Major Project Report
On

"IOT IN VEHICLE PRESENCE DETECTION OF SMART PARKING SYSTEM "

Submitted in Partial Fulfillment
of the Academic Requirement for the Award of Degree of

BACHELOR OF TECHNOLOGY
in
Electronics and Communication Engineering

Submitted By

DYAWARASHETTY SAKETH	**20R01A04K5**
DULAM SRIJA	**20R01A04K6**
GOLLENA VINAY	**20R01A04K9**

Under the esteemed guidance of

Dr.K.Niranjan Reddy
(Head of the Department ECE)

CMR INSTITUTE OF TECHNOLOGY

(UGCAUTONOMOUS)
Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC
Kandlakoya(V),Medchal Dist-501 401
www.cmrithyderabad.edu.in

2023-2024

CMR INSTITUTE OF TECHNOLOGY

(UGCAUTONOMOUS)

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUH, Hyderabad

Accredited by NBA and NAAC With A Grade

Kandlakoya (V), Medchal Dist-501401

www.cmrithyderabad.edu.in

CERTIFICATE

This is to certify that a Major Project entitled with "**IOT IN VEHICLE PRESENCE DETECTION OF SMART PARKING SYSTEM**" is being submitted by:

DYAWARASHETTY SAKETH	**20R01A04K5**
DULAM SRIJA	**20R01A04K6**
GOLLENA VINAY	**20R01A04K9**

To JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree of B.Tech in Electronics & Communication Engineering and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University forward of another degree or diploma.

Project Supervisor	Project Coordinator	HOD , ECE
(Dr. K. Niranjan Reddy)	**(Mrs.V.Sumathi)**	**(Dr. K. Niranjan Reddy)**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M Janga Reddy**, Director, **Dr. B. Satyanarayana**, Principal and **Dr. K. Niranjan Reddy**, Head of Department, Dept of Electronics & Communication Engineering, CMR Institute of Technology for their inspiration and valuable guidance during the entire duration.

We are extremely thankful to Project Coordinator **Mrs. V. Sumathi** Major Project Coordinator and internal guide **Dr. K. Niranjan Reddy**, Dept of Electronics & Communication Engineering, CMR Institute of Technology for their constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of their references and other literature referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly o in directly for every step towards success.

DYAWARASHETTY SAKETH	**20R01A04K5**
DULAM SRIJA	**20R01A04K6**
GOLLENA VINAY	**20R01A04K9**

Declaration

We **DYAWARASHETTY SAKETH (20R01A04K5), DULAM SRIJA (20R01A04K6), GOLLENA VINAY (20R01A04K9),** of the Major-Project entitled as "**IOT IN VEHICLE PRESENCE DETECTION OF SMART PARKING SYSTEM**" hereby declared that the matter embodied in this project is the genuine work done by us only and has not been submitted either to the university or to any university/institute for the fulfillment of the requirement of any course of study.

DYAWARASHETTY SAKETH	**20R01A04K5**
DULAM SRIJA	**20R01A04K6**
GOLLENA VINAY	**20R01A04K9**

ABSTRACT

More and more people cluster in major cities and thus causing urbanization challenges. In this scenario, many cities are employing technologies to lessen the impact on the environment and increase productivity and efficiency to cope with the high demand. Among which, smart transportation is essential in fulfilling the mobility in the urban areas and promising to meet the increasing demand of passengers. IoT application in a smart parking system has been facilitated in various ways. To achieve its full potential, we need to understand the associated issues and principles. Therefore, this paper focuses on understanding the principles within IoT, the process for the layers to function in vehicle detection, and introducing different IoT sensors empowered by different technologies, including cloud computing, big data, RFID, and WSN, that facilitate the smart parking system. After assessing the potential use cases of the IoT in the vehicle presence detection of the smart parking system, this paper will identify and discuss the benefits and challenges along with the recommendation and consideration that serve as the manual to help the industries and the governmental institutions to select appropriate sensors according to different scenarios.

INDEX

\

LIST OF FIGURES

LIST OF TABLES

1.INTRODUCTION:

1.1 OVERVIEW OF THE PROJECT :

The world population is distributed unevenly and highly concentrated in urban areas causing urbanization problems . Besides, there is only a limited amount of parking facilities provided for the citizens resulting in various parking problems such as illegal parking, traffic safety issues, and energy waste. Therefore, it encourages the government to develop a smart parking system so as to make good use of the existing parking utilities to meet the high demand of citizen needs. Along with the rapid development of the Internet of Things (IoT), cloud computing, and big data, it helps the traditional parking system become smarter and move towards the Mobility as a Service goal to achieve a sustainable transportation society. Smart car parking system is an integrated system to organize cars in public areas.All vehicles enter into the parking and waste time searching for parking slots.The smart parking system also allows drivers to use the space remotely via their smartphones, which is convenient if drivers prefer not to walk to find a space or avoid potential traffic congestion.

An IoT based smart parking system, also known as a connected parking system, is a centralized management system that allows drivers to use a smartphone app to search for and reserve a parking spot. The system's hardware features sensors that detect available parking slots and communicate this information to all drivers in the area. This data is updated in real-time, which means drivers never have to worry about not finding an available space. In addition to helping drivers find a spot, the system also sends alerts about peak times and peak prices. The goal of these alerts is to help save drivers money while also reducing congestion.

1.2 PROJECT OBJECTIVE:

- Improving car control and safety
- Monitoring the parking space in real-time
- Anticipating the traffic flow by examining parking patterns
- Making the most of time and space in a congested urban environment
- Enhanced security.

1.3 PROBLEM STATEMENT:

As the population has been increasing tremendously, vehicular traffic and its parking has become an issue of great concern. In public places where there are many visitors, a lot of time is wasted searching for parking slots. Also, a lot of manual labor is required to look after the existing parking arrangement. Moreover, there is no way of knowing whether a vacant parking space is available or not. The smart parking system is going to be implemented in several countries. In India, this type of parking system is most likely to be implemented in Bangalore in the near future.

In recent research in metropolitan cities the parking management problem can be viewed from several angles. High vehicle density on roads. This results in annoying issues for the drivers to park their vehicles as it is very difficult to find a parking slot. The drivers usually waste time and effort in finding parking space and end up parking their vehicles finding a space on streets. In the worst case, people fail to find any parking space especially during peak hours and festive seasons.

1. Parking management influences drivers' search time and cost for parking spaces.
2. It may also cause traffic congestion.
3. Finding a parking space in most metropolitan areas, especially during the rush hours, is difficult for drivers.
4. Difficulty arises from not knowing where the available spaces may be at that time traffic congestion may occur.

Urbanization is on the rise, and the complexity of city traffic is growing on a daily basis, especially post-COVID when everyone seems to be out on the road. Public transport is not easily accessible for all while parking a personal vehicle is even tougher (and costlier). The requirement for an Internet of Things (IoT)-powered parking system has, therefore, gained prominence to minimize the ongoing traffic congestion and to reduce the unpredictability of parking availability. With numerous connected devices on the horizon, devising an IoT-based smart parking system is relatively easy now.

2.SYSTEM PROPOSAL:

2.1 EXISTING SYSTEM:

Implementing IoT (Internet of Things) in the vehicle presence detection of a smart parking system can enhance efficiency, reduce congestion, and improve the overall user experience. Here's how you can integrate IoT into an existing smart parking system .Deploy IoT-enabled sensors in parking spaces: Use sensors such as ultrasonic sensors, infrared sensors, or cameras to detect the presence of vehicles in each parking space. Connect sensors to a microcontroller or IoT gateway: The sensors should be connected to a device that can collect and transmit data. This could be a microcontroller or an IoT gateway. Choose a communication protocol: Select a communication protocol (e.g., MQTT, CoAP, HTTP) to enable the sensors to transmit data to a central server or cloud platform. Ensure secure communication: Implement security measures, such as encryption and authentication, to secure the communication between sensors and the central system. Set up a cloud platform: Use a cloud service (e.g., AWS, Azure, Google Cloud) to collect and process data from the sensors. Develop an IoT backend: Create an IoT backend that can receive, store, and analyze data from the parking sensors. This backend should also provide APIs for interaction with other components. Implement data processing algorithms: Analyze the sensor data to determine the status of each parking space (occupied or vacant). Use machine learning for predictive analysis: Implement machine learning algorithms to predict parking space availability based on historical data. Develop a user interface: Create a user-friendly interface for both administrators and end-users to check real-time parking space availability. Mobile app integration: Develop a mobile application that users can use to find and reserve parking spaces, receive notifications, and make payments. Integrate with payment systems: Allow users to make payments for parking through the mobile app or other payment gateways. Implement automated payment processing: Use IoT data to automate the payment process based on the duration of parking .Provide real-time alert Implement a system that sends alerts to users when a parking space is about to become available or when their parking time is about to expire. Allow users to provide feedback on the parking experience, which can be used to improve the system. Ensure scalability: Design the system to handle a growing number of sensors and users as the smart parking system expands. Regular maintenance and updates: Schedule regular maintenance to ensure sensors are functioning correctly and update the system to address any security vulnerabilities or performance issues.

By incorporating these elements, you can enhance the efficiency and functionality of a smart parking system using IoT technology.

2.1.1 TRADITIONAL PARKING SYSTEM:

In recent years, number of car owners are increasing day by day, when someone tries to find a position to stop his car after a long day of work surprised by the lack of parking to stop his car and see the driver passes positions more than once to find him a position. Traditional parking consists of only two passages to enter and exit port. In some parking only one entrance to enter and out of space, while parking spaces are small for a small car, while the big vehicle takes more than one position, which makes things worse, sometimes the car stands but when it is not possible to open the door (Zhou & Li, 2016). The car is scratched with the next car because the parking is too small and cannot take up enough space between cars for the parking, but when you exit take a lot of time to get out of the car for fear of bumping any car near-by.

2.1.2 SMART PARKING SYSTEM:

Traditional parking has been developed to a parking system that helps the driver to know the occupied and available positions through a display that contains the number of available parking spaces and where they are located (Pham, Tsai, Nguyen, Dow, & Deng, 2015). Vacancy in a car cannot take this position, and in the case of the car out of the position goes light green and gives an update of the screen the presence of a vacant position can any car can stand in that position (Fraifer & Fernström, 2017), this system is used in many places and solve the problem of random parking and not to stand in places not available to stand up (Pham et al., 2015). Smart parking consists of two networks, an external network and an internal network, the external network is that the user connects to the Internet and enter the application of smart parking to reserve a position and be booking anywhere available in a data network, and the user enters the server of smart parking to be able to reserve the desired position without the need to access parking Through the application the user can know any vacant positions and available and places parking , The internal network of parking is the process of connecting all devices smart parking When the server is sending a signal to the cloud and the role of the cloud send this signal to the display screen and from the display screen to the sensor and the sensor to the top of the position and vice versa and this communication is done internally without the intervention of any employee or user.

Table 2.1 Comparison of Traditional and smart parking systems.

	Traditional parking system	Smart parking system
Planning	Without planning	With planning
Right parking	Parking any where	Parking in right area
Using IOT	NO	YES
Secure	No	Yes
Comfortable	No	Yes

2.2 PROPOSED SYSTEM:

The integration of IoT (Internet of Things) technology in vehicle presence detection for a Smart Parking System offers several advantages, improving efficiency, reducing traffic congestion, and enhancing overall parking management. Install IoT-enabled sensors in each parking space to detect the presence of a vehicle. Utilize various sensor technologies such as ultrasonic, infrared, or magnetic sensors. Sensors should be capable of wirelessly communicating with a central system.Use cloud-based solutions for scalability, data storage, and accessibility. Incorporate a database to store real-time and historical parking occupancy data. Develop a user-friendly mobile app for drivers to check parking availability in real-time. Enable features like navigation to available parking spots and mobile payments for parking fees. Provide a web-based interface for parking administrators to monitor and manage the entire parking system. Include features for analytics, reporting, and system configuration. Implement data analytics to derive insights from parking occupancy patterns. Use machine learning algorithms to predict parking space availability based on historical data, events, and trends. Collaborate with navigation system providers to integrate real-time parking data into their platforms. Enhance driver experience by providing seamless navigation to available parking spaces. Integrate a secure and automated payment system for parking fees. Utilize mobile payment options to enhance user convenience. Implement security protocols to protect the system from unauthorized access and data breaches. Ensure the confidentiality and integrity of data transmitted between sensors and the central server. Design the system to be scalable, allowing for the addition of more sensors and parking spaces. Plan for future upgrades and technology advancements in IoT and smart parking solutions. Efficient utilization of parking spaces, reducing congestion and optimizing available resources. Drivers can easily find available parking spaces, reducing search time and frustration. Parking administrators can make informed decisions based on data analytics and insights. Reduced fuel consumption and emissions due to decreased time spent searching for parking. Implementing an IoT-based vehicle presence detection system in smart parking not only addresses current parking challenges but also contributes to creating more sustainable and efficient urban environments.

The car parking system using IoT takes user authorization mechanism through a mobile app or license plate scanning.At the same time,the controller on the barrier or gate may allow or refuse drivers to park their cars according to the set parameters. To find the parking space from anywhere by using the mobile application.

