Fabrication of Automatic car parking toll gate opener by using Arduino UNO

Nayan Atul Gawali[1], Sayyad Jishan Muntaj[2], Devank Sanjay Mahajan[3], Gangesh Anantrao chopade [4], Chavan R.B[5],Virbhadra Narayanrao Hule[6]

[1,2,3,4]*Student, Mechanical Engineering, Shri Shivaji Polytechnic Institute, Parbhani*

[5,6]*Professor, Mechanical Engineering, Shri Shivaji Polytechnic Institute, Parbhani*

Abstract— **The Automatic Car Parking Toll Gate System is an intelligent solution designed to automate vehicle entry, exit, and toll collection in parking areas. This system reduces human intervention, enhances efficiency, and improves security by integrating technologies such as RFID (Radio Frequency Identification), Automatic Number Plate Recognition (ANPR), and smart card systems. This project significantly reduces traffic congestion at toll gates, eliminates manual errors, and enhances security by keeping a digital log of all vehicle movements. It is ideal for implementation in shopping malls, corporate offices, airports, residential complexes, and public parking facilities. By integrating IoT and cloud-based monitoring, the system can provide real-time parking status updates and allow users to reserve parking spaces in advance, making it a smart, efficient, and scalable parking management solution. Key Words: Automatic Parking System, Toll Gate Automation, RFID-based Parking**

I. INTRODUCTION

The fabrication of an automatic car parking toll gate opener using Arduino UNO is an innovative project that integrates embedded systems, sensors, and electromechanical components to automate vehicle entry and exit processes. In modern urban environments, the increasing number of vehicles has made manual toll collection and parking management inefficient, time-consuming, and prone to errors. This project addresses these issues by designing a smart system capable of detecting vehicles, verifying access (if required), and automatically operating a barrier gate without human intervention. The Arduino UNO microcontroller serves as the central processing unit of the system, receiving signals from input devices such as infrared (IR) sensors, ultrasonic sensors, or RFID modules, and controlling output devices like servo motors, LCD displays, and buzzers. The system enhances efficiency, reduces congestion, improves security, and provides a cost-effective solution for parking lots, residential complexes, shopping malls, and toll booths.

The working of the system is based on real-time detection and control logic. When a vehicle approaches the gate, an IR sensor or ultrasonic sensor detects its presence by sensing interruption in the infrared beam or measuring distance changes. This detection signal is transmitted to the Arduino UNO, which processes the input according to the programmed logic. In advanced configurations, an RFID module is incorporated to authenticate the vehicle using RFID tags or cards assigned to authorized users. If the system detects a valid signal or authorized entry, the Arduino sends a control signal to the servo motor, which rotates to lift the barrier arm and allow the vehicle to pass. After a predefined delay or once the vehicle crosses the second sensor, the Arduino commands the servo motor to return to its original position, thereby closing the gate. Simultaneously, an LCD display can show messages such as “Access Granted,” “Access Denied,” or “Welcome,” while a buzzer can provide audio alerts in case of unauthorized access or system errors.

The fabrication process of this system involves several stages including design, component selection, circuit assembly, programming, and testing. Initially, a proper design layout is prepared, including a block diagram and circuit diagram that define the interaction between various components. The hardware setup includes mounting the Arduino UNO on a stable base, connecting sensors to digital input pins, interfacing the servo motor with a PWM pin for precise angular control, and optionally integrating an RFID module

using SPI communication protocol. The LCD display is connected using either 4-bit or I2C mode to minimize wiring complexity. A physical model of the gate is fabricated using lightweight materials such as plastic, wood, or aluminum, with the servo motor mechanically coupled to the barrier arm to ensure smooth opening and closing action. Proper power supply management is crucial, as the Arduino typically operates at 5V while some components may require external power sources.

Programming plays a critical role in the functionality of the system. The Arduino is programmed using the Arduino IDE with Embedded C/C++ language. The code includes libraries for servo motor control, RFID communication, and LCD display handling. The logic is written to continuously monitor sensor inputs, validate conditions, and execute corresponding actions such as rotating the servo motor to specific angles (e.g., 0° for closed and 90° for open). Time delays and conditional statements are used to ensure proper sequencing of operations. Debouncing techniques and filtering may be implemented to avoid false triggering due to noise or environmental factors. In RFID-based systems, unique tag IDs are stored in the program memory, and incoming data from the RFID reader is compared against these stored values to grant or deny access.

