flowchart for our parking space detection system is shown in the diagram 4.3.1

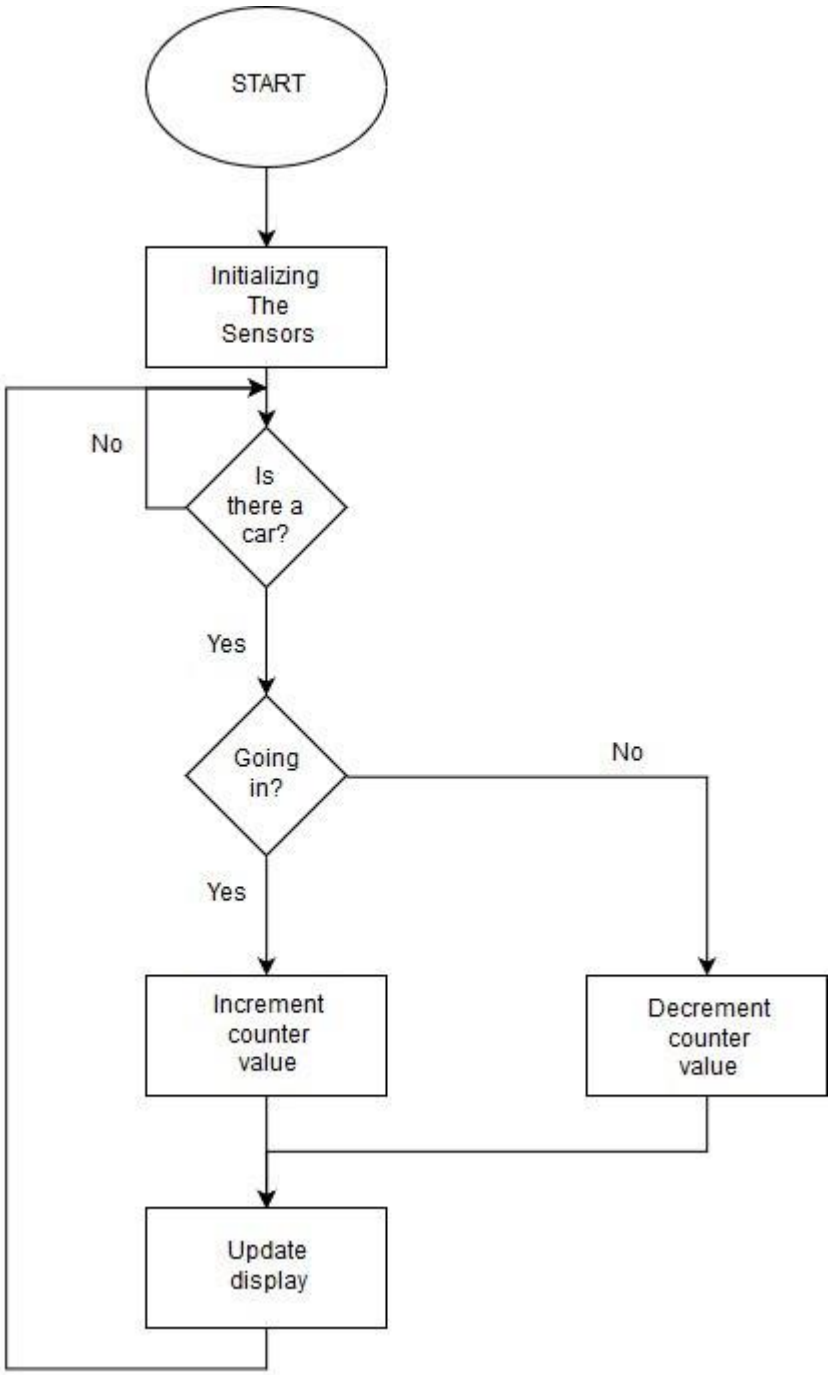

Figure 4.3.1 Parking system detection flowchart

Chapter-5

OVERVIEW OF TECHNOLOGIES

5.1 ARDUINO:

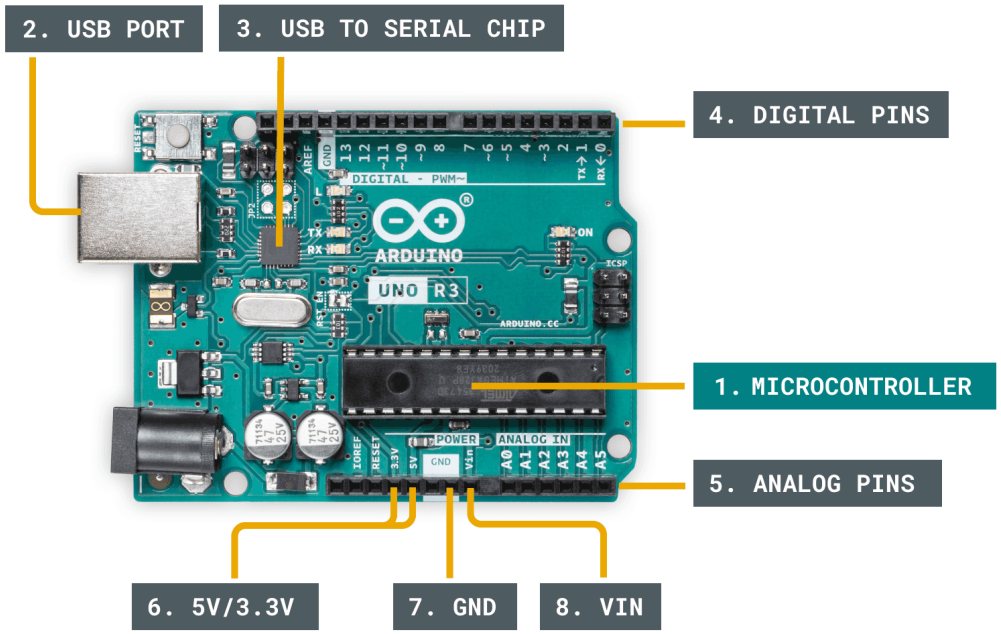

Figure 5.1.1 Arduino board

Arduino is a tool that helps to control electronic stuff with code. Its input is to gather information, and its output is some action to the world. For example, the input device includes light, touch, humidity, and distance sensors. The output will be liberated over DC motors, Servo motors, LED strips, and LCD. Arduino stands between the input and output devices. It consists of plastic holes known as pin headers. Each one makes an electronic connection to the pin in the microcontroller. Pins are through which the Arduino communicates with the world. Pins 0 and 1 are used for communication with the USB port. 1 to transmit and 0 to receive. The pins that are marked with ~ represent pulse with modulation. It has analog 6 pins. The Arduino board has several pins that can be used as digital input or output, including pins 4, 6, 8, 9, 10, and 12. Additionally, there are Serial Pins, also known as UART pins, that enable communication between the Arduino board and other devices, such as a computer

GND- ground

V_{in} – external voltage source

5v/3.5 v- voltage source for small components

Arduino uses Arduino Integrated development environment to perform tasks. It is more like C and C++ programming. The IDE looks like in figure 5.1.2