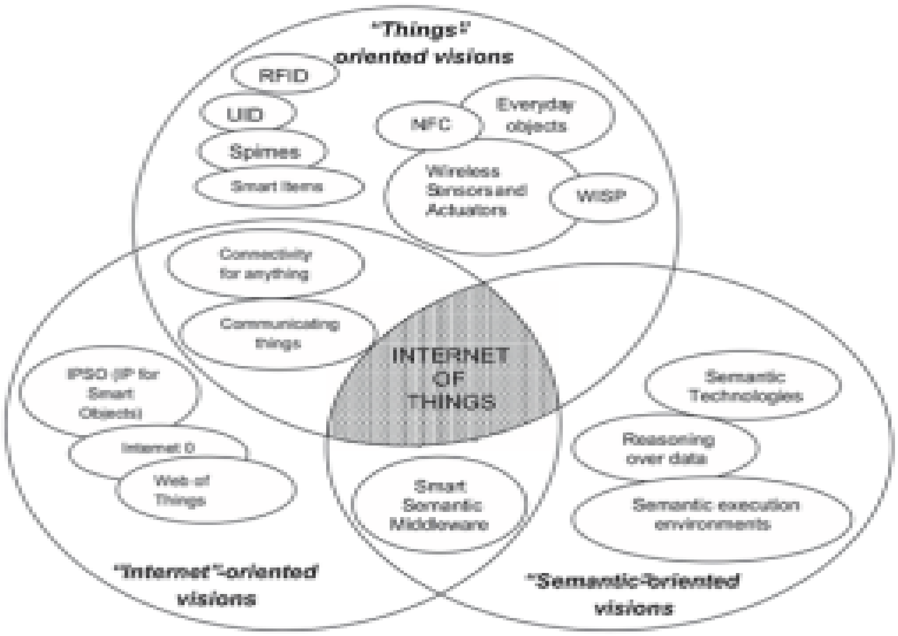
Fig.No. 2.2 : Smart parking management.

3. INTERNET OF THINGS:

3.1 IoT CONCEPT:

IoT is viewed as a global network to connect all items with the Internet through information sensing devices such as radio frequency identification (RFID) to realize intelligent identification and management. People may communicate and interact with each other and gain access to information from the machines through the Internet. IoT is identified in three visions (Fig.1.), which are things, IoT Concept IoT is viewed as a global network to connect all items with the Internet through information sensing devices such as radio frequency identification (RFID) to realize intelligent identification and management. People may communicate and interact with each other and gain access to information from the machines through the Internet. IoT is identified in three visions (Fig.1.), which are things,

Fig.No.3.1.1: Different layers of IoT.



IoT architecture can be divided into three layers (Fig.2.): the perception layer, the network layer, and the application layer. Perception layer is mainly used to study various kinds of variables and collect data from the environment through sensors, RFID, or other actuators. The network layer transports the collected data and coordinates with the application layer.

The application layer ultimately carries out data processing and calculation to perform the function to the end-users.

IoT is often used as a tool to enhance customer value in providing real-time monitor and control by collaborating with business analytics kits and capturing a great amount of personal data to trigger real-time services [10]. Wireless sensor networks (WSN) facilitate vehicle-to-infrastructure communication and the real-time locating systems in the smart parking to enable tracing and tracking systems.

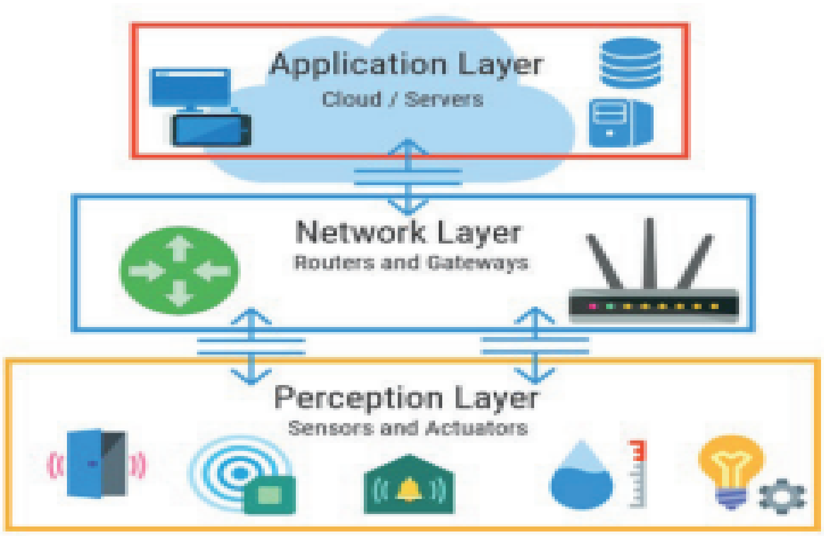


Fig.No.3.1.2: The three layers in IoT Architecture.

3.2 IoT IN SMART PARKING SYSTEM:

A. IoT in Smart Parking System specialized in Vehicle Presence Detection IoT smart parking system is composed of three layers: sense layer, communication layer, and application layer. (Fig.3.2.1) The data center in the application layer uses a cloud-based device to collect, compute, and store the information from different car parking locations [11]. At the same layer, the IoT management team facilitates the process via the IoT integrated portal. The integrated information will then be sent to different car parking information centers.

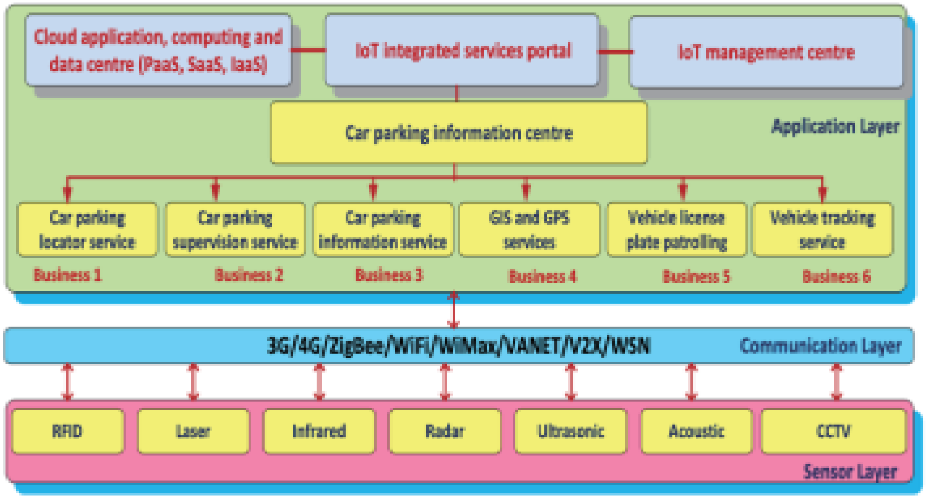


Fig.No. 3.2.1: IoT smart parking system architecture

Within the car parking information center, there are numerous business services connecting to the communication layer. In the communication layer, many wireless networks including WSN, Wi-Fi, and ZigBee are built to translate the information and connect with the sensor layer in real-time. In the sensor layer, different sensing technologies provide different services. RFID is most used for the vehicle parking lot gateway control and payment system, CCTV with image processing function utilizes the vehicle license number recognition, and laser, infrared, radar, acoustic and ultrasonic are used for presence detection [6]. Looking specifically at the presence detection, there are three presence detection sensors, which are radar sensor, wireless ultrasonic sensor, and wireless magnetometer sensor [5]. The radar sensor uses frequency modulated continuous wave radar to detect the vehicles. It can detect both the parked vehicles and the moving cars. Besides, a radar sensor can work in extreme weather conditions. Therefore, a radar sensor is suitable for long-distance outdoor detection. Since European countries are adopting electricity shared cars to cut down emission pollution,

there is a need for controlling recharging stations around the parking spaces. To maximize the usage, the radar sensor is installed to prevent an unpermitted car from parking at these spaces. When one car parked in the spaces, the radar sensor may detect whether the car has plugged with the charger. If it is not plugged in with the charger, the sensor will automatically send an alert to the information center to trigger authorities to conduct further actions. Besides, it can detect the blank spaces and notify the users of the nearest recharge station. However, if the two vehicles are too close to each other, it may cause false detection by recognizing two objects as one vehicle and leads to wrong information collection [1]. The wireless ultrasonic sensor is powered by the wireless network to collect the information through high-frequency ultrasound waves [8]. Ultrasound waves have several characteristics that make it ideal for an electronic application. Firstly, human users may not sense this ultrasound. Thus, it may avoid discomfort. Besides, it has higher directivity and lowers propagation speed than radio waves, allowing low-speed signal processing measurement at a low cost. Many large-scale parking lots are installed with such sensors. As for the wireless magnetometer sensor, it detects the object by sensing the change in the magnetic field. When a vehicle enters the parking lots, it will disrupt the original magnetic field making the sensor to recognize the changes. Therefore, a wireless magnetometer sensor requires no external control box or wiring environment. In that matter, it may cut down the cost. Besides, a wireless magnetometer sensor is small and self-contained, making it easier to maintain and commission quicker compared with the inductive sensors powered by electricity [7]. The installation of this kind of sensor only requires a three-inch diameter drill hole, indicating that it only takes a short time and leads to low cost. However, since this sensor measures the degree of disruption in the magnetic field, it is important to settle the typical ambient magnetic conditions during the installation stage. If the wireless magnetometer sensor is not aligned to the correct condition, it may cease the function. Therefore, it requires high accuracy during the installed learning process.

B. Comparison of different sensors Among the three different vehicle presence detection sensors, there are three key differences that affect the decision (Table 3.2) .

	Max Sensing Range	Size of Target	Mounting
Radar Sensor	Forty meters	Large, predictable targets	Minimum of six feet from target
Wireless Ultrasonic Sensor	Forty meters	All sizes	Must be installed overhead
Wireless Magnetometer Sensor	Depends on the size of target	All sizes	Can be installed above or below the vehicle

Table 3.2. Comparison of different sensors.

Radar sensors may sense up to forty meters wide field so it is ideal for detecting large vehicles such as buses, trucks, or cruises. The mounting configuration for a radar sensor requires to be a minimum of six feet away from the targeted vehicle. It can withstand the weather disruption so it provides higher detection accuracy. Besides, while inductive and capacitive sensors can only recognize moving objects, the radar sensor can detect both moving and stationary vehicles. A wireless ultrasonic sensor is restricted to install overhead and uses the sound wave to detect the vehicles. Therefore, this sensor can sense any vehicles regardless of the size, shape, or parking angle. A wireless magnetometer sensor measures the change of the magnetic field to detect the vehicle thus the sensing range depends on the size of the target object. Therefore, the wireless magnetometer sensor is often used for detecting large-volume vehicles such as trucks, trains, or buses. Besides, since this sensor detects objects by wireless magnetic technology, it can be installed both on top of the object or below the ground.

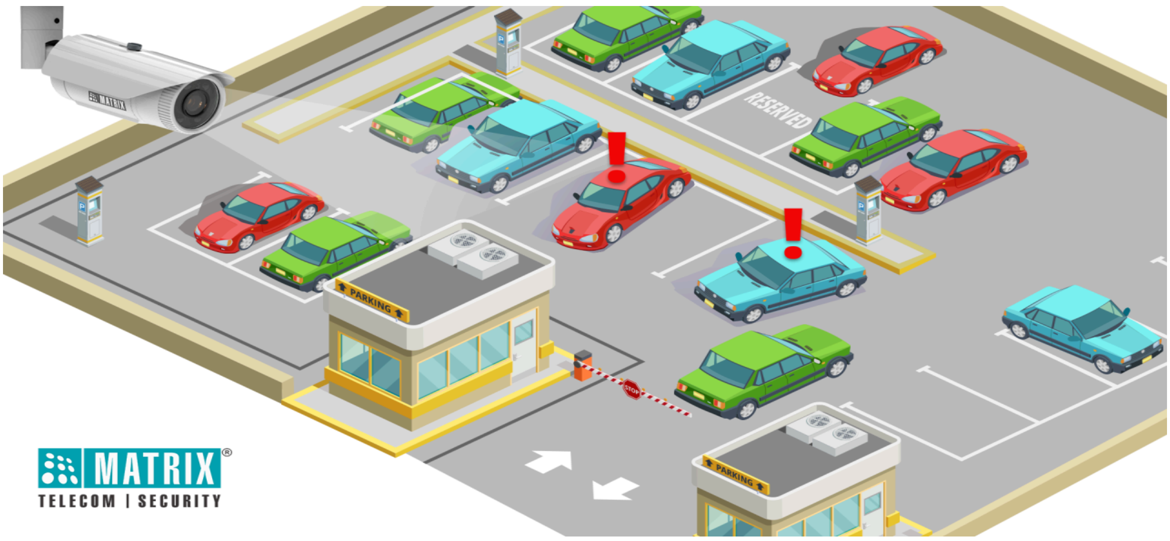


Fig.No.3.2.3: Observation CCTV Camera.

