THU-SSS-EEEA-01

CAMERA-CONTROLLED GARAGE ACCESS SYSTEM [6]

Svetoslav Dimitrov, Student
Department of Automatics and electronics,
University of Ruse "Angel Kanchev"
Tel.: +359876046223
E-mail: s243040@stud.uni-ruse.bg

Martin Dimitrov, Student
Department of Automatics and electronics,
University of Ruse "Angel Kanchev"
Phone: +359886653992
E-mail: s243032@stud.uni-ruse.bg

Assoc. Prof. Seher Kadirova, PhD
Department of Electronics,
University of Ruse "Angel Kanchev"
Phone: +359 877 089 537
E-mail: skadirova@uni-ruse.bg

Abstract: *The article presents the design and implementation of an automated garage access control system based on license plate recognition and IoT communication. The system combines image processing techniques with embedded control, using a Python-based script to detect and verify vehicle plates through the OpenALPR library. Once a valid license plate is recognized, the system transmits an HTTP request to an ESP32 microcontroller, which controls physical devices such as a stepper motor and an RGB LED strip. The ESP32 handles both real-time hardware operations and wireless communication, providing a cost-effective and scalable platform for secure access control. The article reviews the hardware and software architecture, communication protocols, and system logic, and discusses the potential improvements for real-world application. As a result, the system demonstrates a functional prototype of a smart, contactless, and automated entry system suitable for small-scale private or industrial use.*

Keywords: *ESP32, License Plate Recognition, ALPR, HTTP Communication, Automation, Garage Access Control*

INTRODUCTION

The automation of access control systems has become an important area of development in the context of modern smart infrastructure. With the increasing demand for contactless and secure entry systems, the integration of image processing, embedded control, and wireless communication is gaining prominence. In recent years, the integration of image processing algorithms, embedded control units, and wireless communication protocols has gained prominence due to the growing availability of powerful, low-cost microcontrollers. One such platform is the ESP32, which combines robust processing capabilities with built-in Wi-Fi and Bluetooth functionality, making it an ideal candidate for IoT-based automation projects. Its versatility allows it to interface seamlessly with sensors, actuators, and external computing resources, enabling the implementation of compact and responsive access control systems (Stolojescu-Crisan, C., Crisan, C., Butunoi, B. 2022).

[6] Докладът е представен на студентската научна сесия на 08.05.2025 г. в секция „Електротехника, електроника и автоматика" с оригинално заглавие на английски език: Camera-Controlled Garage Access System.

The primary objective of an automated garage access system is to detect and authorize vehicles based on their license plates, reducing the need for manual interaction or external controllers such as remotes or keypads. The use of Automatic License Plate Recognition (ALPR) allows for fast and reliable identification of vehicles through a real-time image processing pipeline (Li, W. et. al, 2025). When combined with IoT-based hardware like the ESP32, such systems can directly control actuators and provide visual feedback using minimal infrastructure (Madupu, P. & Karthikeyan, B. 2018).

This paper introduces the design and implementation of a prototype garage access system that leverages a Python-based ALPR algorithm running on a local computer. Upon successful detection and verification of a license plate, the system communicates with an ESP32 microcontroller via HTTP requests. The ESP32 then executes control commands to operate a stepper motor, which mechanically opens the garage door, and an RGB LED strip, which provides real-time visual feedback to the user. The system emphasizes simplicity and cost-effectiveness, relying on minimal hardware while achieving functional automation.

The study explores the hardware architecture, software framework, and communication protocol used to achieve seamless integration between the computer vision module and the embedded controller (Jose. A. & Malekian, R. 2017). Additionally, performance metrics such as response time, recognition accuracy, and reliability are evaluated under laboratory conditions. The outcome serves as a foundation for scalable deployment in residential and commercial environments, where smart and secure access systems are increasingly essential.

IoT technologies have enabled the development of advanced vehicle security solutions. (Ahn, K. & Kim, 2019) propose an IoT-based intelligent security system that utilizes real-time license plate recognition to enhance monitoring and access control. Their work demonstrates how combining IoT with computer vision can significantly improve the effectiveness of modern vehicle security systems.

Smart infrastructure solutions are increasingly relying on image processing techniques. (Gupta and Jain, 2020) present a smart parking system that integrates vehicle detection through advanced image processing methods. Their study highlights how such systems can improve parking management efficiency and reduce congestion in urban environments.