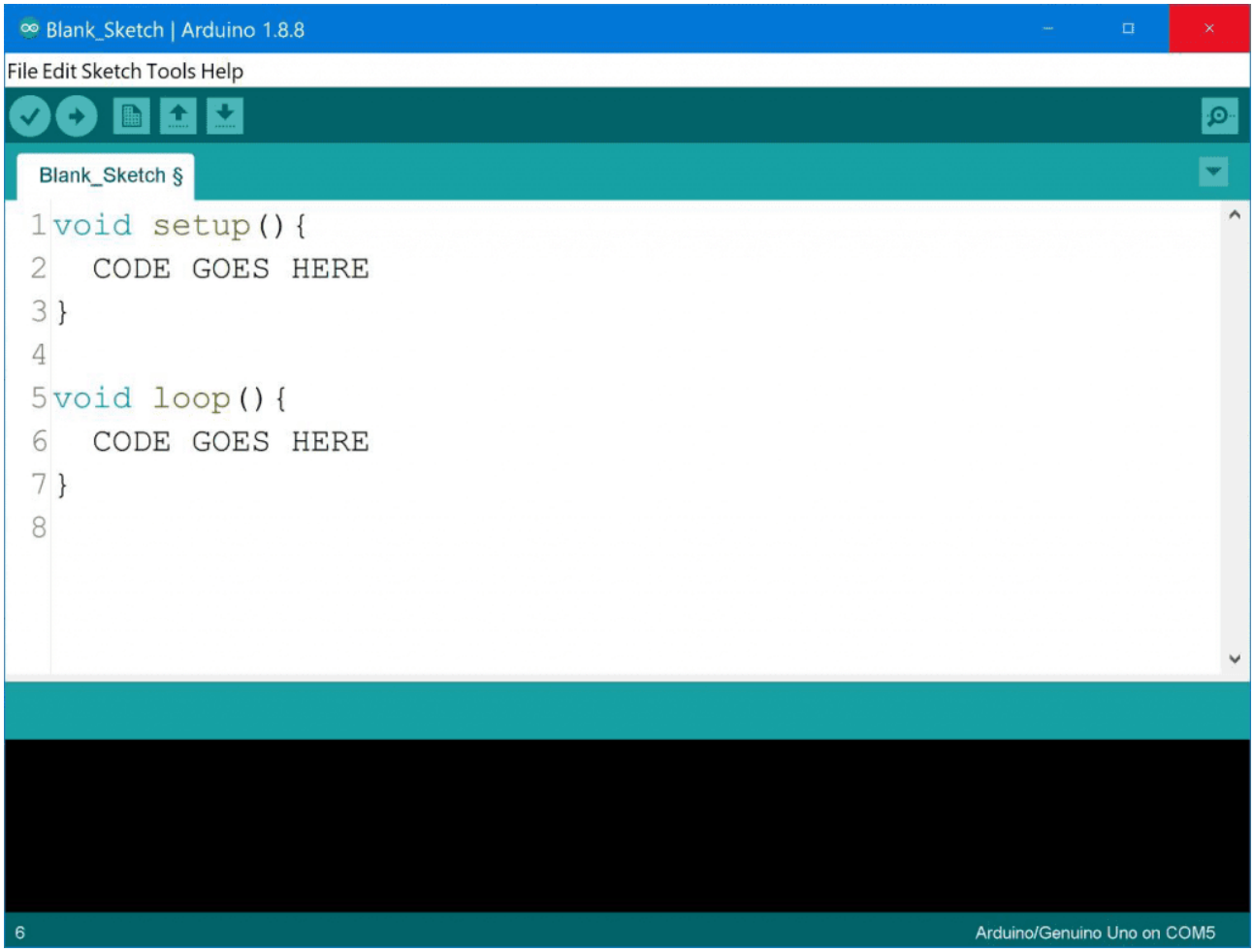

Figure 5.1.2 Arduino IDE

The toolbar in the Arduino software displays icons for various functions such as creating a new project, opening an existing one, saving changes, uploading the code to the board, and verifying the sketch for errors.

To upload the code, it is essential to select the correct board and ports, and connect the board to the computer using a USB cable. After completing these steps, click on the Upload button in the toolbar, which compiles and runs the code, and uploads it to the connected board. In some newer boards, the reset is done automatically, while in older ones, we need to press the reset button. If there are any errors during uploading, the software displays them in the error window.

The toolbar in the Arduino IDE provides several buttons, including New, Open, Save, Verify, and Serial Monitor. The Open button allows users to open a previously created file, while the Save button enables users to save the current sketch or code they are working on. The Verify button is used to check for any compilation errors in the sketch. Additionally, the Serial Monitor button can be found on the right corner of the toolbar, which opens the serial monitor for communication between the computer and the Arduino board. The Tools section of the IDE provides additional features, such as the ability to select the board type and serial port.

Auto Format

The Auto Format button in the Arduino IDE toolbar is a useful feature that helps to format the written code automatically. It ensures that the code is properly structured, such as lining up the open and closed curly brackets in the code. This feature saves time and makes the code more readable and easier to understand.

Archive Sketch

The Archive Sketch button allows the user to create a backup of their current sketch or code in a compressed .zip format. This file is stored in the same directory as the original sketch.

Fix Encoding and Reload

The "Fix Encoding and Reload" button is used to resolve inconsistencies between the character maps used by the operating system and the character map encoding used by the editor. This can happen when a file is opened or imported into the editor from another source, and the character encoding is not compatible with the editor's settings. The button can be used to automatically detect and adjust the character encoding, allowing the file to be displayed and edited correctly in the editor.

Manage Libraries

The "Manage Libraries" option displays an updated list of all the libraries installed in the Arduino IDE. It also provides the option to install a new library into the IDE.

Serial Monitor

The Serial Monitor feature enables the communication and exchange of data between the computer and the connected board via the port.

Serial Plotter

The Serial Plotter button is a built-in feature of the Arduino IDE that allows for the display of serial data in a plot. It provides a graphical representation of data received from the connected board on the port.