4.LITERATURE SURVEY

Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. Computer Networks, 51(4), 921-960. doi:10.1016/j.comnet.2006.10.002

The availability of low-cost hardware such as CMOS cameras and microphones has fostered the development of Wireless Multimedia Sensor Networks (WMSNs), i.e., networks of wirelessly interconnected devices that are able to ubiquitously retrieve multimedia content such as video and audio streams, still images, and scalar sensor data from the environment. In this paper, the state of the art in algorithms, protocols, and hardware for wireless multimedia sensor networks is surveyed, and open research issues are discussed in detail. Architectures for WMSNs are explored, along with their advantages and drawbacks. Currently off-the-shelf hardware as well as available research prototypes for WMSNs are listed and classified. Existing solutions and open research issues at the application, transport, network, link, and physical layers of the communication protocol stack are investigated, along with possible cross-layer synergies and optimizations. Wireless sensor networks (WSN) [22] have drawn the attention of the research community in the last few years, driven by a wealth of theoretical and practical challenges. This growing interest can be largely attributed to new applications enabled by large-scale networks of small devices capable of hard vesting information from the physical environment, performing simple processing on the extracted data and transmitting it to remote locations. Significant results in this area over the last few years have ushered in a surge of civil and military applications. As of today, most deployed wireless sensor networks measure scalar physical phenomena like temperature, pressure, humidity, or location of objects. In general, most of the applications have low bandwidth demands, and are usually delay tolerant. More recently, the availability of inexpensive hardware such as CMOS cameras and microphones that are able to ubiquitously capture multimedia content from the environment has fostered the development of Wireless Multimedia Sensor Networks (WMSNs) [54,90], i.e., networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images, and scalar sensor data. With rapid improvements and miniaturization in hardware, a single sensor device can be equipped with audio and visual information collection modules. As an example, the Cyclops image capturing and inference module [103], is designed for extremely light-weight imaging and can be interfaced with a host mote such as Crossbow's MICA2 [4] or MICAz [5]. In addition to the ability to retrieve multimedia data, WMSNs will also be able to store, process in real-time, correlate and fuse multimedia data originated from heterogeneous sources. Wireless multimedia sensor networks will not only enhance existing sensor network applications such as tracking, home automation, and environmental monitoring, but they will also enable several new applications such as: • Multimedia surveillance sensor networks. Wireless video sensor networks will be composed of interconnected, battery-powered miniature video cameras, each packaged with a low-power wireless transceiver that is capable of processing, sending, and receiving data. Video and audio sensors will be used to enhance and complement existing surveillance systems against crime and terrorist attacks. Large-scale networks of video sensors can extend the ability of law enforcement agencies to monitor areas, public events, private properties and borders. • Storage of potentially relevant activities. Multimedia sensors could infer and record potentially relevant activities (thefts, car accidents, traffic violations), and make video/audio streams or reports available for future query. • Traffic avoidance, enforcement and

control systems. It will be possible to monitor car traffic in big cities or highways and deploy services that offer traffic routing advice to avoid congestion. In addition, smart parking advice systems based on WMSNs [29] will allow monitoring available parking spaces and provide drivers with automated parking advice, thus improving mobility in urban areas. Moreover, multimedia sensors may monitor the flow of vehicular traffic on highways and retrieve aggregate information such as average speed and number of cars. Sensors could also detect violations and transmit video streams to law enforcement agencies to identify the violator, or buffer images and streams in case of accidents for subsequent accident scene analysis. • Advanced health care delivery. Telemedicine sensor networks [59] can be integrated with 3G multimedia networks to provide ubiquitous health care services. Patients will carry medical sensors to monitor parameters such as body temperature, blood pressure, pulse oximetry, ECG, breathing activity. Furthermore, remote medical centers will perform advanced remote monitoring of their patients via video and audio sensors, location sensors, motion or activity sensors, which can also be embedded in wrist devices [59]. • Automated assistance for the elderly and family monitors. Multimedia sensor networks can be used to monitor and study the behavior of elderly people as a means to identify the causes of illnesses that affect them such as dementia [106]. Networks of wearable or video and audio sensors can infer emergency situations and immediately connect elderly patients with remote assistance services or with relatives. • Environmental monitoring. Several projects on habitat monitoring that use acoustic and video feeds are being envisaged, in which information has to be conveyed in a time-critical fashion. For example, arrays of video sensors are already used by oceanographers to determine the evolution of sandbars via image processing techniques [58]. • Person locator services. Multimedia content such as video streams and still images, along with advanced signal processing techniques, can be used to locate missing persons, or identify criminals or terrorists. • Industrial process control. Multimedia content such as imaging, temperature, or pressure amongst others, may be used for time-critical industrial process control. Machine vision is the application of computer vision techniques to industry and manufacturing, where information can be extracted and analyzed by WMSNs to support a manufacturing process such as those used in semiconductor chips, automobiles, food or pharmaceutical products. For example, in quality control of manufacturing processes, details or final products are automatically inspected to find defects. In addition, machine vision systems can detect the position and orientation of parts of the product to be picked up by a robotic arm. The integration of machine vision systems with WMSNs can simplify and add flexibility to systems for visual inspections and automated actions that require high-speed, high-magnification, and continuous operation.

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805. doi:10.1016/j.comnet.2010.05.010.

This paper addresses the Internet of Things. Main enabling factor of this promising paradigm is the integration of several technologies and communications solutions. Identification and tracking technologies, wired and wireless sensor and actuator networks, enhanced communication protocols (shared with the Next Generation Internet), and distributed intelligence for smart objects are just the most relevant. As one can easily imagine, any serious contribution to the advance of the Internet of Things must necessarily be the result of synergetic activities conducted in different fields of knowledge, such as telecommunications, informatics, electronics and social science.

In such a complex scenario, this survey is directed to those who want to approach this complex discipline and contribute to its development. Different visions of this Internet of Things paradigm are reported and enabling technologies reviewed. What emerges is that still major issues shall be faced by the research community. The most relevant among them are addressed in details. The Internet of Things (IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects - such as Radio Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. - which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [1]. Unquestionably, the main strength of the IoT idea is the high impact it will have on several aspects of everyday-life and behavior of potential users. From the point of view of a private user, the most obvious effects of the IoT introduction will be visible in both working and domestic fields. In this context, domotics, assisted living, e-health, enhanced learning are only a few examples of possible application scenarios in which the new paradigm will play a leading role in the near future. Similarly, from the perspective of business users, the most apparent consequences will be equally visible in fields such as, automation and industrial manufacturing, logistics, business/process management, intelligent transportation of people and goods. By starting from the considerations above, it should not be surprising that IoT is included by the U.S. National Intelligence Council in the list of six “Disruptive Civil Technologies” with potential impacts on U.S. national power [2]. NIC foresees that “by 2025 Internet nodes may reside in everyday things—food packages, furniture, paper documents, and more”. It highlights future opportunities that will arise, starting from the idea that “popular demand combined with technology advances could drive widespread diffusion of an Internet of Things (IoT) that could, like the present Internet, contribute invaluably to economic development”. The possible threats deriving from a widespread adoption of such a technology are also stressed. Indeed, it is emphasized that “to the extent that everyday objects become information security risks, the IoT could distribute those risks far more widely than the Internet has to date”. This is a pre-printed version of the paper that has been published in the Computer Networks Journal, Volume 54, Issue 15, 28 October 2010, Pages 2787-2805, Elsevier. The accepted and published version is available from the Journal Website. DOI: https://doi.org/10.1016/j.comnet.2010.05.010 Actually, many challenging issues still need to be addressed and both technological as well as social knots have to be untied before the IoT idea being widely accepted. Central issues are making a full interoperability of interconnected devices possible, providing them with an always higher degree of smartness by enabling their adaptation and autonomous behavior, while guaranteeing trust, privacy, and security. Also, the IoT idea poses several new problems concerning the networking aspects. In fact, the things composing the IoT will be characterized by low resources in terms of both computation and energy capacity. Accordingly, the proposed solutions need to pay special attention to resource efficiency besides the obvious scalability problems. Several industrial, standardization and research bodies are currently involved in the activity of development of solutions to fulfill the highlighted technological requirements. This survey gives a picture of the current state of the art on the IoT. More specifically, it: • provides the readers with a description of the different visions of the Internet of Things paradigm coming from different scientific communities. • reviews the enabling technologies and illustrates which are the major benefits of spread of this paradigm in everyday life. • offers an analysis of the major research issues the scientific community still has to face. The main objective is to give the reader the opportunity of understanding what has been done (protocols, algorithms, proposed solutions) and what still remains to be addressed, as well as which are the enabling factors of this evolutionary process and what are its weaknesses and risk factors.

Guilbert, D., Le Bastard, C., Sio-Song, I., & Yide, W. (2016). State Machine for Detecting Vehicles by Magnetometer Sensors. IEEE Sensors Journal, 16(13), 5127-5128. doi:10.1109/JSEN.2016.2560903.

This letter presents a vehicle detection state machine for magnetometer sensors. The state machine designed by Chinrungrueng et al. [1] for a single lane is extended to n motorway lanes with multiple lane changes. The proposed method has good performance in terms of the detection rate and false detections which are mainly due to the interference from a vehicle in an adjacent lane and lane changing. The state machine is evaluated on a real motorway and compared to industrial sensors. The understanding of road traffic congestion phenomena, the cause of accidents and the environmental aspect due to pollutants requires observations of road traffic. Many projects in Europe such as the MOCoPo project [2] are installing observation areas with different types of sensors, among which, the magnetometer is a promising new road sensor. In the MOCoPo project [2], the magnetometers are used to detect vehicles on the entry and exit ramp of a French motorway (Fig. 1), a short (about 300m) and very congested section. Wireless 3D magnetometers are specially designed to detect and reidentify vehicles in a congested situation. The sensors are sited with the commercial Sensys Network magnetometer at the entrance (Fig. 2b) or the classical Inductive Loop Detector (ILD) at the exit (Fig. 2a) for comparison purposes. In a congested situation, the detection of each vehicle is vital and the algorithm must properly segment the signal. Before undertaking the re-identification from the magnetometer, we evaluate the vehicle detection from different sensors. An experimental session with video recording takes place. The videos have been manually analyzed to give a reference database containing all vehicle which transits at the entrance/exit for each lane. The detection rate for the wireless magnetometer specially designed for the MOCoPo project [2] is considered insufficient to our purposes. In this paper, we propose a new detection algorithm based on the state machine proposed by Chinrungueng et al. [1]. In [1], the method was proposed for detecting vehicles in a single lane. Thus, it can not be directly applied to a multi-lane roadway. In this paper, we propose to extend this method to n motorway lanes with multiple lane changes. This complex situation can lead to false detections (FD) due to the interference from vehicles on adjacent lanes (AL) and lane changes (LC).One of the intelligent systems for car parking has been proposed by making use of Image processing [9]. In this system, a brown rounded image on the parking slot is captured and processed to detect the free parking slot. The information about the currently available parking slots is displayed on the 7-segment display.

Initially, the image of parking slots with brown rounded image is taken. The image is segmented to create binary images. The noise is removed from this image and the object boundaries are traced. The image detection module determines which objects are round, by estimating each object‟s area and perimeter. Accordingly, the free parking space is allocated. A vision-based car parking system [10] is developed which uses two types of images (positive and negative) to detect free parking slot. In this method, the object classifier detects the required object within the input. Positive images contain the images of cars from various angles. Negative images do not contain any cars in them. The coordinates of parking lots specified are used as input to detect the presence of cars in the region. Haar-like features are used for feature detection.Data within the SNs is usually collected from the GN in addition to delivered to Road Side Equipment (RSU) to blame for fusing the idea by means of traffic related data files produced just by free options. Their system has become tested using quite a few serious use-case circumstances. Their report comprises addition particulars by using TelosB. The work marks solving the useful traffic organization test by giving a good framework to get traffic data.

5. REQUIREMENTS SPECIFICATIONS:

SOFTWARE REQUIREMENTS:

1. **Arduino IDE**
2. **Python3**
3. **PyCharm**
4. **CV zone**
5. **Windows 7**
6. **ThingSpeak**

HARDWARE REQUIREMENTS:

1. **Monitor**
2. **Processor**
3. **CCTV**
4. **Arduino UNO**

5.1 SOFTWARE PRINCIPLES:

5.1.1 PYTHON:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms ,including structured (particularly, procedural), object- oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.
Python uses dynamic typing and a combination of reference counting and a cycle- detecting garbage collector for memory management. It also features. dynamic name resolution (late binding), which binds method and variable names during program execution. Python's design offers some support for functional programming in the Lisp tradition.It has filter, map and reduce functions; list comprehensions, dictionaries, sets, and generator expressions.The standard library has two modules (itertools and functools) that implement functional tools Borrowed from Haskell and Standard ML.

5.1.2 PyCharm:

PyCharm is an integrated development environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm is developed by the Czech company JetBrains. It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License.PyCharm Community Edition is less extensive than the Professional Edition.

History of PyCharm:

PyCharm was released to the market of the Python-focused IDEs to compete with PyDev (for Eclipse) or the more broadly focused Komodo IDE by ActiveState.

The beta version of the product was released in July 2010, with the 1.0 arriving 3 months later. Version 2.0 was released on 13 December 2011, version 3.0 was released on 24 September 2013, and version 4.0 was released on November 19, 2014.

PyCharm became Open Source on 22 October 2013. The Open Source variant is released under the name *Community Edition* – while the commercial variant, *Professional Edition*, contains closed-source modules.

Features:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python code refactoring: including rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line coverage
- Google App Engine Python development

- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with changelists and merge
- Scientific tools integration: integrates with IPython Notebook, has an interactive Python console, and supports Anaconda as well as multiple scientific packages including Matplotlib and NumPy.

5.1.3 CV Zone:

This is a Computer vision package that makes it easy to run Image processing and AI functions. At the core it uses OpenCV and Mediapipe libraries.OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly for real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage, then Itseez (which was later acquired by Intel). The library is cross-platform and licensed as free and open-source software under Apache License 2. Starting in 2011, OpenCV features GPU acceleration for real-time operations.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations.