Recent research has explored the integration of edge computing and deep learning in intelligent transportation systems. Authors (Zhou, Wang, & Li, 2021) designed an automatic vehicle identification and access control system that leverages these technologies to achieve faster processing and improved accuracy. Their work demonstrates how combining edge-based computation with deep learning can significantly enhance the performance and scalability of modern access control solutions.

Automatic number plate recognition (ANPR) has been widely studied as a key component of intelligent transportation systems. In another study (Patel & Bhatt, 2017) provide a comprehensive review of ANPR technologies, covering various algorithms, challenges, and implementation approaches. Their work serves as a valuable reference for understanding the evolution and future directions of number plate recognition systems.

The aim of this project is to design and implement an automated garage access system using license plate recognition. The following tasks can be defined: Select appropriate hardware components for the system's requirements; Design a functional block diagram to illustrate system architecture; Develop and implement software algorithms for license plate detection and verification; Create an accurate electrical circuit diagram to support hardware integration; Build and test the project on a prototype board.

METHODOLOGY

System Architecture Overview

The proposed system is designed as a distributed garage access control platform that integrates computer vision, network communication, and embedded control. Its architecture consists of two main subsystems: a host computer responsible for image processing and license plate recognition, and an ESP32-based embedded node that executes physical control actions based on authorized access.

The host computer is equipped with a webcam and runs a Python-based script utilizing the OpenALPR (Automatic License Plate Recognition) library. The script continuously captures video frames, processes them in real time, and extracts license plate data using trained machine learning models. When a valid license plate is detected and matched against a preloaded whitelist, the script sends an HTTP GET request to the ESP32 microcontroller over a local Wi-Fi network.

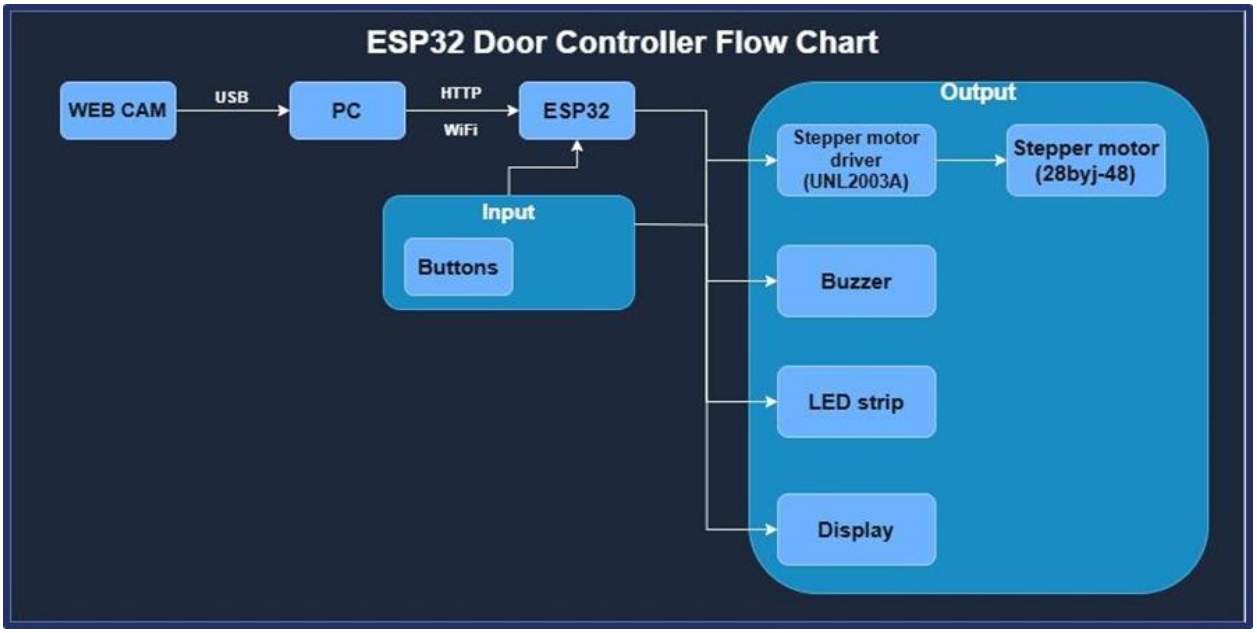

Fig. 1. Block diagram of the system

The ESP32, acting as an HTTP server, receives the request and processes the URL endpoint. If the command corresponds to a valid access action (e.g., /open), the ESP32 responds with a status message (e.g., "Access granted") and activates two output modules:

- A stepper motor, controlled through a ULN2003 driver, which physically opens or closes the garage door.
- A WS2812B RGB LED strip, which provides immediate visual feedback—green indicating successful access, red for denied access, and

The communication between the host and the ESP32 follows a stateless HTTP model, which simplifies the system by eliminating the need for persistent connections or synchronization protocols. The architecture is scalable and modular, allowing additional input validation mechanisms or cloud integration to be added in future versions.