WiFi101/WiFiNINA Firmware Updater

The WiFi101/WiFiNINA Firmware Updater is a tool in the Arduino IDE that enables users to verify and update the Wi-Fi firmware of the connected board. This is particularly useful for boards that have built-in Wi-Fi modules such as the Arduino WiFi101 or the WiFiNINA. Keeping the firmware up to date can ensure that the board is running optimally and is compatible with the latest features and technologies.

Board

The Board option on the Arduino IDE allows users to select the appropriate board for their project from a list of supported boards. It is crucial to ensure that the selected board matches the board currently connected to the computer to ensure that the code is compiled and uploaded correctly.

Processor

The Processor option displays the available processors compatible with the selected board. This option refreshes every time a different board is selected.

Port

The Port option in Arduino IDE displays a list of all virtual and real serial devices available on the computer. It allows the user to select the port to which the Arduino board is connected for serial communication. The

option is refreshed every time a board is selected, and the user needs to ensure that the correct port is selected before uploading the sketch or code to the board.

Get Board Info

The Get Board Info option provides information regarding the selected board, and we must select the correct port to access this information.

Programmer

Selecting the hardware programmer is necessary when programming the board if the onboard USB serial connection is not in use. The programmer is also needed when burning the bootloader.

Burn Bootloader

The Burn Bootloader option is used to install or update the bootloader on a microcontroller present on the board. It is helpful when we have a microcontroller without a bootloader or need to update it. Before using this option, we must ensure the correct board and port are selected in the Arduino IDE.

5.2 INFRARED SENSOR

Infrared (IR) sensors detect and measure infrared radiation emitted from objects. Some examples of IR sensors include pyroelectric detectors, bolometers, and thermocouple-thermopile sensors. These sensors are generally classified into two types: thermal and quantum. IR sensors can detect changes in temperature and motion, making them useful in various applications. The IR sensor (5.2.1)and its working principle are illustrated in figure 5.2.2

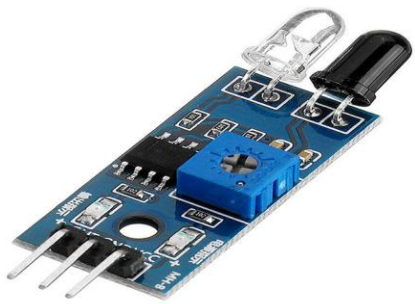

Figure 5.2.1 Infrared sensor

In an IR sensor, an IR LED (Light Emitting Diode) is used as the emitter, while an IR photodiode is the detector. These sensors are designed to detect the same wavelength of infrared light that the IR LED emits. The resistance and output voltage of the photodiode will vary proportionally to the intensity of the IR light received. A typical infrared detection system comprises five main components: an infrared source, a transmission medium, optical components, infrared detectors or receivers, and signal processing. Infrared sources can include infrared lasers and LEDs emitting a specific light wavelength. The primary types of infrared transmission mediums are vacuum, atmosphere, and optical fibers. Optical components are used to focus the infrared radiation or light.