History of CV Zone:

In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.
In July 2020, OpenCV announced and began a Kickstarter campaign for the OpenCV AI Kit, a series of hardware modules and additions to OpenCV supporting Spatial AI.
In August 2020, OpenCV launched OpenCV.ai – the professional consulting arm. The team of developers provides consulting services and delivers Computer Vision, Machine Learning, and Artificial intelligence solutions.

Applications:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object detection
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion video tracking

5.1.4 Arduino IDE:

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') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages (Embedded C), using a standard API which is also known as the Arduino Programming Language, inspired by the Processing language and used with a modified version of the Processing IDE. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute 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.

History of Arduino IDE:

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $50. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega128 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller.

In 2005, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, extended Wiring by adding support for the cheaper ATmega8 microcontroller. The new project, forked from Wiring, was called Arduino.

5.1.5 ThingSpeak:

ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

- 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
- 3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- 1 status field - A short message to describe the data stored in the channel.

To use ThingSpeak, we need to signup and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same. Let us start exploring ThingSpeak by signing up and setting up a channel.

ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices.It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.

ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks,allowing ThingSpeak users to analyze and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks.

ThingSpeak has been the subject of articles in specialized "Maker" websites like Instructables, Codeproject, and Channel 9.

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one. Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

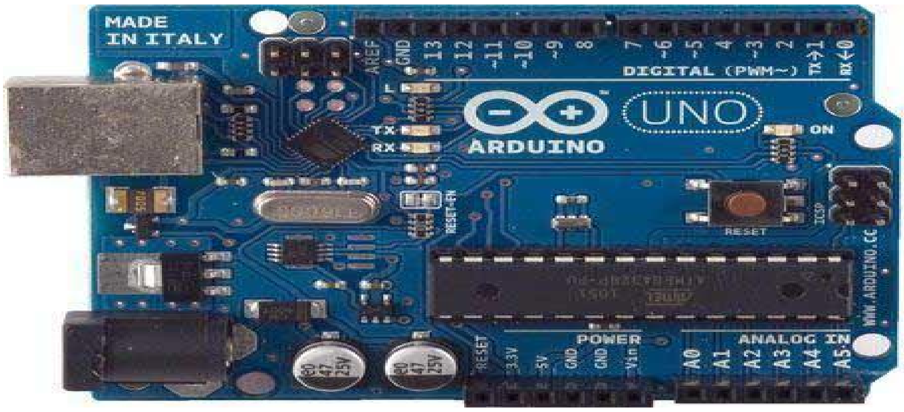
5.2 HARDWARE PRINCIPLES:

COMPONENTS:

5.2.1 Arduino UNO:

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

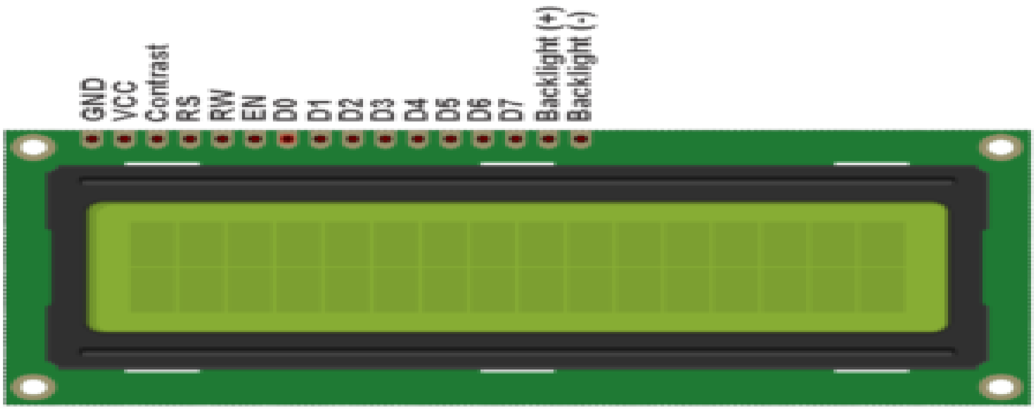
Fig.No.5.2.1: Arduino UNO



5.2.2 LCD Module:

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce images in color or monochrome.

Fig.No.5.2.2: LCD Module.



5.2.3 REGULATED POWER SUPPLY:

This is a device or system that supplies electrical or other types of energy to an output load or group of loads, also called as power supply unit or PSU.

This includes a power distribution system as well as primary or secondary sources of energy such as Conversion of one form of electrical power to another desired form and voltage, typically involving converting AC line voltage to a well-regulated lower-voltage DC for electronic devices. Low voltage, low power DC power supply units are commonly integrated with the devices they supply, such as computers and household electronics.

Fig.No.5.2.3: Regulated power supply.

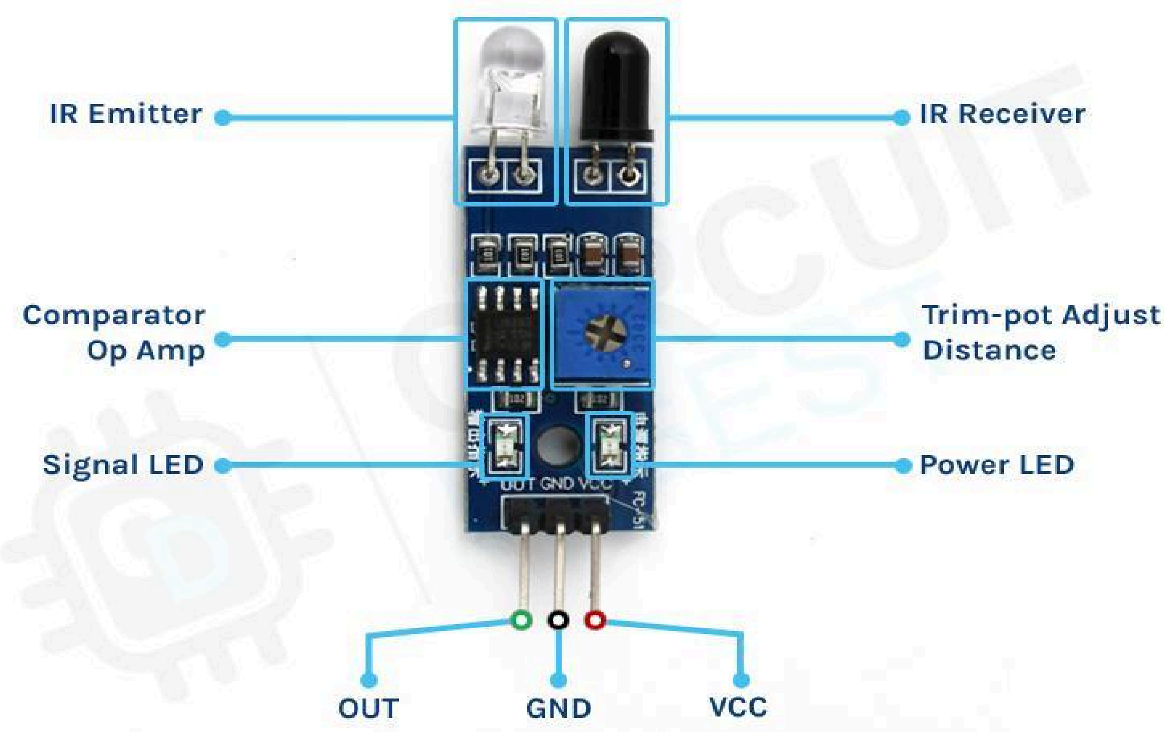
5.2.4 IR SENSOR:

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation.

When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analyzed as such, therefore this output can be fed to a comparator circuit. Here an operational amplifier (op-amp) of LM 339 is used as a comparator circuit.

Fig.No.5.2.4: IR Sensor



5.2.5 RFID READER:

Radio Frequency Identification (RFID) technology uses radio waves to identify people or objects. There is a device that reads information contained in a wireless device or “tag” from a distance without making any physical contact or requiring a line of sight.

Active RFID and Passive RFID technologies, while often considered and evaluated together, are fundamentally distinct technologies with substantially different capabilities.RFID Reader Module, are also called as interrogators. They convert radio waves Returned from the RFID tag into a form that can be passed on to Controllers, which can Make use of it. RFID tags and readers have to be tuned to the same frequency in order to Communicate. RFID systems use many different frequencies, but the most common and Widely used & supported by our Reader is 125 KHz.

Fig.No.5.2.5.1: RFID Reader

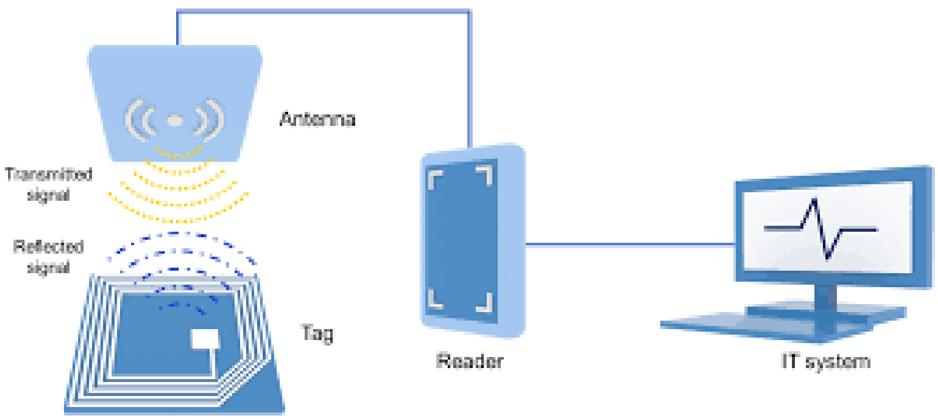


Fig.No.5.2.5.2: System Operation.

5.2.6 SERVO MOTOR:

Servo motor is a rotary actuator or motor that allows for a precise control in terms of angular position, acceleration and velocity, capabilities that a regular motor does not have special type of motor which is automatically operated up to certain limit for a given command with help of error-sending feedback to correct the performance. It is used to open the barrier gate when the user arrives at entry point and departure at exit point.

Fig.No.5.2.6: Servo Motor

5.2.7 LED :

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for lighting.

The structure of the LED light is completely different from that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's color. The LED is based on the semiconductor diode. When a diode is forward biased (switched on), electrons can recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light.

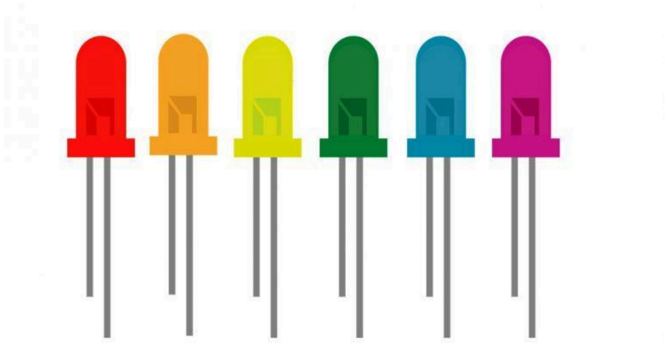


Fig.No.5.2.7: LED’s

5.2.8 Cables:

Cables Use it to connect Arduino Uno, Arduino Mega 2560, Arduino 101 or any board with the USB female A port of your computer. Cable length is approximately 100 cm. Cable color and shape may vary slightly from image as our stock rotates.

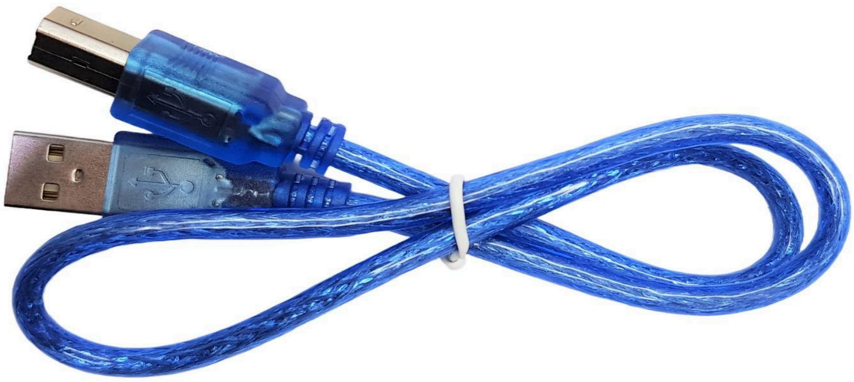


Fig.No.5.2.8: Cables.