From a results and performance perspective, the fabricated system demonstrates reliable and efficient operation under various conditions. It significantly reduces human effort and operational delays, allowing vehicles to pass through the gate smoothly and quickly. The response time of the system is minimal, and the accuracy of detection is high when sensors are properly calibrated. The inclusion of RFID enhances security by restricting access to authorized users only. The system is also scalable and can be further enhanced with features such as automatic billing, integration with IoT platforms for remote monitoring, use of cameras for number plate recognition, and data logging for vehicle tracking. Overall, the project successfully achieves automation in toll gate operation and serves as an excellent example of applying embedded systems in real-world applications.

In conclusion, the fabrication of an automatic car parking toll gate opener using Arduino UNO is a practical and efficient solution for modern traffic and parking management challenges. It combines simplicity, low cost, and high functionality, making it suitable for both academic projects and real-life implementation. The system not only improves operational efficiency but also provides a foundation for further advancements in smart transportation systems and automation technologies.

BLOCK DIAGRAM:

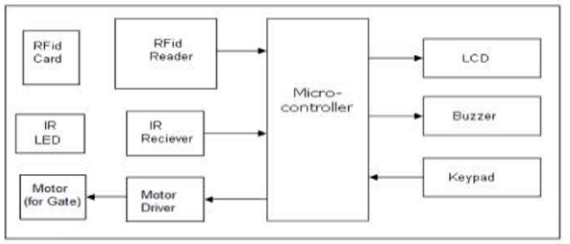

Fig 1 : Blockdiagram

In Today's world wherein we have parking spaces everywhere due to the ever increasing number of vehicles it is only sensible to look for smarter alternatives. This led us to design an automated guidance system which directs cars to empty spots, ad also an automatic charging system. The main advantages are that it completely automates our work and hence we don't have to designate a specific personal for this task. As in most high profile places a person is present for this purpose. The components required to do so are also very cheap and the implementation logic is simple enough to be implemented on a large scale. The availability of parking spaces or the ease with which one can park becomes an issue in malls and public parking spaces. Which is why stopping to collect fees, takes a long time, and creates unnecessary traffic. Even with a lot of extensive labor parking is still inefficient and not as effectively coordinated. This leads to a need of an effective tolling system which can integrate itself with any infrastructure. This system can be fully automated with minimum manual intervention and made modular so as to integrate it into any topology of parking spaces. This system would not include any complications of having a necessary App or a device to use it; it can be accessed at any parking space by any car. Similarly, tolling cars and especially large vehicle have also become a problem. The time they take (statistically proven to be around a thousand hours per week) is too much loss of a precious

resource. Further discrepancies due to blacklisted vehicles/ unaccounted for vehicles, etc. cause security problems. We provide a alternate, efficient, automated system to avoid and address these issues.

Design Details

Arduino Uno: The Arduino is programmed to read inputs from the ultrasonic sensor and control the servo motor based on those inputs. The program is written in the Arduino IDE, using simple logic to open the gate when the sensor detects an object and close the gate once the object has passed. Ultrasonic Sensor (HC-SR04): Positioned strategically near the toll gate, the silttamusic sensor continuously emits sound waves and waits for the reflected waves to return. The sensor measures the time taken for the sound waves to bounce hack and calculates the distance, If the distance is less than the threshold value (e.g., 30 cm),the sensor signals the Arduino to open the gate. Servo Motor (5G90): The servo mour receives signals from the Arkano and rotates accordingly. The motor in attached to the toll gate mechariam, and its rotation opens or closes the gate. The motor's movesnem is controlled through PWM signals sem by the Arduino, allowing for precise control of the gate's movement. Circuit Design: The clecuit is designed using a breadboard in simplify the connectiom between components. The The ultrasonic sensne is connected to the Arduino's digital pias, while the servo motor is connected to a PWM-enabled pin. Proper grounding and voltage regulation ensure the system operates efficiently.