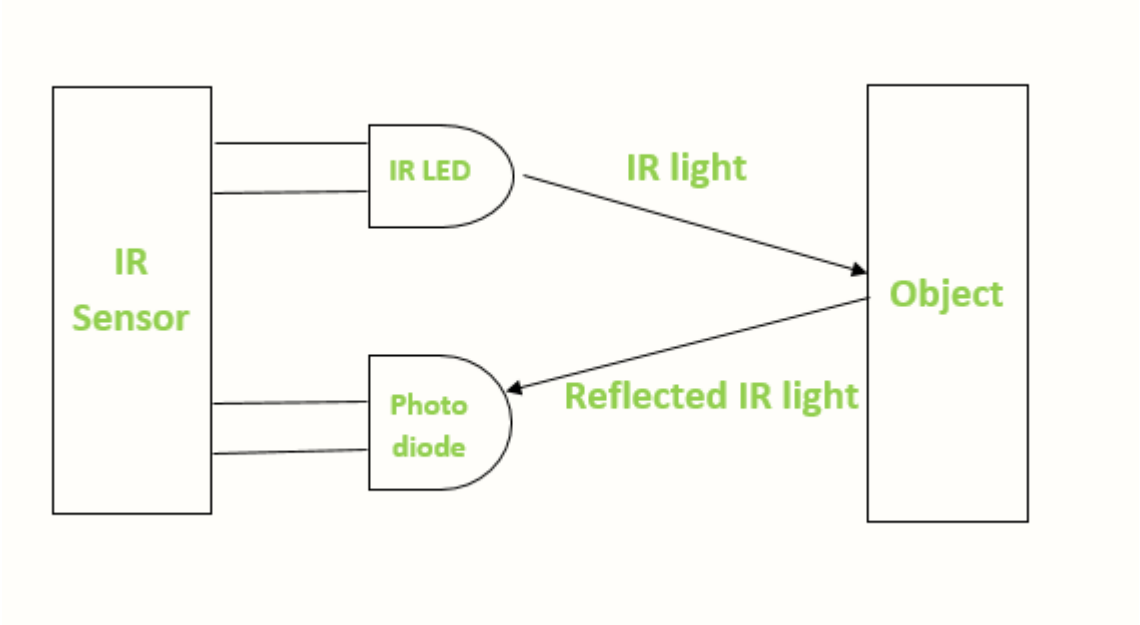

Figure 5.2.2 IR receiver

IR receivers are available in different types based on wavelength, voltage, package, etc. When combined with an infrared transmitter, the receiver's wavelength should match the transmitter's. In an IR transmitter-receiver setup, the emitter is an IR LED, while the detector is an IR photodiode. The photodiode can detect the same wavelength of infrared light that an IR LED emits. The resistance and output voltage of the photodiode varies proportionally to the intensity of the IR light it receives. This underlying principle is what makes IR sensors work.

5.3 LIQUID CRYSTAL DISPLAYS

The LCD (Liquid Crystal Display) is a type of display that utilizes liquid crystals for its operation. In this scenario, we will receive serial input from the computer and upload the sketch to the Arduino. Once uploaded, the characters will be displayed on the LCD screen. The characters will be displayed on the LCD.(figure 6.3.1)

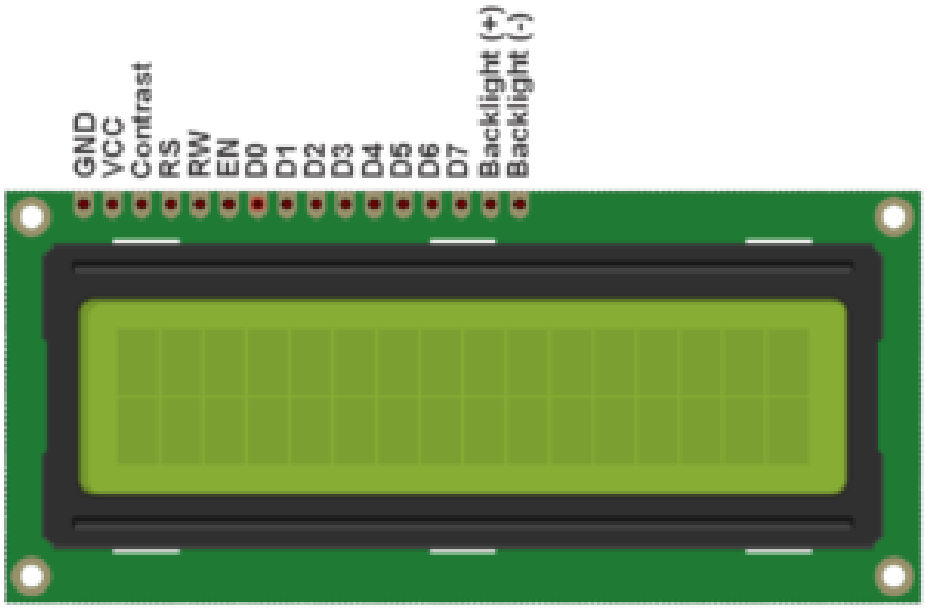

Figure 5.3.1 LCD

The Liquid Crystal Display (LCD) is a type of display that utilizes liquid crystals to operate. It is typically controlled through a parallel interface, meaning the microcontroller uses multiple pins at once to control the LCD. The process involves loading data into the data registers displayed on the LCD screen, with the instruction register used to store instructions received from the Register Select. Thankfully, displaying characters on the LCD has been made easier using the liquid crystal library.