6. SYSTEM DESIGN:

6.1 SYSTEM ARCHITECTURE:

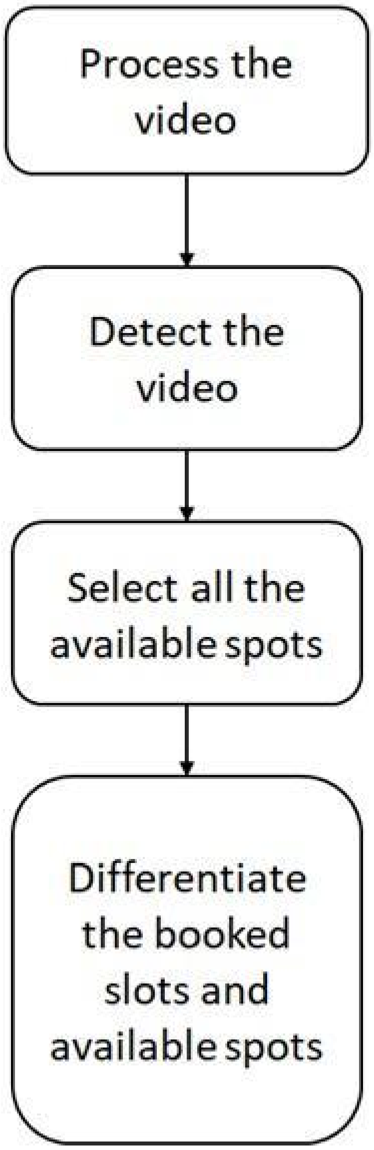


Fig.No.6.1: System Architecture

6.2 BLOCK DIAGRAM:

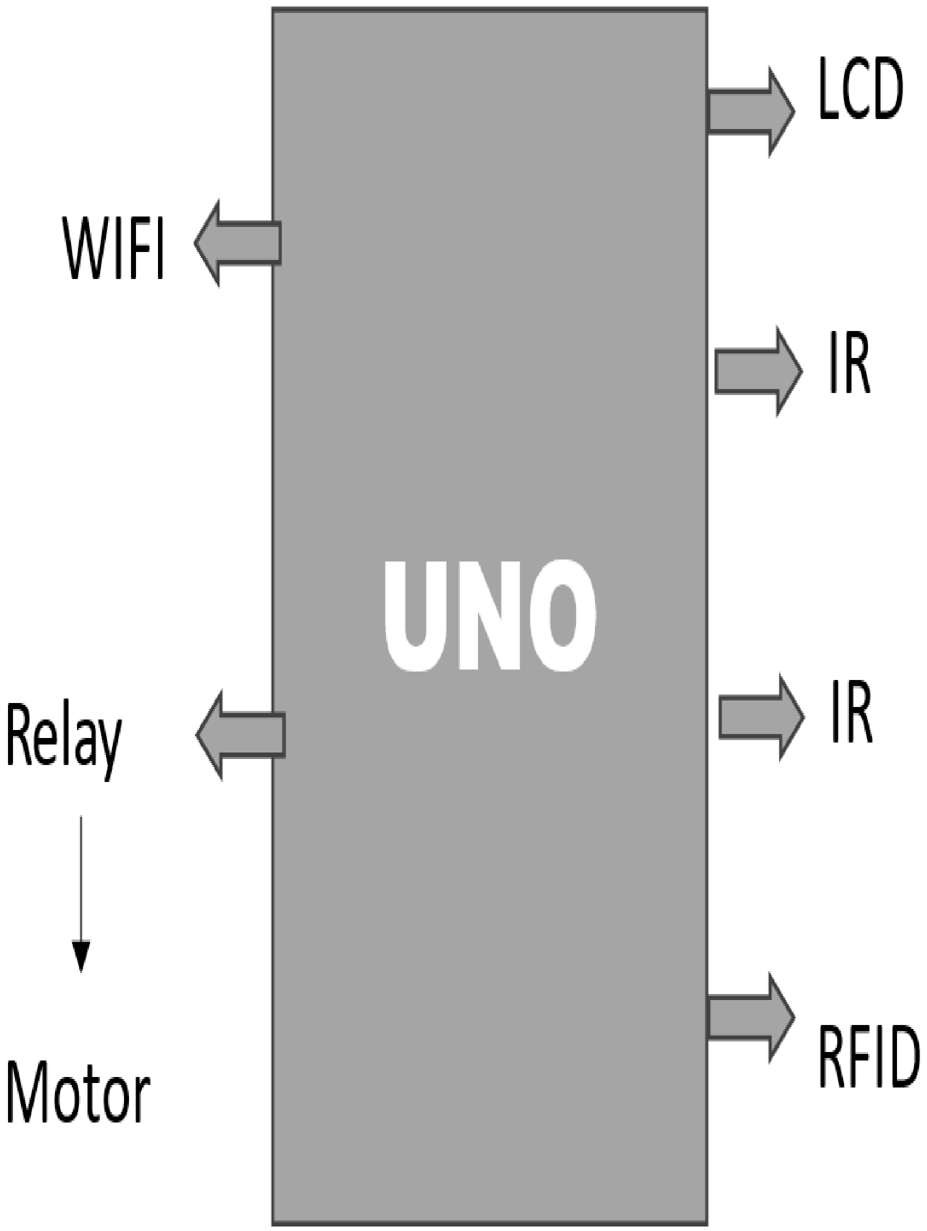


Fig.No.6.2: Block Diagram

6.3 UML DIAGRAMS:

Fig.No.6.3.1: Activity diagram

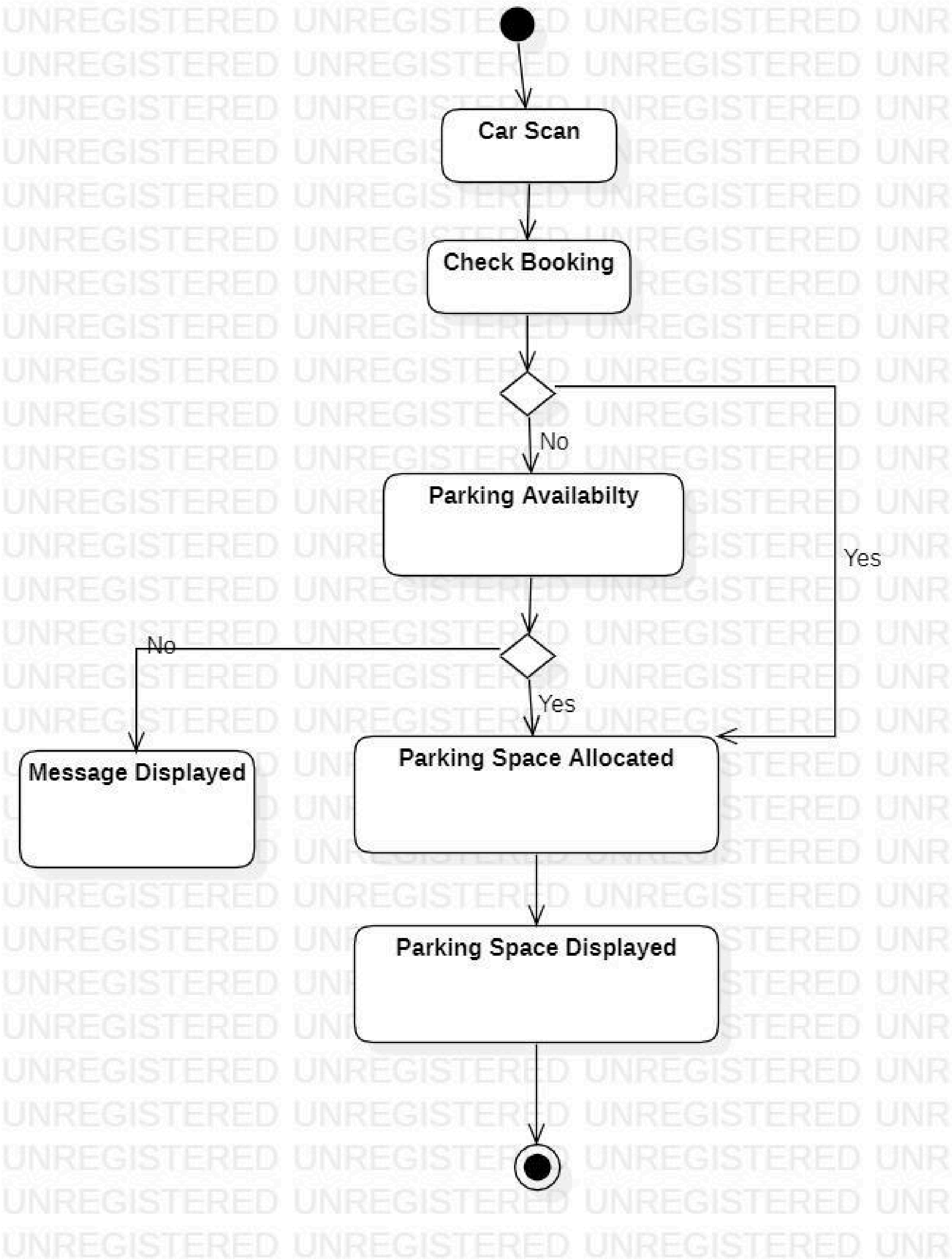


Fig.No. 6.3.2: Use case diagram

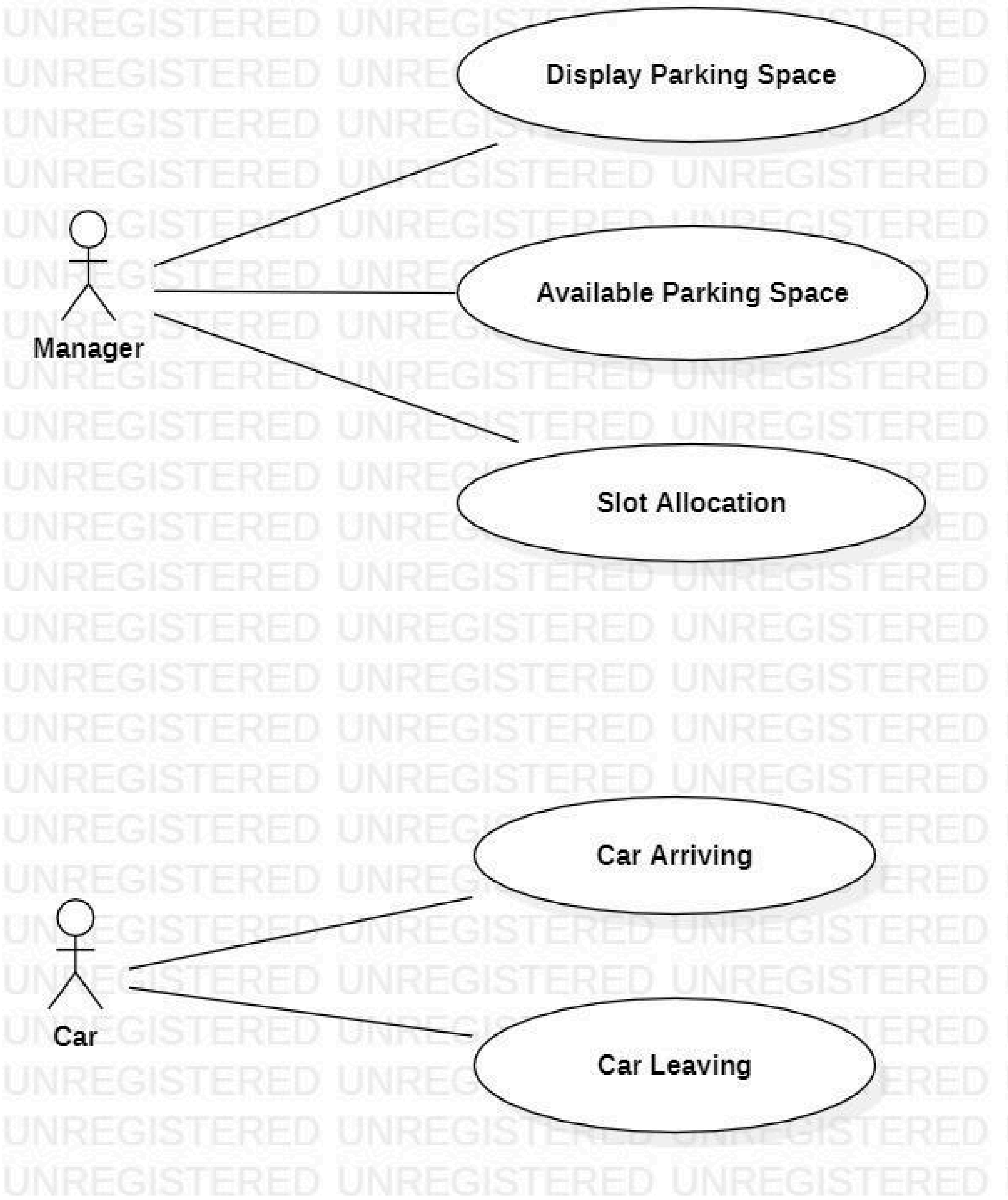
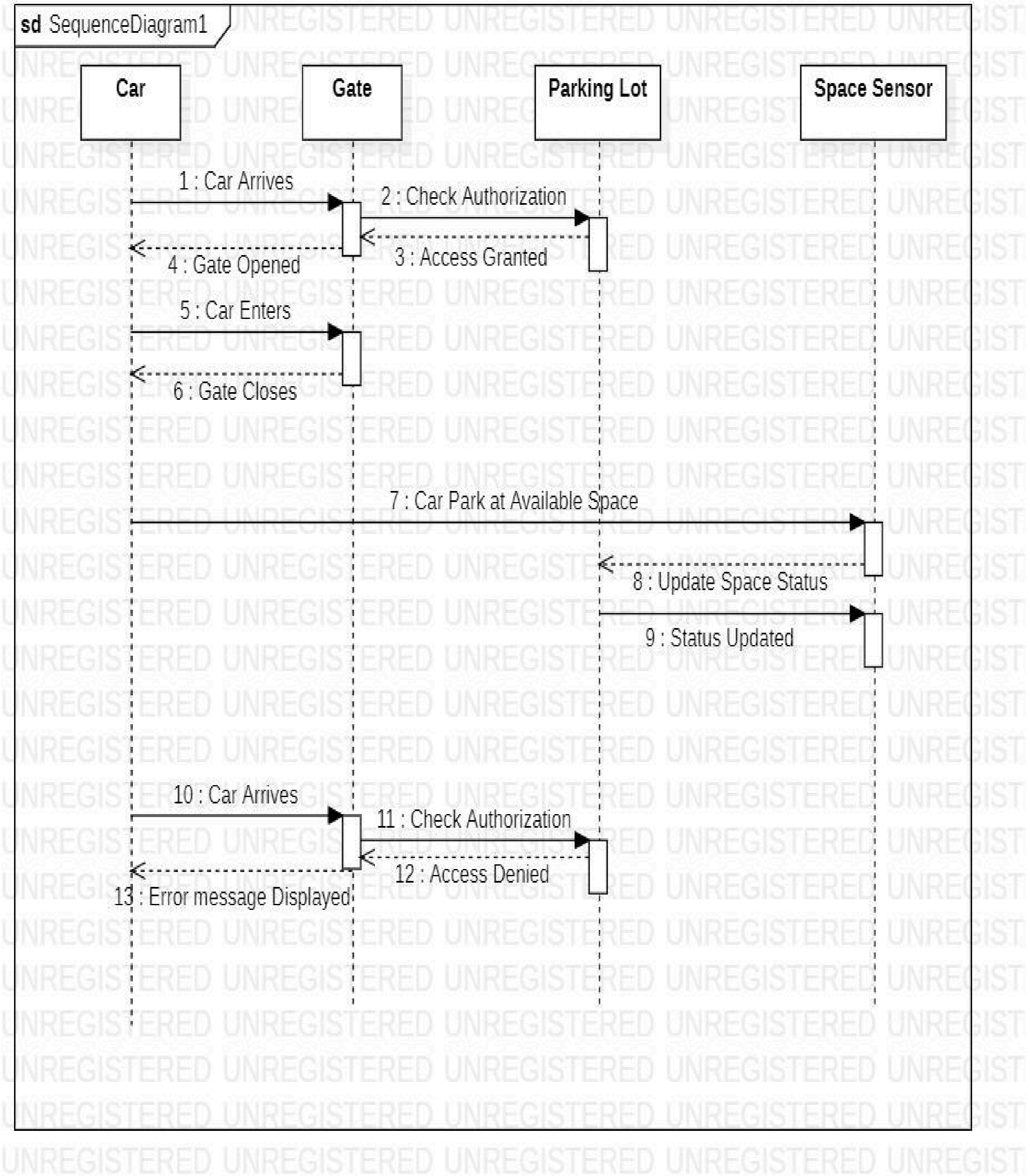


