

# ДОДАТОК А

## Апробація результатів роботи

Міністерство освіти і науки України



**NURE**

Харківський національний університет  
радіоелектроніки

## ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

**ADED-2025**

(Випуск 1)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitap>



<http://itez.zntu.edu.ua/>



<http://kafedra.kdu.edu.ua>

Харків 2025

УДК 629.4

## MODERN VEHICLE ACCESS CONTROL TECHNOLOGIES AT INDUSTRIAL FACILITIES

**K. O. Nagovitsyn**

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky Ave. 14

E-mail: [kyrylo.nahovitsyn@nure.ua](mailto:kyrylo.nahovitsyn@nure.ua)

**Annotation:** The article explores the use of modern access control technologies for vehicles in industrial environments. It focuses on two of the most widespread solutions – automatic number plate recognition (ANPR) and radio frequency identification (RFID). These technologies are analyzed in terms of their working principles, integration capabilities, and operational efficiency. The study highlights their practical benefits in logistics, security, and automation, while also pointing out technical and implementation-related challenges. The results provide insight into how such systems contribute to improving the safety and efficiency of industrial transport access management.

**Key words:** ANPR, RFID, access control, vehicle identification, industrial infrastructure

## СУЧАСНІ ТЕХНОЛОГІЇ КОНТРОЛЮ ДОСТУПУ ТРАНСПОРТНИХ ЗАСОБІВ НА ПРОМИСЛОВИХ ОБ'ЄКТАХ

**К. О. Наговіцин**

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: [kyrylo.nahovitsyn@nure.ua](mailto:kyrylo.nahovitsyn@nure.ua)

**Анотація:** У статті досліджено застосування сучасних технологій контролю доступу транспортних засобів на територію промислових об'єктів. Основна увага приділяється двом найпоширенішим рішенням – автоматичному розпізнаванню номерних знаків (ANPR) та радіочастотній ідентифікації (RFID). Розглянуто принципи їхньої роботи, переваги в умовах автоматизації логістики та охорони, а також труднощі інтеграції з існуючими системами. Стаття висвітлює як переваги, так і технічні виклики впровадження цих рішень на сучасних виробничих підприємствах.

**Ключові слова:** ANPR, RFID, контроль доступу, ідентифікація транспорту, промислова інфраструктура.

**RELEVANCE OF THE WORK.** In modern conditions of increasing demand for automation and security, the implementation of intelligent access control systems at industrial enterprises is gaining particular relevance. The control of vehicle entry and exit is critical for ensuring perimeter protection, preventing unauthorized access, managing logistics operations, and improving the efficiency of internal business processes. Among the most promising technologies are ANPR (Automatic Number Plate Recognition) and RFID (Radio Frequency Identification), which demonstrate high reliability, accuracy, and identification speed.

**INTRODUCTION.** Industrial enterprises are constantly challenged with the need to organize effective control of vehicle flows. Traditional inspection methods involving human personnel no longer meet modern requirements for speed, accuracy, and security. In the context of industrial digitalization, innovative systems are being actively implemented to automate access control processes without human intervention. This article provides an overview and comparison of two key technologies ANPR and RFID their operating principles, advantages, and practical applications at industrial sites.

**MATERIALS AND RESEARCH RESULTS.** In modern conditions, various technologies are being actively implemented for effective control of vehicle access to industrial facilities. There are many solutions that have already proven their effectiveness in industry, logistics centers, warehouses, and other infrastructure objects. The most common among them are:

- Automatic Number Plate Recognition (ANPR);
- RFID technology.

Automatic Number Plate Recognition (ANPR) technology applies optical character recognition methods to images to read vehicle registration numbers. Dahua Technology has integrated this technology into their surveillance cameras. Thanks to this, ordinary cameras acquire new functions, increasing efficiency in areas where vehicle identification is needed.

The recognition function is implemented at the software level of the cameras. Due to a large database of images and the use of deep learning technologies, the system is able to perform license plate recognition in real time, as well as perform functions such as:

- support for license plate recognition in different languages (English, Arabic, Chinese, Cyrillic, Thai, etc.);
- high recognition accuracy (95%-98% for alphanumeric plates);
- recognition speed of up to 1 second [1].