5.4 LIGHT EMITTING DIODE

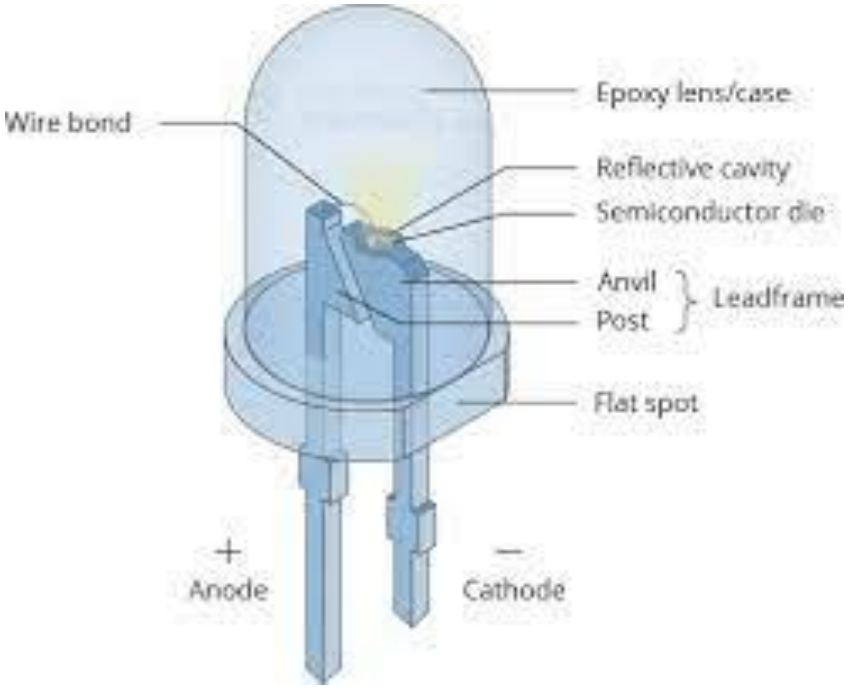

Figure 5.4.1 Light-emitting diode

The LED (Light Emitting Diode) combines electrons and holes within the semiconductor material, producing light. These particles are confined within energy bands in the semiconductor material. The LED's emitted light depends on the photon energy, determined by the distance between the bands or bandgap. Different semiconductor materials have different bandgaps, which generate different colors of light. By altering the active region's composition, the precise wavelength or color of the emitted light can be adjusted. In this project, red and green LEDs are used.

Aluminum gallium arsenide (AlGaAs) is commonly used to produce red and infrared LEDs, while gallium phosphide (GaP) is used to create yellow and green LEDs.

Chapter-6

PARKING SYSTEM DETECTION IMPLEMENTATION

- Identifying the parking spots using Infrared sensors
- Each slot is equipped with the IR-sensor
- The IR sensor, which is connected to the Arduino, which in turn is connected to the LCD, shows the availability
- When the car enters the parking area and gets parked in one of the slots, the respective sensor will send a signal to the Arduino (1-0).
- The signal will be interpreted and displayed on the LCD

After compiling every part, following the circuit diagram, and programming the Arduino board. Now, precisely position the sensors and servo motor. This project has four parking spaces, and the IR sensors 1, 2, 3, and 4 are situated at slots 1, 2, 3, and 4, respectively. LCD screen is situated close to the entrance gate. The system used IR sensors 1, 2, and 3 to detect whether or not vehicles were arriving at the gate and IR sensors to determine whether or not the parking space was empty. Initially, all the slots are empty, so it displays "none" on the LCD. After entering the parking area, the LED display shows that the vehicle is "full" when occupying a slot. When the vehicle leaves, the IR sensor will detect it and display it as "none." The figure 6.0 shows our system architecture.