Fig.No.6.3.3: Sequence Diagram.



7.IMPLEMENTATION:

7.1 SETUP THINGSPEAK CLOUD:

We are using a (free) cloud service called Thingspeak where we will send parking lot's data to share with the public.

- First you need to sign up for Thingspeak
- Enter the credentials it asks for and create a new channel and do the following to your new channel:

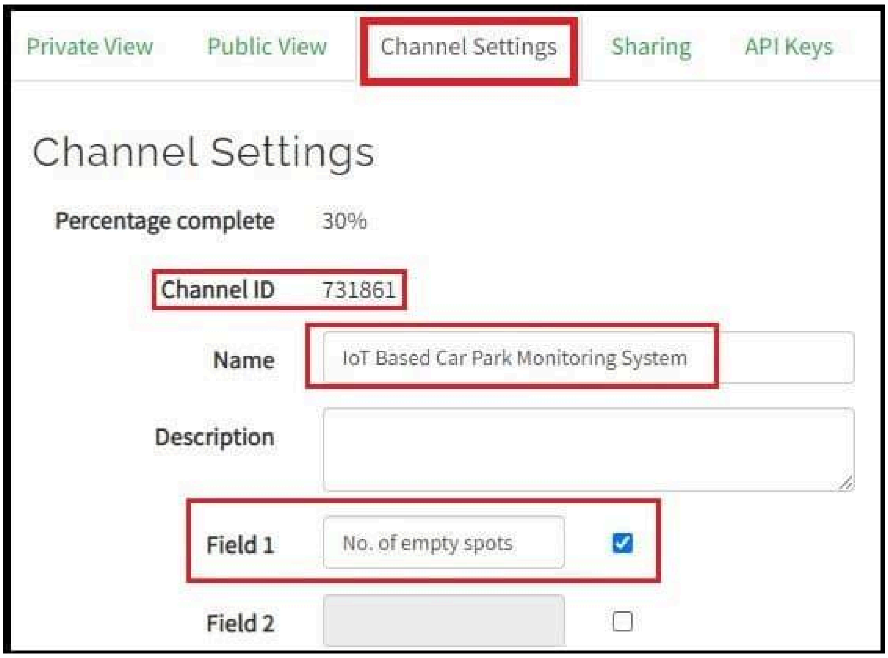


Fig No. 7.1.1 Channel settings

- Go to **channel settings** and enter the things as shown above and take note of your channel ID which we need to enter in the program code.
- Scroll down and **click save** to save the changes.
- Now click on the API **keys tab** and you will see your keys as illustrated below. API keys are responsible for writing and reading the data to your Thingspeak account.
- Go to **channel settings** and enter the things as shown above and take note of your channel ID which we need to enter in the program code.
- Scroll down and **click save** to save the changes.

Now click on the API **keys tab** and you will see your keys as illustrated below. API keys are responsible for writing and reading the data to your Thingspeak account.

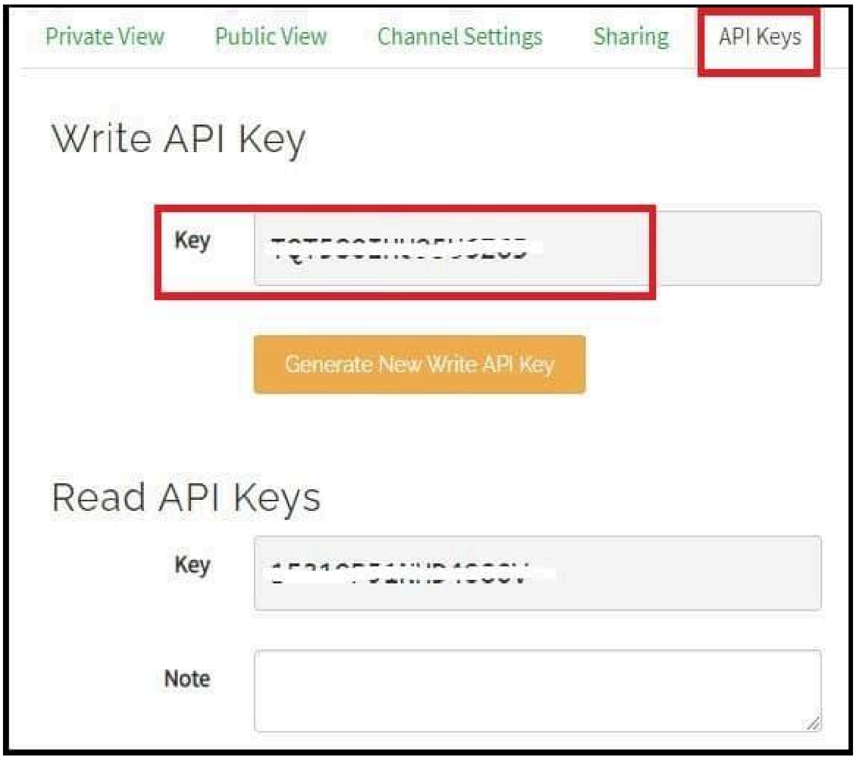


Fig No.7.1.2 API Key

- Take note of your "write API key" which needs to be entered in the program code and read API key is not used in this project.
- Now go to the sharing tab and click on "share channel view with everyone", this makes your channel visible to those who have the URL of the "public view" page.

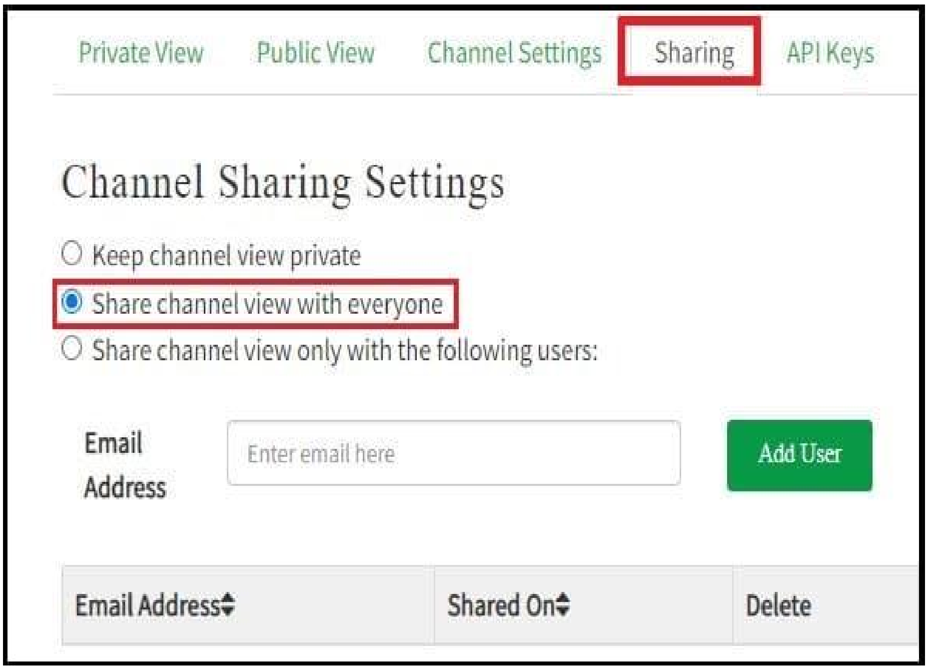


Fig No. 7.1.3 Channel Sharing

- Click on the public **view tab** and you will see an empty graph field and the **URL of this page can be shared to the public**.

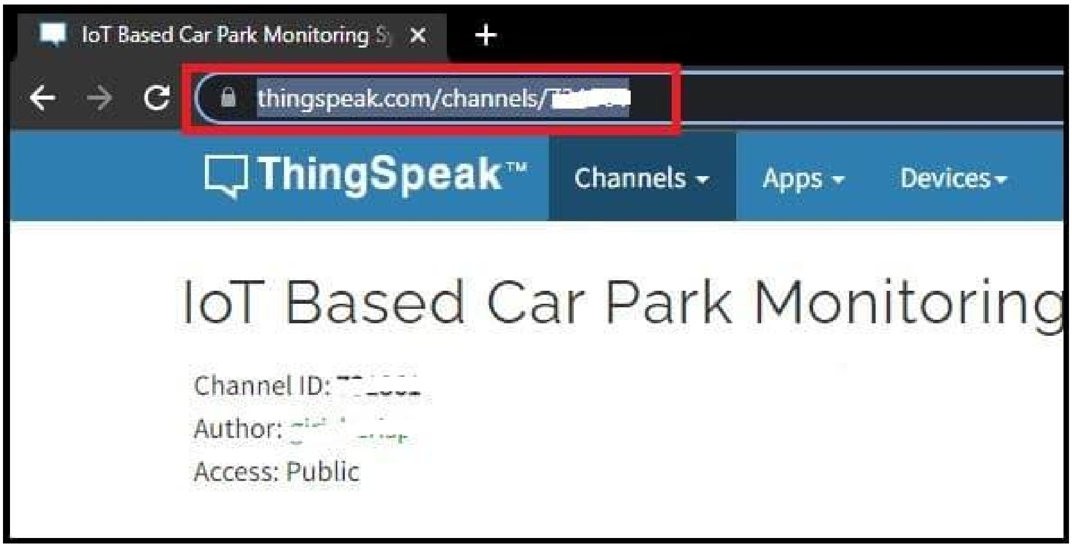


Fig No.7.1.4 Public URL

- Now on the public view tab we are going to setup a number widget where public can view the number of vacant spots on the parking lot, click on “add widget”.

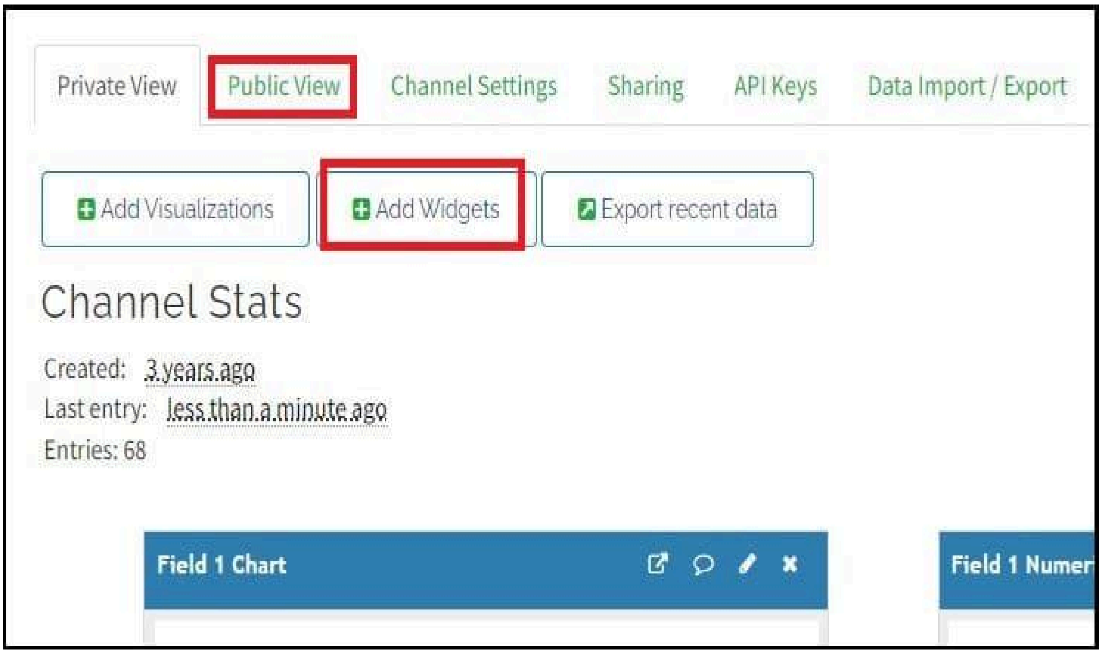


Fig No. 7.1.5 Public view settings

- A window will pop-up as illustrated below, choose numeric display and click next.