ANPR is an integrated solution that performs a sequence of operations to identify vehicles. The system is based on the use of surveillance cameras installed in strategically important locations for effective traffic monitoring. This allows obtaining high-quality images of both the vehicle itself and its license plate.

After obtaining the image, it undergoes primary processing aimed at improving visual characteristics – lighting, contrast, and sharpness. This creates favorable conditions for further analysis.

The image is then processed by software that determines the frame with the license plate, coordinates, dimensions, and its position. After this, the system proceeds to the segmentation stage – each character is separated for further recognition.

The characters are analyzed using Optical Character Recognition (OCR) technology, which converts graphic images of characters into text form.

The final stage of the system's operation is interaction with the database. After recognition, the number is verified against available information. In case of a match, the system performs programmed actions: opening the barrier, recording the entry, notifying the operator, or other actions according to settings.

ANPR technology (Fig. 1) is used in various fields:

- monitoring compliance with traffic rules (detection of speeding, violations);
- assisting law enforcement agencies in finding stolen or unregistered vehicles;
- automation of parking systems and facilitating payment for parking;
- optimization of logistics processes through monitoring of transport routes;
- accounting on toll highways for electronic payment systems [2].

Radio Frequency Identification (RFID) is a set of technologies and standards that provide contactless identification of objects using radio signals at distances ranging from a few centimeters to hundreds of kilometers [4]. Unlike other systems, RFID does not require direct visibility or physical contact between devices.



Figure 1 – Example of ANPR technology in operation [3]

An RFID tag (Fig. 2) is a small device that stores data. It consists of an integrated circuit for storing and processing information and an antenna [5].

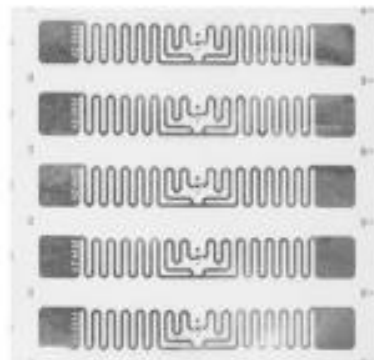


Figure 2 – RFID tag [7]

The functioning of an RFID system is based on the interaction between two main elements: a microchip and an antenna. The reader transmits electromagnetic waves that activate the tag. In response, the tag transmits encrypted information.

The advantage of this technology is the absence of the need for direct visual contact. Radio waves can pass through most materials, allowing tags to be placed inside objects.

A unique identification code is stored in the tag's memory. Modern models have rewritable memory, making them even more versatile. Readers can work both in connection with a central system and autonomously.

The RFID information infrastructure includes software that performs collection, analysis, and processing of data, ensuring the functional integrity of the system [5].

In enterprises, RFID is actively used, for example, in access control systems (Fig. 3). After completing the documentation, a tag is attached to the driver's windshield, which, when read, automatically opens passage through the checkpoint [6].

**CONCLUSIONS.** This article examined modern technologies for vehicle access control at industrial sites, focusing on ANPR and RFID. Both systems offer high accuracy, speed, and efficiency, significantly enhancing security and optimizing logistics processes. ANPR is ideal for

scenarios requiring rapid license plate detection, while RFID offers flexibility and contactless identification without line-of-sight requirements. The implementation of these technologies promotes automation, reduces human error, and improves overall enterprise efficiency.