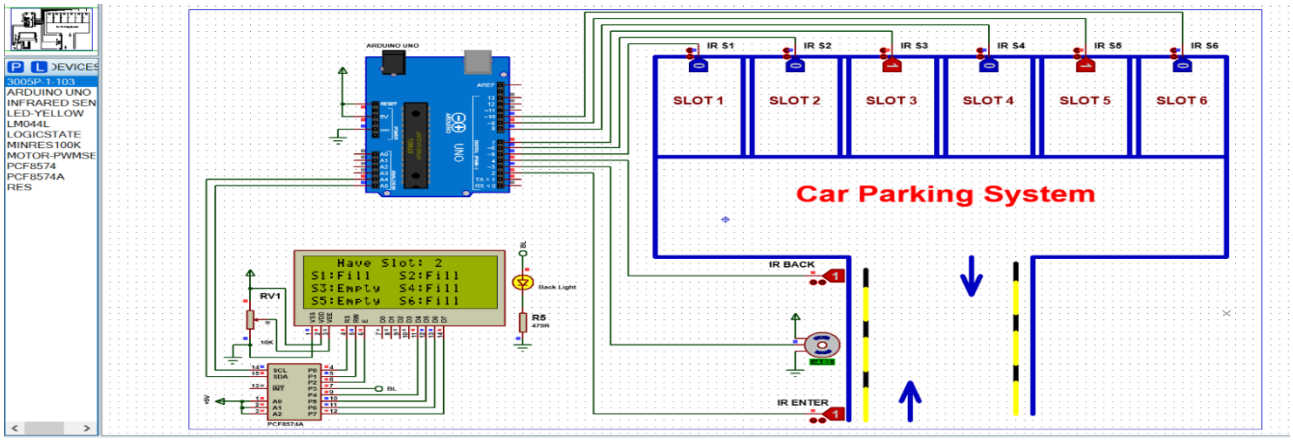

Figure 6.0 Parking System architecture

6.1 CODING

We use the wire and I2C modules to enable communication with I2C devices. The wire library is essential for controlling data transfers in master mode. It also allows for communication between the Arduino and LCDs that use the Hitachi HD44780 chipset, commonly found on text-based LCDs. The library supports 8-bit and 4-bit modes, with the bit mode determining the data lines and the rs and rw control lines. We can initialize the LCD using the begin() function, and the set cursor() function is used to move to the next position on the display. The IR ow signal indicates the presence of an object, meaning the slot is filled, and vice versa. IR ow means the object is present, i.e., the slot is filled and vice versa.

6.2 TESTING:

The parking lot is installed with sensor for each slot. The LCD display is placed in front of the the parking space. When all slot are empty the green\ LED glows. When a car enters the slot the green LED will go off and the red LED will glow. This will be received by the LCD through the Arduino and the LCD displays that the specific slot is “FULL”. The other slots will be “NONE”

Test case 1: Initially all slots are empty so green LED in all slots glows. The LCD shows that S1: NONE; S2:NONE; S3: NONE; S4:NONE. When the car enters a slot, say S1, then the respective slot’s red LED glows and the LCD display S1: FULL.

Test case 2: Here all slots are occupied. So, The red LED glows and the LCD shows S1: FULL; S2:FULL; S3: FULL; S4:FULL. When the car leaves a slot, say S1, then the respective slot’s green LED glows and the LCD display S1: NONE.

Test case id	Test case name	Description	Precondition	Postcondition	Input	Output
1	Case 1	Car is entering the slot	All the slots are empty. All green LEDs were glowing	A single car has occupied the spot. The display shows that the particular lot is full	The car	The red led glow in the spot of the occupied space

2	**Case 2**	**All slots are occupied**	**All slots are filled with cars, and the display shows full for all slots.**	**The display shows that, except for the empty spot, all other slots are full**	**Car is leaving**	**The respective slot's green light glows, and the red led goes off**

Table 6.2.1 Parking system testing

Chapter - 7

RESULTS AND DISCUSSIONS

The LCD module allows the user to check the status of parking vacancies. When the IR sensor detects high, the parking spot is empty; when it detects low, the spot is occupied. When a car enters the parking space, the red LED lights up, indicating that the spot is occupied. When the space empties, the green LED lights up, indicating that the spot is available.