Fig No.7.1.6 Widget Settings

- Now fill the fields as shown below and click on create.

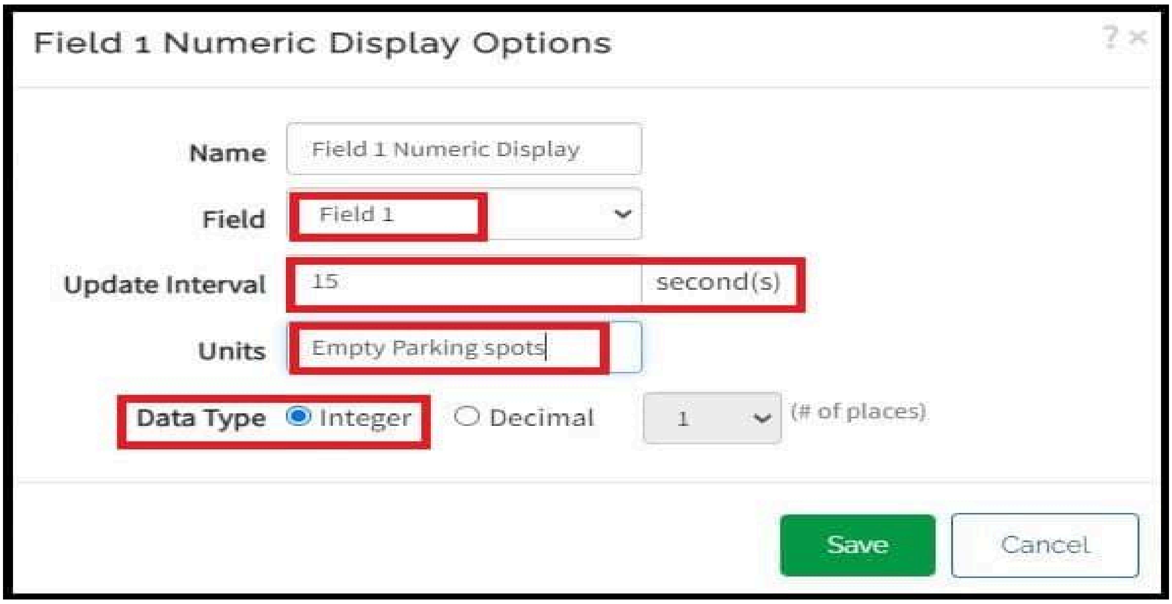


Fig No. 7.1.7 Display settings

- Now you will see a new widget where number will be displayed once we send data. Close the "field chart" by clicking on 'X', the public just need to know the number of vacant parking spots and not the parking history.

Fig No.7.1.8 Display and Charts

- Click on **private view tab**, you see an empty field chart; this is where you can view the history car parking history and this is not visible to public. This concludes on Thingspeak account setup.

7.2 MODULE DESCRIPTION:

This section contains the implementation of the proposed system. Every user who enters the parking slot contains a RFID card which contains the details of the user.When the RFID card is scanned by the reader module, the details of the user are transferred into the module. Now the IR sensor checks whether the parking space is free .If there is no space available the parking barrier gate will not open.A Message is sent to the user with the help of a GSM module which sends a registered message depending upon the availability and unavailability of the parking space. The WIFI module supports the system by storing all the data in the cloud.

It connects the devices with the cloud server. Here, the user scans the RFID card provided to the user. If space is available, the user receives a message “Welcome username” the barrier gate will open and the user can park the car. When the user exits the parking space the user again has to scan the RFID and a message will be received by the user “thanks for using smart parking username”. The database about the user’s activity in the parking space will be stored in a cloud database. The user will know that a particular space is available with the help of the cloud status. When the car is parked the IR sensor detects the presence of an object and updates the cloud status from 0 to 1 and when the car leaves, the cloud status is updated from 1 to 0. So, the user can park his car where the cloud status is 0. The cloud status is updated every 2 minutes.

Smart parking is an emerging market with immense potential. According to a recent study, the global smart parking market is expected to grow from $3.8 billion in 2020 to $5.4 billion by 2025. In the US, more than $8 billion yearly revenue is generated through commercial parking lots and the garages industry. But even with such a large market, only a small percentage of parking facilities have adopted smart parking technologies.

8. SAMPLE CODES:

8.1 PYTHON CODE:

```
import cv2

import pickle

import cv zone

import numpy as np

#video feed

cap = cv2.VideoCapture('carPark.mp4')

with open('CarParkPos', 'rb') as f:

   posList = pickle.load(f)

width, height = 107, 48

def checkParkingSpace(imgPro):

   spaceCounter = 0

   for pos in posList:

      x,y = pos

      imgCrop = imgPro[y:y+height,x:x+width]

      #cv2.imshow(str(x*y),imgCrop)

      count = cv2.countNonZero(imgCrop)

      cvzone.putTextRect(img, str(count), (x, y + height - 3), scale=1.5,

                     thickness=2, offset=0, colorR=(0, 0, 255))

         if count<900:

         color = (0,255,0)

         thickness = 5

         spaceCounter += 1
```

```
        else:
            color = (0,0,255)
            thickness = 2
        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), color, thickness)
      cvzone.putTextRect(img,f'Free:
{spaceCounter}/{len(posList)}',(100,50), scale = 3,
          thickness`=5, offset=20,colorR=(0,200,0))
while True:
    if cap.get(cv2.CAP_PROP_POS_FRAMES) ==
cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES,0)
    success, img = cap.read()
    imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (3,3),1)
 imgThreshold = cv2.adaptiveThreshold(imgBlur,255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
  cv2.THRESH_BINARY_INV, 25, 16)
 imgMedian = cv2.medianBlur(imgThreshold,5)
    kernel = np.ones((3,3),np.uint8)
    imgDilate = cv2.dilate(imgMedian,kernel,iterations=1
  checkParkingSpace(imgDilate)
    cv2.imshow("Image", img)
    #cv2.imshow("ImageBlur", imgBlur)
    #cv2.imshow("ImageThreshold", imgMedian)
```

8.2 PARKING SPACE PICKER ACTIVITY:

```
import cv2

import pickle

width, height = 107, 48

try:

    with open('CarParkPos', 'rb') as f:

        posList = pickle.load(f)

except

posList = []

def mouseClick(events,x,y,flags,params):

    if events == cv2.EVENT_LBUTTONDOWN:

        posList.append((x,y))

    if events == cv2.EVENT_RBUTTONDOWN:

        for i,pos in enumerate(posList):

            x1,y1 = pos

            if x1<x<x1+width and y1<y<y1+height:

                posList.pop(i)

    with open('CarParkPos','wb') as f:

        pickle.dump(posList, f)

while True:

    img = cv2.imread('carParkImg.png')

    for pos in posList:

        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), (255, 0, 255), 2)

    cv2.imshow("Image",img)
```

```
cv2.setMouseCallback("Image",mouseClick)

cv2.waitKey(1)
```

8.3 ARDUINO CODE:

```
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11);
LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line
display
const int trig_1 = 2;
const int echo_1 = 3;
const int trig_2 = 4;
const int echo_2 = 5;
const int trig_3 = 6;
const int echo_3 = 7;
float distanceCM_1 = 0, resultCM_1 = 0;
float distanceCM_2 = 0,resultCM_2 = 0;
float distanceCM_3 = 0, resultCM_3 = 0;
long Time_1, Time_2, Time_3;
float car_1, car_2, car_3;
float Dist_1 = 8.0, Dist_2 = 8.0, Dist_3 = 8.0;
int total = 0, timer_cnt = 0;
void setup()
{
  mySerial.begin(115200);
  pinMode(trig_1, OUTPUT);
  pinMode(trig_2, OUTPUT);
  pinMode(trig_3, OUTPUT);
  pinMode(echo_1, INPUT);
  pinMode(echo_2, INPUT);
  pinMode(echo_3, INPUT);
  digitalWrite(trig_1, LOW);
  digitalWrite(trig_2, LOW);
```

```
  digitalWrite(trig_3, LOW);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("  IoT CAR PARK");
  lcd.setCursor(0, 1);
  lcd.print(" MONITOR SYSTEM");
  delay(2000);
  lcd.clear();
}
void loop()
{
  total = 0;
  car_1 = sensor_1();
  car_2 = sensor_2();
  car_3 = sensor_3();
  lcd.setCursor(0, 0);
  lcd.print("CAR1:");

 if (car_1 <= Dist_1)
  {
   lcd.print("OK ");
  }
  else
  {
   total += 1;
  }
  if (car_1 > Dist_1)   lcd.print("NO ");
  lcd.print("CAR2:");
  if (car_2 <= Dist_2)
  {
   lcd.print("OK ");
  }
```

```
 else
 {
  total += 1;
 }
 if (car_2 > Dist_2)
 lcd.print("NO ");
 lcd.setCursor(0, 1);
 lcd.print("CAR3:");
 if (car_3 <= Dist_3)
 {
  lcd.print("OK ");
 }
 else {
  total += 1;
 }
 if (car_3 > Dist_3)
lcd.print("NO ");
 lcd.print("FREE:");
 lcd.print(total);
 if (timer_cnt >= 50)
 {
  mySerial.print('*');
  mySerial.print(total);
  mySerial  .println('#');
  timer_cnt = 0;
 }
 timer_cnt += 1;
 delay(200);
}
float sensor_1(void)
{
 digitalWrite(trig_1, HIGH);
 delayMicroseconds(10);
```

```
  digitalWrite(trig_1, LOW);
  Time_1 = pulseIn(echo_1, HIGH);
  distanceCM_1 = Time_1 * 0.034;
  return resultCM_1 = distanceCM_1 / 2;
}
float sensor_2(void)
{
  digitalWrite(trig_2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_2, LOW);
  Time_2 = pulseIn(echo_2, HIGH);
  distanceCM_2 = Time_2 * 0.034;
  return resultCM_2 = distanceCM_2 / 2;
}
float sensor_3(void)
{
  digitalWrite(trig_3, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_3, LOW);
  Time_3 = pulseIn(echo_3, HIGH);
  distanceCM_3 = Time_3 * 0.034;
  return resultCM_3 = distanceCM_3 / 2;
```

9. Testing:

This system testing is conducted in two main circuits, i.e. sensor circuit and RFID sensor.

9.1 SENSOR CIRCUIT TESTING:

The test of the circuit data communication of the whole circuit to and from cloud is done by detecting the existence of the car 10 times with the assumption of 150 cm car height and floor to ceiling distance of 350 cm. The measuring point of this test is the success of the circuit in sending data, the length of time required, and the validity of the data. Calculation of respond time on the Arduino and server is done by the progr am by recording the difference between the start and end time of the program. The calculation of respond time on cloud is done by calculating the difference between server sent time and Ubidots received time. Based on the testing result as shown in Table 3 below, The sensor circuit has 100 % success rate for sending data to cloud with an average total required time of 6.4004 s for each data. Data errors were not found during the test.

Table 9.1. Sensor circuit testing results

Test	1010001	Response Time			Total Time (s)	Data Error
		Arduino (ms)	Server (ms)	Ubi dots (s)		
1	Pass	237	228	4	4.465	0
2	Pass	249	315	6	6.564	0
3	Pass	312	226	6	6.538	0
4	Pass	369	235	6	6.604	0
5	Pass	253	234	6	6.487	0
6	Pass	245	209	6	6.454	0
7	Pass	232	210	6	6.442	0
8	Pass	274	229	6	6.503	0
9	Pass	233	225	6	6.458	0
10	Pass	268	221	7	7.489	0
Avg	100 %	267.2	233.2	5.9	6.400 4	0 %

9.2 RFID CIRCUIT TESTING:

The method used to test RFID circuit communications to cloud is by tapping RFID cards ten times. In the program, there are already methods of calculating the respond time such as testing the sensor circuit. The measured points are the success of the circuit sending the data, the time it takes to send from the circuit to the cloud, and the validity of the data se nt with the received.

Test	RFID 1	Response Time			Total Time (s)	Data Error
		Arduino (ms)	Server (ms)	Ubi dots (s)		
1	Pass	1 396	751	7	9.147	0
2	Pass	925	362	8	9.287	0
3	Pass	1 406	776	7	9.182	0
4	Pass	1 303	550	7	8.853	0
5	Pass	2 156	500	2	4.656	0
6	Pass	1 488	603	7	9.091	0
7	Pass	1 714	622	8	10.333 6	0
8	Pass	1 214	161 4	6	8.828	0
9	Pass	1 062	121 1	3	5.273	0
10	Pass	1 302	546	6	7.848	0
Avg	100 %	1 396.6	753.5	6.1	8.250 1	0 %

Table 9.2. RFID circuit testing result.