#### REFERENCES

1. Технологія ANPR (розпізнавання автомобільних знаків) у камерах Dahua URL: <https://dnepsecurity.com/statji/tehnologija-anpr--raspoznawanie-avtomobilnih-znakov--v-kamerah-dahua.html> (дата звернення: 20.04.2024).
2. How Does ANPR Work? A Comprehensive Guide to Automatic Number Plate Recognition URL: [https://carmencloud.com/how-does-anpr-work-a-comprehensive-guide-to-automatic-number-plate-recognition/?utm\\_source=chatgpt.com](https://carmencloud.com/how-does-anpr-work-a-comprehensive-guide-to-automatic-number-plate-recognition/?utm_source=chatgpt.com) (дата звернення: 20.04.2024).
3. HIT ACC ANPR - High Information Technologies SIA. URL: <https://hi-technologies.lv/003-2-2/> (дата звернення: 20.04.2024).
4. RFID технологія | Що таке RFID | Як працює RFID-технологія URL: <https://www.vostok.dp.ua/ukr/infal/rfid/> (дата звернення: 20.04.2024).
5. RFID – технологія автоматичної ідентифікації URL: [https://www.vostok.dp.ua/ukr/infal/rfid/rfid\\_tekhnologiya/](https://www.vostok.dp.ua/ukr/infal/rfid/rfid_tekhnologiya/) (дата звернення: 20.04.2024).
6. RFID-ідентифікація транспорту на агропідприємстві URL: [https://www.vostok.dp.ua/ukr/infal/rfid/rfid\\_resheniya\\_dlya\\_kpp/](https://www.vostok.dp.ua/ukr/infal/rfid/rfid_resheniya_dlya_kpp/) (дата звернення: 20.04.2024).
7. UHF RFID Tag (Set of 5) URL: <https://www.amazon.de/-/en/UHF-RFID-Tag-Set-5/dp/B06XG9HNRR> (дата звернення: 20.04.2024).
8. Vladyslav, Y., & Bronnikov, A. (2020, October). ANALYSIS OF THE CMMI MODEL APPLICATION FOR SOLVING THE TASKS OF CPPS CONTROL PROCESSES AUTOMATION DEVELOPMENT. In The 4 th International scientific and practical conference "Actual trends of modern scientific research"(October 11-13, 2020) MDPС Publishing, Munich, Germany. 2020. 386 p. (p. 128).
9. Yevsieiev, V. V., & Bronnikov, A. I. (2020). Development of databases interconnection "essences" information model for cyber-physical production systems additive cyber design creation automation. Збірник Наукових Праць НУК, №3. С.56-62. DOI [https://doi.org/10.15589/znp2020.3\(481\).7](https://doi.org/10.15589/znp2020.3(481).7)
10. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development. III International scientific-practical conference "Theory, science and practice" (Japan, Tokyo, 5–8 October 2020). P. 398–401. DOI: 10.46299/ISG.2020.II.III
11. Yevsieiev V., Bronnikov A. Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production. The IIth International scientific and practical conference «Development of scientific and practical approaches in the era of globalization» (USA, Boston, 28–30 September. 2020). P.221–226. DOI:10.46299/ISG.2020.II.II.
12. Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. Computational Intelligence and Neuroscience, 2022, Article ID 3046116, <https://doi.org/10.1155/2022/3046116>.
13. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29-43.

**Науковий керівник:** *Бронніков Артем Ігорович, к.т.н., доц., доцент кафедри КІТАР Харківського національного університету радіоелектроніки*

## ДОДАТОК Б

### Програмний код для ESP32-CAM

```

#include "esp_camera.h"
#include <WiFi.h>
#include <HTTPClient.h>
#include <ESP32Servo.h>

//Wi-Fi
const char* ssid = "Tenda_09C8D8";
const char* password = "26321363";

//Сервер
const char* serverName = "http://192.168.0.105:5000/upload";

//Піни
#define TRIG_PIN 2
#define ECHO_PIN 15
#define PIR_PIN 14
#define SERVO_LEFT_PIN 12
#define SERVO_RIGHT_PIN 13

//Змінні
bool accessGranted = false;
bool photoSent = false;

Servo servoLeft;
Servo servoRight;

#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM       5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

```

```

void startCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;
    config.frame_size = FRAMESIZE_VGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
    esp_camera_init(&config);
}

bool sendPhoto() {
    Serial.println("📷 Робимо фото...");
    camera_fb_t* fb = esp_camera_fb_get();
    if (!fb || !fb->buf || fb->len <= 0) {
        Serial.println("❌ Помилка фото");
        return false;
    }

    HTTPClient http;
    http.begin(serverName);
    http.addHeader("Content-Type", "image/jpeg");
    int response = http.POST(fb->buf, fb->len);

    Serial.print("📡 HTTP статус: ");
    Serial.println(response);

    if (response == 200) {
        String res = http.getString();
        Serial.println("📄 Відповідь сервера: " + res);

        if (res.indexOf("OPEN") >= 0) {
            accessGranted = true;
            Serial.println("✅ Доступ дозволено");
        }
    }
}