Working Principle:

Vehicle Detection: The ulirasenac naur continuously emits ultrasonic waves, Whim a vehicle approaches the toll gate, the sensor detects the reflected waves and calculates the distance between the vehicle and the gate. If the vehicle is within the predefined detection range, a signal is sent to the Ardaino to initiate the gate opening process Gate Operation: Upon recriving the signal from the sensor, the Arduino processes the information and sends a PWM (Pulse Width Modulation) signal to the servo motor, instructing it is rotate and lift the gate. The servo motor rotates between D and 180 degrees, corresponding to the open and closed positions of the gate Gate Closing: After the vehicle passes through the gate, the ultrasonic sensor no longer detects an object in its range. This triggers the Arduino to send another signal to the servo motor, causing it to rotate back to its initial position and close the gate. Power Management: The system is powered by an external power source connected to the Arduino, which distributes power to the other components via the breadboard. Proper voltage regulation is essential to ensure smooth operation of the sensor and motor

Step-by-Step Working Process

1. Vehicle Detection & Identification
   • When a vehicle approaches the parking entrance, it is detected by a sensor (IR sensor, ultrasonic sensor, or loop detector).
   • The system identifies the vehicle using one of the following methods:
   o RFID (Radio Frequency Identification): The vehicle is equipped with an RFID tag, which is scanned by an RFID reader for authentication.
   o ANPR (Automatic Number Plate Recognition): A camera captures the license plate and verifies it in the system database.
   o Barcode or Smart Card: Users scan their card to gain entry.
2. Authentication & Access Control
   • The system checks whether the vehicle is authorized or if payment needs to be processed.
   • If the vehicle has pre-registered access (such as monthly subscriptions or employee parking), it is allowed to enter without further action.
   • For visitors, the system may generate a ticket with a QR code or barcode for later payment.
3. Parking Slot Allocation & Monitoring
   • Once the vehicle enters, IoT-based sensors update the parking slot availability in real time.
   • A digital display or mobile app directs drivers to available parking spaces.
   • Some systems allow users to reserve a parking spot in advance via a mobile application.
4. Toll Collection & Payment Process • When the driver exits, the system calculates the parking fee based on:
   o Time-based charges (hourly/daily rates).
   o Fixed fee parking (flat rate for entry).
   o Subscription or prepaid account deduction (automatic payment from a linked wallet).

• Payment is processed using one of the following methods:
o Automatic deduction from an RFID account (linked to bank/wallet).
o Mobile app payments (UPI, credit/debit card, e-wallet).
o QR code or barcode scanning for online payment.
o Cash or card payment at a kiosk (optional).

5. Exit & Barrier Control
• After successful payment, the system triggers the boom barrier to open, allowing the vehicle to exit.
• If the payment fails or the vehicle is unauthorized, the barrier remains closed, and an alert is sent to security personnel.
• The system logs the entry and exit time, vehicle details, and payment status in a database for security and analytics.

RELATED WORKS

A lot of research has been done to automate toll in the past few years. Some of the works are : Reference [1] Here they have used GSM module, a high range detection, instead of a near range Communication, thus enabling transmission of data to be global. Reference[2] , a smart project, uses another simple approach of notification through GSM Module. Reference [3] This paper introduces the concept of monitoring the vehicles passing by, and uses the GSM module. Reference [4]In this paper a weight-sensor(FSR) has been included to weigh the vehicle before being tolled. Reference [5] discusses about payment in parking slots on the basis Vehicle number recognition. Reference [6] uses image processing algorithms to identify the plate. Reference [7] A smart system that connects the parking system to the cloud, and has monitoring systems as well.

II. PROPOSED MODEL

In this paper, we have proposed a simple, yet efficient combination of hardware and software solution to billing in two of the largest traffic concerns in our country: Parking Systems and Toll Systems. Using short range detection, we maintain a database of information, automatically calculating fees against type of vehicle, timings, and weight of vehicle. We have also proposed a strict monitoring system, which increases the security enforcements, and reduces the number of blacklisted vehicles on the roads (especially near the borders). These models can be easily integrated with all kinds of infrastructure making it easy to implement, with minimal costs.