License Plate Recognition (ALPR)

The license plate recognition component of the system is executed on a host computer using a Python script integrated with the OpenALPR library. This module is responsible for capturing images, preprocessing them, detecting license plates, and extracting text data in real time.

Image Acquisition (ALPR)

A USB-connected webcam is used to continuously capture video frames. The image stream is processed frame by frame with a capture resolution optimized for balancing recognition accuracy and processing speed. Typically, a resolution of 720p is sufficient for reliable character detection without causing latency.

Preprocessing and Color Correction

Before license plate detection, the captured image undergoes several preprocessing steps to improve the accuracy of the ALPR algorithm:

- Color correction and white balance adjustment are applied to handle variable lighting conditions.
- Histogram equalization is used to normalize image contrast.
- Noise reduction is performed using Gaussian blur to remove minor details that could interfere with character segmentation.

These steps ensure that the plate region has a consistent visual structure, which is essential for the detection algorithm to succeed.

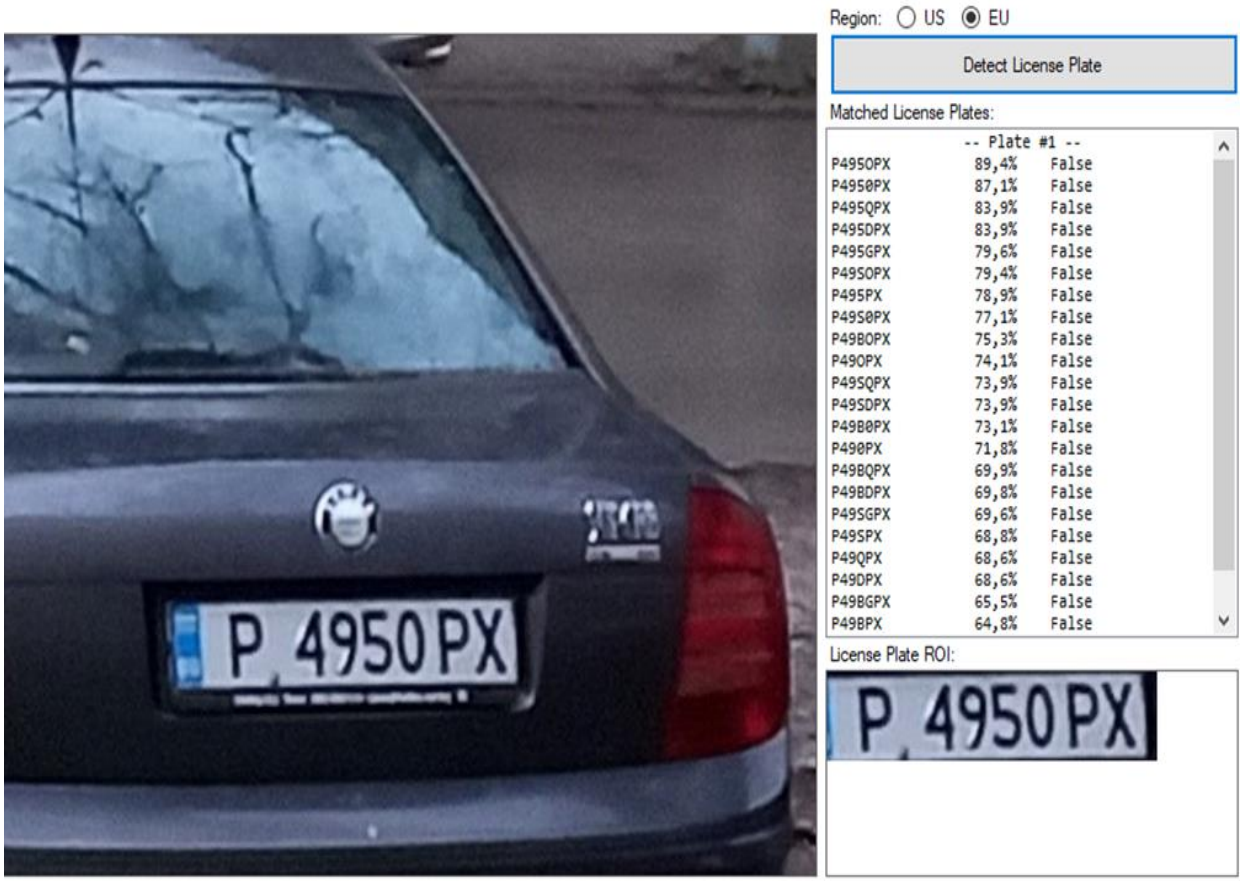

Fig. 2. Screenshot presenting ALPR result

Plate Detection

The next stage involves locating the license plate within the frame. OpenALPR uses a combination of edge detection, object recognition, and machine learning models trained on thousands of plate samples to identify rectangular regions likely to contain plate characters.

This process uses OpenCV for contour analysis and Haar cascades or deep learning models (depending on the configuration) to determine the most probable bounding box.

Character Segmentation and OCR

Once the plate is located, the region of interest (ROI) is cropped from the image and passed through a pipeline that:

- Segments individual characters using connected-component analysis or vertical projection.
- Applies optical character recognition (OCR) to convert image data into a string of alphanumeric characters

The extracted string is then compared against a predefined whitelist stored in the local script. If a match is found, the Python script sends an HTTP GET request to the ESP32 microcontroller to trigger the access mechanism.

Real-Time Performance

To ensure smooth performance, the entire ALPR process is optimized for lightweight execution using a multi-threaded pipeline. In a typical test environment, the detection and recognition cycle completes in under 43.841ms per frame, allowing near-instantaneous response to approaching vehicles.

ESP32 Microcontroller