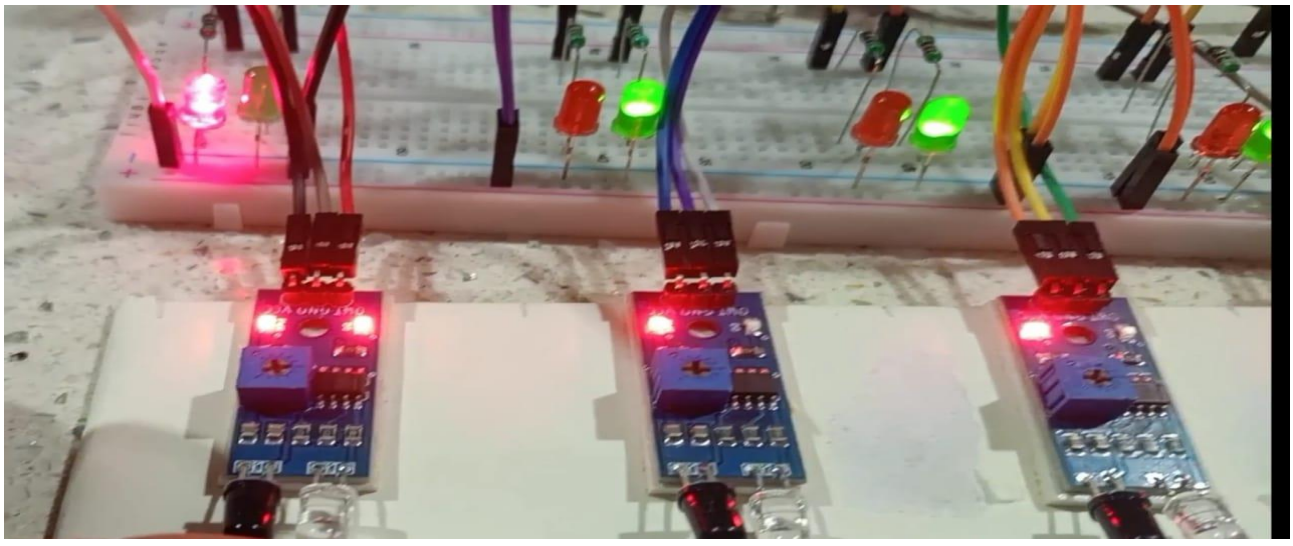

Figure 7.1 LED glowing

Figure 7.2 LCD displaying the status

Figure 7.1 and figure 7.2 shows the final output of the project. Here, slot S1: "full" means the parking space is occupied. "None" means it is empty. We divided the 16 matrix display into 4 using setcursor () during coding.

Chapter-8

CONCLUSION AND FUTURE SCOPE

The availability of vehicles is expanding quickly, which means that vehicle usage will rise year over year. In order to avoid congestion, we, therefore, need more parking space. If we know where there are parking spaces, we can make reservations in advance as needed. Therefore, we introduce our internet of a things-based system for detecting vacant parking spaces. Parking cars improperly and damaging other cars while doing so are the two main problems.

With more cars on the road, parking becomes a severe problem in urban areas. Finding a parking spot wastes time, energy, money, and time. Our project aims to develop a system to detect vacant parking slots and help users find available spots easily. We display the vacancy status on an LED display outside the parking lot to achieve this. Users can check the display before entering the parking lot to see if any spots are available. To make the parking process more efficient, we use a panorama image to track the available spaces and select the best one. This is particularly useful when multiple spaces are available. In the future, we can also develop a simple inner navigation diagram to lead us back to the vehicle. The above said can be executed by developing an app or portal. With the help of machine learning, we can advance this project by setting up a camera for computer vision. Furthermore, train a model based on the requirement.

REFERENCES

Conference:

[1] *Smart Parking System using IOT*, Waleed Zahir Al Qaidhi, Muhammad Sohail, Fourth research conference of Middle East College Muscat, Sultanate of Oman.

Journals /Articles:

[1]. Parking detection method using quadtree decomposition analysis,
Khaled Shaaban a, Houweida Tounsi, *Journal of Traffic and Transportation Engineering*

[2]. Prof.S.S.Thorat et al., *"IoT Based Smart Parking System Using RFID," International Journal Of Computer Engineering In Research Trends*, 4(1):9-12**, January-2017**

[3]. *Automatic Parking Space Detection and Tracking* by Jae Kyu Suhr 2016

[4]ThanhNamPham1,Ming-FongTsai1,Duc Bing Nguyen1, Chyi-Ren Dow1 and Der- Jiunn Deng2. *"A Cloud-Based Smart-Parking System Based on Internetof-Things Technologies."* IEEE Access**, volume 3, pp. 1581 1591, September 2015**

[5] VishwanathY, AishwaryaDKuchalli, DebarupaRakshit. "*Survey paper onSmart Parking System based on Internet of Things*."Information Science, New HorizonCollegeof Engineering, Bengaluru

Books:

- *Developing Future Smart Parking Solutions for Hangzhou's IoT Town*, Evan R. LeBeau, Josue David Contreras Albuja, Michael David Altavilla, Shuxing Li, **Worcester Polytechnic Institute**

Image references:

Figure 1.1.2 www.justdial.com/jdmart/kolkata/Puzzle -Parking-System-PPs-Pit-2xn

Figure 1.1.3 www.fingertec.com/newsletter/HL_76.html

Figure 1.1.4 www.anupriyaultratech.com/automatic-parking-system

Figure 1.1.5 www.easychair.org

Figure 4.3.1 www.github.com

Figure 5.1.1 www.arduino.cc

Figure 5.1.2 www.arduino.cc

Figure 5.2.1 www.diyables.io

Figure 5.2.2 www.robu.in/ir-sensor-working

Figure 5.3.1 www.pinterest.com/pin

Figure 5.4.1 www.lastminuteengineers.com/light-emitting-diode-led

Figure 6.0 www.github.com/anmspro/Smart-Car-Parking-System