HARDWARE REQUIRED

A. RFID: A radio frequency identification reader (RFID reader) is a device used to collect data from an RFID tag, which is used to have individual, unique objects. Radio waves, primarily, are used to transport information from the tag to a reader. Since, the RFID tag does not have to be scanned directly, nor does it require LOS to a reader, it can easily be utilized for passing cars.. The RFID tag must be within the range of an RFID reader(3 to 300 feet), to be read. RFID technology allows multiple items to be scanned and enables fast unique identification of a particular product. B. ArduinoUno: Arduino is an open-source platform based on user friendly hardware and software. It's intended for making interactive and innovative projects. Arduino board designs use an assortment of microprocessors and controllers. The boards feature serial communications interfaces, which are also used for loading programs from PCs and laptops. The microcontrollers are typically programmed using a dialect of languages C and C++. C. LEDs: A light-emitting diode is a two, a p–n junction diode that produces light when switched on. When a suitable voltage is applied to the leads, electrons are able to combine with electron holes within the device, releasing energy in the form of light. LEDs are typically small (less than 1 mm2) and integrated optical components may be used to shape the radiation pattern. Here we have used green LEDs to show “Go” and red ones to show “Stop”.

D.Ultrasonic Sensor (HC-SR04): This sensor detects objects based on the time it takes for an ultrasonic pulse to be sent out, bounce back from an object (in this case, a vehicle), and return to the sensor. The sensor emits sound waves that are inaudible to humans and measures the distance to the object by calculating the time taken for the echo to return. The sensor is placed near the toll gate and continuously monitors the area for any approaching vehicles. If a vehicle is detected within a certain range (e.g.. 30 cm), the sensor

signals the Arduino to open the gate. E. Servo Motor (SG90): The servo motor is a key actuator in the system that controls the movement of the toll gate. When the Arduino receives a signal from the ultrasonic sensor indicating the presence of a vehicle, it instructs the servo motor to rotate, lifting the gate. Once the vehicle has passed and the sensor no longer detects any object, the Arduino sends another signal to the servo motor, instructing it to lower the gate back to its closed position. The servo motor is ideal for this project because it provides precise control over the movement of the gate. F. Breadboard and Wiring: A breadboard is used for prototyping the circuit, allowing for easy and temporary connections between components like the Arduino, sensor, and motor. Jumper wires are used to connect the different components, ensuring that power and data signals are transmitted corectly between them.

III. FLOW CHART

The toll system has detection as well as monitoring modules attached to it. The RFID is triggered only when car is detected by motion sensor