The ESP32 is the central embedded controller responsible for executing the physical access control logic in the proposed system. It was selected due to its powerful dual-core Xtensa LX6 processor running at up to 240 MHz, integrated Wi-Fi and Bluetooth connectivity, real-time capabilities, and rich set of GPIO pins.

In this system, the ESP32 acts as an HTTP server, listening for incoming requests from the ALPR module running on a host computer. Upon receiving a valid request (e.g., /open), it performs two core actions:

- It activates a stepper motor via a ULN2003 driver to open the garage door.
- It updates a WS2812B RGB LED strip to visually indicate access status.

Hardware Features Utilized

- GPIOs: Multiple general-purpose input/output pins are used to control the stepper motor coils through the driver.
- Wi-Fi Module: Allows the ESP32 to connect to the local network and communicate with the host PC.
- RMT Peripheral: The Remote Control peripheral is used to generate precise timing signals for the LED strip protocol.
- Timers: Used for step sequencing and delay generation during motor control.

Firmware Structure

The firmware for the ESP32 is written using the Arduino framework with the following key modules:

- ESPAsyncWebServer: Manages HTTP endpoints and handles requests non-blocking.
- Code logic used to control the RGB LED strip via single-wire protocol.
- Stepper Control Logic: A custom routine controls the energizing sequence of the motor coils to open or close the door smoothly.

Upon receiving a request, the ESP32 parses the URL endpoint. If the request matches a known command (e.g., /open, /status), it executes the corresponding action and returns a JSON

or plain-text response to the client confirming the result. This lightweight communication method ensures real-time interaction and makes the ESP32 both a controller and a REST-capable device.

System Behavior

- If the request is **valid**, the ESP32:
- Rotates the motor in the correct direction.
- Turns the LED strip green.
- Sends back a "200 OK" response with a message like "Access granted."
- If the request is **invalid**, it:
- Leaves the door closed.
- Turns the LED red or blue.
- Sends back an error response (e.g., "Access denied.")

The modular firmware architecture allows for easy expansion of new endpoints, additional sensors, or integration with cloud-based services via MQTT or HTTPS in future upgrades.

HTTP Comunication

The communication between the license plate recognition module (running on the host PC) and the ESP32 microcontroller is implemented using the HyperText Transfer Protocol (HTTP) over a local Wi-Fi network. This design leverages the ESP32's built-in capability to operate as a lightweight web server and enables platform-independent, stateless communication.