RFID circuits have a 100% success rate in sending data to Ubidots cloud with the percentage of sent data error is 0%. The average time it takes to send data from scratch when the card is detected until the data received by Ubidots is 8.2501 s. The process in Arduino takes an average of 1 396.6 ms and 753.5 ms is on the server. The longest time spent when sending data from the server to Ubidots is 6.1 s.

9.3 CAR DETECTION:

Testing the accuracy of sensor readings conducted by car detection test with the assumption that the car has a height of 150 cm. Testing is done as much as ten times for each circuit of sensors by shifting obstacles. If there is a parking car, then the obstacles will be placed between the sensor and the floor. If the condition of the parking slot is empty, then the barrier will be removed so that the sensor will immediately detect the floor.

Car Present (Car Height 150 cm) - (Ceiling distance 350 cm)						
Test	1010001	1010002	1010003	1010004	1010005	1010006
1	Detected	Detected	Detected	Detected	Detected	Detected
2	Detected	Detected	Detected	Detected	Detected	Detected
3	Detected	Detected	Detected	Detected	Detected	Detected
4	Detected	Detected	Detected	Detected	Detected	Detected
5	Detected	Detected	Detected	Detected	Detected	Detected
6	Detected	Detected	Detected	Detected	Detected	Detected
7	Detected	Detected	Detected	Detected	Detected	Detected
8	Detected	Detected	Detected	Detected	Detected	Detected
9	Detected	Detected	Detected	Detected	Detected	Detected
10	Detected	Detected	Detected	Detected	Detected	Detected

Table 9.3. Car detection testing with occupied parking slot condition.

Car Not Present - (Ceiling distance 350 cm)						
Test	1010001	1010002	1010003	1010004	1010005	1010006
1	Empty	Empty	Empty	Empty	Empty	Empty
2	Empty	Empty	Empty	Empty	Empty	Empty
3	Empty	Empty	Empty	Empty	Empty	Empty
4	Empty	Empty	Empty	Empty	Empty	Empty
5	Empty	Empty	Empty	Empty	Empty	Empty
6	Empty	Empty	Empty	Empty	Empty	Empty
7	Empty	Empty	Empty	Empty	Empty	Empty
8	Empty	Empty	Empty	Empty	Empty	Empty
9	Empty	Empty	Empty	Empty	Empty	Empty
10	Empty	Empty	Empty	Empty	Empty	Empty

Table 9.4. Car detection testing with unoccupied parking slot condition.

There are six circuits that are used for the testing process. From the two tables of sensor testing, it can be concluded that the whole circuits of managed sensors is able to detect the presence of the car accurately. Each parking application is unique, from off-street parking to underground parking to complex facilities with a combination of parking lots, ramps, or underground garages. Many parking applications depend on accurate vehicle counting to reliably verify when a garage is full, many require reliable sensing in order to activate a pay station, and many can benefit from systems that direct drivers to available parking spots. It is therefore important to match the right technology to meet the demands of the application. Some technologies, like ultrasonic sensors, are best suited for covered parking applications where they are sheltered from the elements that can negatively impact sensing function. Other technologies, like wireless magnetometers, excel in harsh outdoor conditions.

10. RESULTS:

An IoT-based smart parking system provides real-time data on parking space availability, pricing, payments, and beyond, evolving as a helpful tool for businesses and consumers. It positively impacts the environment and traffic. It also ensures efficient parking reservation and management.

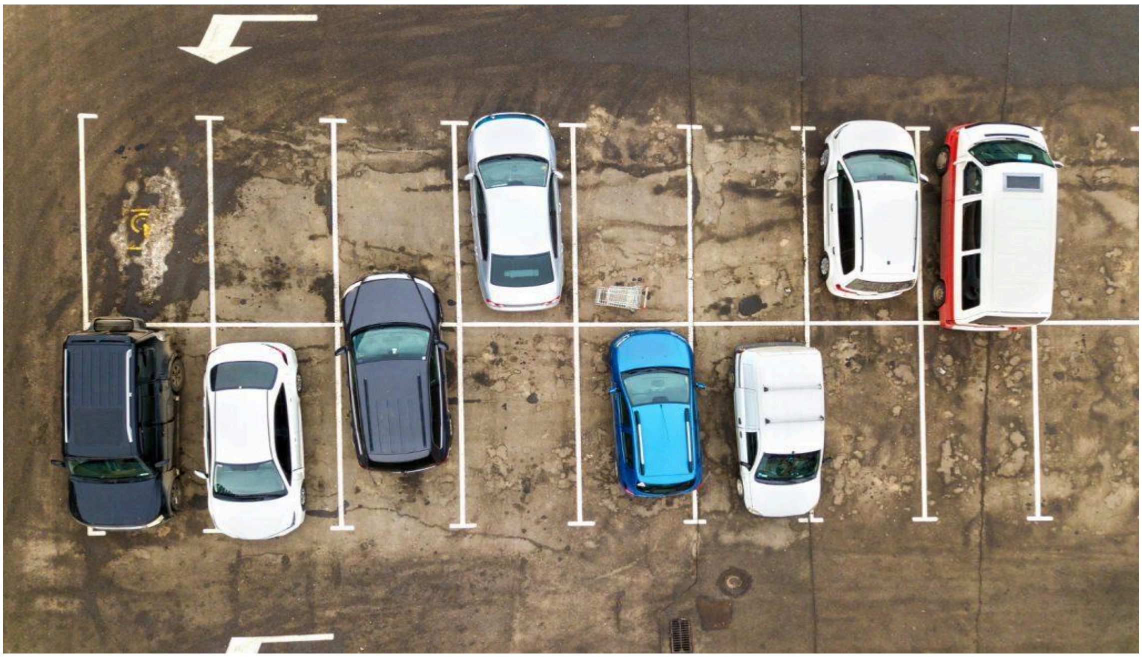


Fig.No.10.1: Result 1

Fig.No.10.2: Result 2

Fig.No.10.3: Thingspeak Private View:

Fig.No.10.4: Thingspeak public view

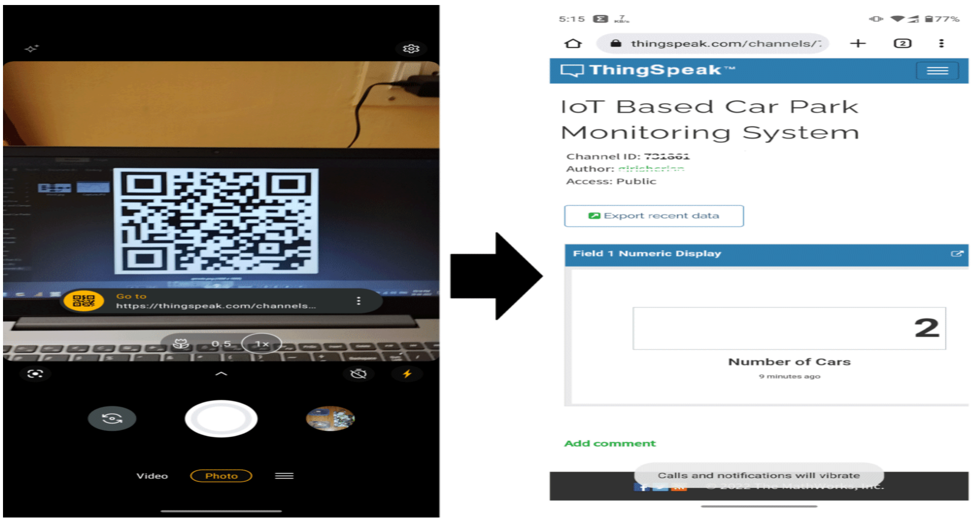


Fig.No.10.5: Thingspeak public view on smartphone:

11.CONCLUSION AND FUTURE ENHANCEMENT:

There are three main IoT sensors for vehicle presence detection. For city street parking lots management, it is more suitable to install a wireless magnetometer sensor due to street parking may include both indoor or outdoor spaces which require flexibility on installation location. Besides, the wireless magnetometer sensor is able to withstand normal weather condition disruption. In addition, installing wireless magnetometer sensors only require a small inch drill hole which can be done quickly making it an ideal choice for large scale city implementation. However, an IoT smart parking system affects a whole network environment thus emphasizing the importance of personal security and privacy protection. It is the government's responsibility to establish related regulations to monitor data usage and avoid privacy issues from happening . Besides, since the traffic system involves life safety issues, at the design stage, the government should take safety issues into consideration and educate the designers with effective prevention approaches.

such as mitigate the impact of security vulnerabilities through patching and establish vulnerability management policies to promote security updates. According to the U.S. Federal guidance report of the automotive industry aiming to improve motor vehicle cybersecurity, indicated that it is important for infrastructure operators and cloud computing service providers to bear the notification obligations.

In other words, the government needs to clarify the data acquisition level among different official departments and local companies and set up proper information security management systems to undergo key actions including education and training of information security, monitor and management approach, safety inspection, and testing. IoT realizes the vision of connecting and communicating with individuals through computing and analyzing a vast amount of information sources. However, to successfully adopt new technology requires a proper framework and clear understanding of the potential challenges and associated issues.

The parking industry is in the midst of a transformation. Rapid technological advances are enabling a new breed of parking solutions that are more efficient, convenient, and sustainable. The smart parking management system's independent operating time will be extended via the LoRaWAN protocol, designed for low power consumption.

In the future, you do not need to change IoT sensors, devices, and gateways batteries more frequently than once every two to five years or according to the LoRa Alliance's specifications.

Additionally, the IoT-driven smart parking system is made to expand from a single gateway deployment to massive worldwide networks with billions of IoT devices. Using this or any other LPWAN protocol for a smart parking facility would be fantastic. You may use ioT-based advancements coupled with data Science, Machine Learning (ML), and Computer Vision to automate the process. The technology will facilitate car number identification and auto payment deduction to make smart parking seamless.

The introduction of Autonomous Vehicles (AVs) is anticipated to have a substantial impact on the future of the IoT-enabled smart parking sector. Self-parking automobiles, specialized AV parking lots, and robotic parking valets are already being tested in several places worldwide.

Cities and parking managers from across the globe are well aware of how vital smart parking is. Many other technological advancements will be crucial for this upcoming generation of parking facilities. Some essential elements include next-generation digital cameras, wireless connectivity (like 5G), and Big Data analysis.

12.REFERENCES :

[1] Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. Computer Networks, 51(4), 921-960. doi:10.1016/j.comnet.2006.10.002.

[2] Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805. doi:10.1016/j.comnet.2010.05.010.

[3] Chen, Z., & Xiaoping, Z. (2014). Toward the Internet of Things application and management: A practical approach. In (pp. 1-6).

[4] FTC, Internet of Things: Privacy & Security in a Connected World (2015), retrieved from: https://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf.

[5] Gartner, G. (2014). Chip sector to benefit from Internet of Things, Gartner says. Electronics News, p. Electronics News, Nov 2014.

[6] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660. doi:10.1016/j.future.2013.01.010.

[7] Guilbert, D., Le Bastard, C., Sio-Song, I., & Yide, W. (2016). State Machine for Detecting Vehicles by Magnetometer Sensors. IEEE Sensors Journal, 16(13), 5127-5128. doi:10.1109/JSEN.2016.2560903.

[8] Jeon, S., Kwon, E., & Jung, I. (2014). Traffic measurement on multiple drive lanes with wireless ultrasonic sensors. Sensors (Switzerland), 14(12), 22891-22906. doi:10.3390/s141222891.

[9] Ji, Z., Ganchev, I., O'Droma, M., Zhao, L., & Zhang, X. (2014). A cloud-based car parking middleware for IoT-based smart cities: Design and implementation. Sensors (Switzerland), 14(12), 22372-22393.

[10] Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431-440. doi:10.1016/j.bushor.2015.03.008

[11] Miao, Y., & Bu, Y. (2010). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In (pp. 69-72).

[12] Nikitas, A., Kougias, I., Alyavina, E., & Njoya Tchouamou, E. (2017). How can autonomous and connected vehicles, electromobility, BRT, hyperloop, shared use mobility and mobility-as-a-service shape transport futures for the context of smart cities?. Urban Science, 1(4), 36.

[13] NHTSA, U.S. DOT issues Federal guidance to the automotive industry for improving motor vehicle cybersecurity (2016/10/24), retrieved from: https://www.nhtsa.gov/press-releases/us-dot-issues-federal-gu idance-automotive-industry-improving-motor-vehicle.

[14] Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges for realizing the Internet of Things. Cluster of European Research Projects on the Internet of Things, European Commision, 3(3), 34-36.

[15] Taherkhani, M., Kawaguchi, R., Shirmohammad, N., & Sato, M. (2016). BlueParking: An IoT based Parking Reservation Service for Smart Cities. In (Vol. 24-25-, pp. 86-88).

[16] Tang, Y., Zhang, C., Gu, R., Li, P., & Yang, B. (2017). Vehicle detection and recognition for intelligent traffic surveillance system. An International Journal, 76(4), 5817-5832.

[17] Filippov, Dmitry (November 19, 2014). "Announcing General Availability of PyCharm 4". PyCharm Blog. Retrieved 24 February 2015.