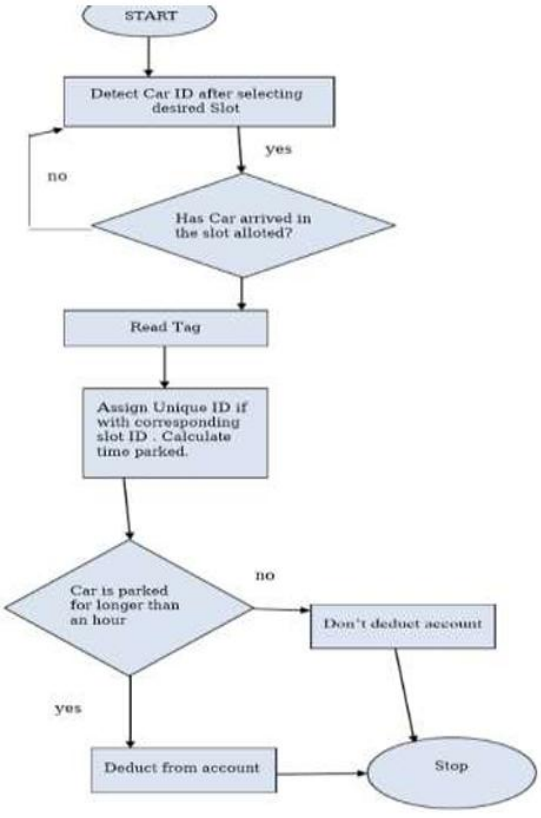

Fig 2 : Flowchart

IV. RESULT

The project successfully achieved all the criteria required to qualify as a smart system, and has effectively achieved the aims of the concept, being a power-efficient, automated and concise approach, involving little to no human involvement, to parking systems. The LEDs that were used, and the logic implemented, being simple yet efficient, made the entire project power-friendly and cost effective. Another plus point of the project is the scalability, infrastructure wise. No major change in the layout, circuit or logic has to be made except changing sensors to industry-level ones. This project thus has the flexibility of fitting in any infrastructural set-up without excessive changing in planning and with no major recurring costs. The developer board that we use (Arduino Uno) is also compatible with large-scale implementation, thus needing no major change in the coding of the entire program. Another major advantage of this project is the

accuracy it assures with an entire module for error handling, taking care not only of the human errors that may occur, but also the accuracy of the functioning, given we have used IR sensors instead of Ultrasonic sensors, thus being able to eliminate the noise factor which might have otherwise been a major source of error. Thus, with the guarantee of accuracy, error handling, scalability, and ease of operation, this smart parking system makes a tedious real world problem smoother and hassle free.

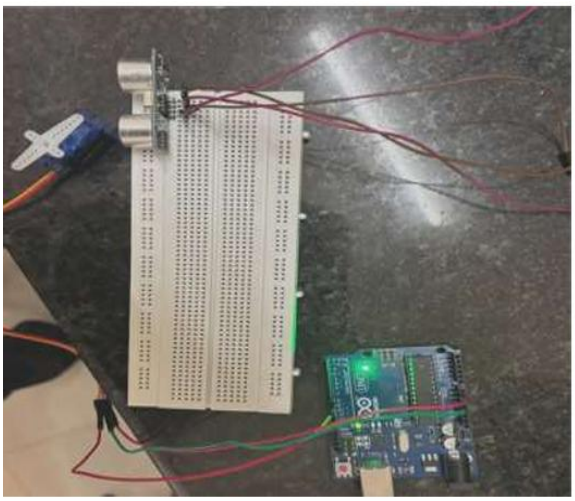

Fig 3 : Snapshot of project

V.CONCLUSION

Our project thus achieved the goals of an optimized parking fee structure where users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities, also traffic is saved by no more queues. This does the job of reduced traffic, where traffic flow increases as fewer cars are required to drive around in search of an open parking space. It achieves multifold benefits like reduced pollution where an optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint. It also provides an enhanced user experience and also increases the safety of these systems. With easy installation, independent operations with less malfunctioning, easy relocation and user-friendly experience, our project turns a dilemma into a delight!

REFERENCES

[1] Alberto Carini and Silvia Malatini., "Automated Toll Plaza System using RFID", IEEE Transactions on Signal Processing, Vol.16, pp.1558- 1563, 2008.

[2] Bram Cornelis, Simon Doclo, Tim Van dan Bogaert, Marc Moonen, Fellow and Jan Wouters., "RFID Based Toll Deduction System", IEEE Transactions on Signal Processing, Vol.18, pp.1452-1458.

[3] International Journal of Computer Science and Mobile Computing A Monthly Journal of Computer Science and Information Technology ISSN 2320–088X IJCSMC, Vol. 4, Issue. 2, February 2015, pg.216 – 224 2010.

[4] Automated Toll Collection System Rafiya Hossain, Moonmoon Ahmed, Md. Mozadded Alfasani, Hasan U. Zaman, and Mohammad Rezaul Islam.

[5] International Journal of Machine Learning and Computing, Vol. 2, No. 2, April 2012. Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition M. M. Rashid, A. Musa, M. Ataur Rahman, and N. Farahana, A. Farhana.

[6] International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 3, March 2016 Copyright to IJIRCCE DOI: 10.15680/IJIRCCE.2016. 0403075 3369 Automatic Parking System using Image Processing A.N.Bhor , P.M.Kale , P.V.Nalawade, C.S.Aryan .

[7] A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies, IEEE. 2169-3536