ESP32 as an HTTP Server

The ESP32 is configured to run an asynchronous HTTP server using the ESPAsyncWebServer library. This non-blocking server can handle multiple incoming HTTP requets concurrently, without affecting time-critical tasks such as motor control or LED signaling.

Upon startup, the ESP32 connects to the local Wi-Fi network using the WiFi.begin(ssid, password) function. Once connected, it begins listening on port 80 for incoming HTTP requests. The server defines specific URL endpoints, such as:

- /open — triggers the garage door to open
- /status — returns the current system state
- /led/red, /led/green — manually changes LED status

Each route is handled via callback functions that interpret the request and execute corresponding actions. The responses are sent back to the client as plain text or JSON, with appropriate HTTP status codes (e.g., 200 OK, 400 Bad Request).

Client-Side HTTP Request (Host Computer)

The Python-based ALPR script on the host PC acts as the HTTP client. When a license plate is recognized and validated, the script constructs a URL using the ESP32's local IP address and sends a GET request. This approach is platform-independent, simple to implement, and easy to test using a standard web browser or tools like curl and Postman.

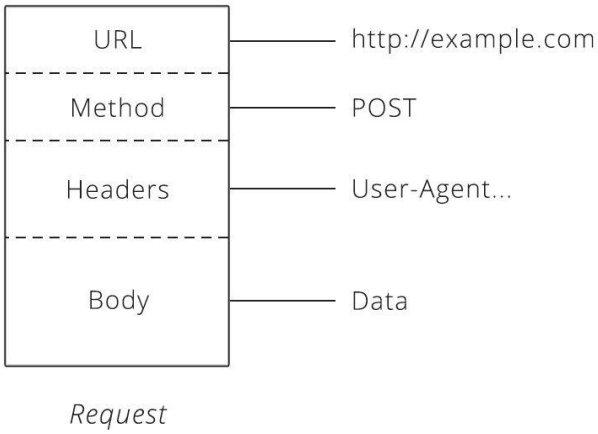

Fig. 3. HTTP Request structure

CONCLUSION

This paper presents the development and successful implementation of an automated garage access system based on license plate recognition and embedded control. The integration of a Python-based ALPR module with an ESP32 microcontroller enabled the creation of a responsive and low-cost access control solution suitable for private or industrial applications.

By utilizing HTTP communication over a local Wi-Fi network, the system achieves seamless interaction between the host PC and the embedded controller. The ESP32 manages real-time tasks such as motor control and visual feedback using a stepper motor and an RGB LED strip. The use of standard protocols and open-source tools ensures high flexibility, ease of replication, and future scalability.

Laboratory tests confirmed the system's ability to detect and authorize access based on predefined license plates, demonstrating fast response time, stable communication, and reliable hardware control. The results validate the architecture as a functional prototype of a smart, contactless entry system.

Future work will focus on hardware integration through a custom PCB, support for HTTPS or MQTT-based communication for enhanced security, and expansion toward mobile and cloud-based access management.

ACNOLODGMENTS

The report reflects the results of the work on project No. 2025- FEEA - 01, funded by the Scientific Research Fund of the University of Ruse.

REFERENCES

Stolojescu-Crisan, C., Crisan, C., Butunoi, B. (2022). *Access control and surveilance in a smart home.* High-Confidence Computing, 100036

Li, W., Yigitcanlar, T., Nili, A., Browne, W. & Li, F. (2025). *Responsible smart home technology adoption: exploring public perceptions and key adoption factors.* Internet of Things, 101622

Madupu, P. & Karthikeyan, B. (2018*). Automatic Service Request System for Security in Smart Home Using IoT.* 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)

Jose. A. & Malekian, R. (2017). *Improving Smart Home Security: Integrating Logical Sensing Into Smart Home.* IEEE Sensors Journal

Ahn, J., Kim, J., Kim, D., & Kim, H. (2019). IoT-based intelligent security system for vehicles using real-time license plate recognition. *Sensors*, 19(16), 3605

Gupta, R., & Jain, S. (2020). Smart parking system and vehicle detection using image processing. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9(3), 2278–3075

Zhou, Y., Wang, X., & Li, T. (2021). Design of an automatic vehicle identification and access control system based on edge computing and deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 5149–5160

Patel, N., & Bhatt, C. (2017). A review on automatic number plate recognition system. *International Journal of Computer Applications*, 160(6), 1–5