```

```

    } else if (res.indexOf("NON_FOUND") >= 0) {
        Serial.println("⊖ Доступ заборонено – номер не знайдено у базі");
    } else if (res.indexOf("DENY") >= 0) {
        Serial.println(" ! Номер не розпізнано");
    }
} else {
    Serial.println(" ! Помилка запиту до сервера");
}

http.end();
esp_camera_fb_return(fb);
return true;
}

//Керування воротами
void openGate() {
    servoLeft.writeMicroseconds(500);
    servoRight.writeMicroseconds(2400);
    Serial.println("🚪 Ворота відкриті");
}

void closeGate() {
    servoLeft.writeMicroseconds(1500);
    servoRight.writeMicroseconds(1500);
    Serial.println("🔒 Ворота зачинені");
}

long getDistance() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    long duration = pulseIn(ECHO_PIN, HIGH, 30000);
    return duration * 0.034 / 2;
}

void setup() {
    Serial.begin(115200);

    WiFi.begin(ssid, password);
    Serial.print("📶 З'єднання з Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\n✅ WiFi підключено");
    Serial.print("📶 IP: ");
    Serial.println(WiFi.localIP());
    startCamera();
}

```

```

servoLeft.setPeriodHertz(50);
servoRight.setPeriodHertz(50);
servoLeft.attach(SERVO_LEFT_PIN);
servoRight.attach(SERVO_RIGHT_PIN);
delay(1000);
closeGate();

pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
pinMode(PIR_PIN, INPUT);
}

void loop() {
  long distance = getDistance();
  bool pir = digitalRead(PIR_PIN);

  Serial.print("📏 Відстань: ");
  Serial.print(distance);
  Serial.print(" см | PIR: ");
  Serial.print(pir);
  Serial.print(" | Доступ: ");
  Serial.println(accessGranted);

  if (distance <= 15 && !photoSent) {
    if (sendPhoto()) {
      photoSent = true;

      if (!accessGranted) {
        delay(3000);
        photoSent = false;
      }
    }
  }

  if (accessGranted) {
    openGate();

    Serial.println("⌚ Очікуємо на спрацювання PIR...");
    unsigned long startWait = millis();
    while (digitalRead(PIR_PIN) == LOW && millis() - startWait < 10000) {
      delay(100);
    }

    delay(1000);
    closeGate();

    accessGranted = false;
    photoSent = false;
  }

  delay(200);
}

```

## ДОДАТОК В

### Програмний код для Flask-серверу

```
from flask import Flask, request, send_file, render_template_string, redirect
import os
from datetime import datetime
import easyocr

app = Flask(__name__)
UPLOAD_FOLDER = "uploads"
LOG_FILE = "log.txt"
ALLOWED_FILE = "allowed_plates.txt"

os.makedirs(UPLOAD_FOLDER, exist_ok=True)
reader = easyocr.Reader(['uk', 'ru', 'en'])

def normalize_plate(text):
    cyr_to_lat = str.maketrans({
        "А": "A", "Б": "B", "Е": "E", "І": "I", "К": "K",
        "М": "M", "Н": "H", "О": "O", "П": "P", "С": "C",
        "Т": "T", "У": "Y", "Х": "X", "З": "3"
    })
    return text.translate(cyr_to_lat)

def load_allowed_plates():
    if not os.path.exists(ALLOWED_FILE):
        return []
    with open(ALLOWED_FILE, "r", encoding="utf-8") as f:
        return [normalize_plate(line.strip().replace(" ", "").upper()) for line
in f.readlines()]

def save_allowed_plates(plates):
    with open(ALLOWED_FILE, "w", encoding="utf-8") as f:
        for plate in plates:
            f.write(normalize_plate(plate.strip().replace(" ", "").upper()) +
"\n")
```

```

def log(message):
    timestamp = datetime.now().strftime("[%d.%m.%Y %H:%M:%S]")
    log_entry = f"{timestamp} {message}\n\n"
    print(log_entry.strip())
    with open(LOG_FILE, "a", encoding="utf-8") as f:
        f.write(log_entry)

@app.route('/')
def index():
    allowed = "\n".join(load_allowed_plates())
    return render_template_string("""
<html>
<head>
    <meta charset="utf-8">
    <title>Система розпізнавання</title>
    <style>
        body { font-family: sans-serif; padding: 20px; }
        textarea { width: 100%; height: 200px; }
        pre { max-height: 300px; overflow-y: scroll; background: #f0f0f0;
padding: 10px; }
    </style>
    <script>
        let lastLog = "";
        async function fetchLog() {
            const response = await fetch('/log.txt');
            const text = await response.text();
            if (text !== lastLog) {
                const logElem = document.getElementById("log");
                logElem.innerText = text;
                logElem.scrollTop = logElem.scrollHeight;
                lastLog = text;
            }
        }
        setInterval(fetchLog, 1000);
        window.onload = fetchLog;
    </script>
</head>
<body>
    <h2> 📖 Журнал логів</h2>

```

```

<pre id="log">Завантаження...</pre>

<h3>🚗 Редагування дозволених номерів:</h3>
<form method="POST" action="/save-plates">
    <textarea name="plates">{{allowed}}</textarea><br>
    <button type="submit">💾 Зберегти</button>
</form>
</body>
</html>
""", allowed=allowed)

@app.route('/log.txt')
def get_log():
    return send_file(LOG_FILE, mimetype="text/plain")

@app.route('/save-plates', methods=['POST'])
def save_plates():
    plates = request.form['plates'].strip().splitlines()
    save_allowed_plates(plates)
    log("📄 Список дозволених номерів оновлено.")
    return redirect("/")

@app.route('/upload', methods=['POST'])
def upload_image():
    try:
        image_data = request.data
        timestamp = datetime.now().strftime("%d.%m.%Y_%H-%M-%S")
        temp_filename = os.path.join(UPLOAD_FOLDER, f"temp_{timestamp}.jpg")

        with open(temp_filename, 'wb') as f:
            f.write(image_data)

        results = reader.readtext(temp_filename)
        allowed_plates = load_allowed_plates()

        for (_, text, prob) in results:
            plate = normalize_plate(text.replace(" ", "").upper())
            print(f"🔍 Розпізнано: {plate} (ймовірність: {prob:.2f})")
            if plate:

```

```

        if plate in allowed_plates:
            final_filename = os.path.join(UPLOAD_FOLDER,
f"{plate}_{timestamp}.jpg")
            os.rename(temp_filename, final_filename)
            msg = f"✅ Доступ дозволено для {plate}. Фото збережено:
{final_filename}"
            log(msg)
            return "OPEN", 200
        else:
            unknown_path = os.path.join(UPLOAD_FOLDER,
f"denied_{plate}_{timestamp}.jpg")
            os.rename(temp_filename, unknown_path)
            msg = f"❌ Доступ заборонено. Номер: {plate}. Фото
збережено: {unknown_path}"
            log(msg)
            return "NON_FOUND", 200

# Якщо нічого не розпізнано – зберігаємо лише останнє
for fname in os.listdir(UPLOAD_FOLDER):
    if fname.startswith("unrecognized_"):
        try:
            os.remove(os.path.join(UPLOAD_FOLDER, fname))
        except:
            pass

    unknown_path = os.path.join(UPLOAD_FOLDER,
f"unrecognized_{timestamp}.jpg")
    os.rename(temp_filename, unknown_path)
    msg = f"❓ Номер не розпізнано. Фото збережено: {unknown_path}"
    log(msg)
    return "DENY", 200

except Exception as e:
    error_msg = f"❌ Помилка: {e}"
    log(error_msg)
    return "ERROR", 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

```

**ДОДАТОК Г**  
Демонстраційний матеріал

