

ДОДАТОК А

Апробація результатів наукових досліджень

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2023

(Випуск 2)

[електронне видання]



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



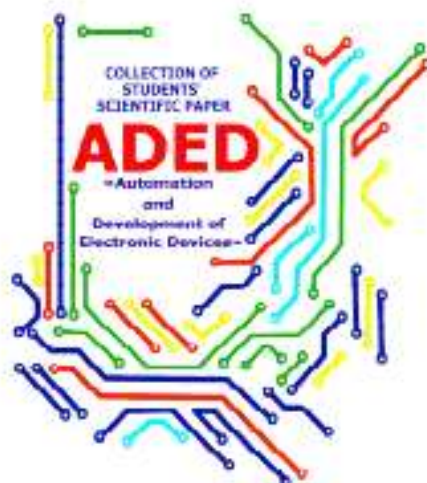
<http://itez.zntu.edu.ua/>



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

Харків 2023

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2023

(Випуск 2)

[електронне видання]

Харків 2023

- Головий редактор** **Невлюдов Ігор Шакирович**, доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харківського національного університету радіоелектроніки.
- Редакційна колегія:** **Филипенко Олександр Іванович**, доктор технічних наук, професор, декан факультету Автоматики та комп'ютеризованих технологій, Харківського національного університету радіоелектроніки.
Цимбал Олександр Михайлович, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харківського національного університету радіоелектроніки.
Андрусевич Анатолій Олександрович, доктор технічних наук, професор, начальник Криворізького коледжу національного авіаційного університету
Косенко Віктор Васильович, доктор технічних наук, професор, зам. директора Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості».
Замірець Микола Васильович, доктор технічних наук, професор, директор Державного підприємства Науково-дослідного технологічного інституту приладобудування.
Свищ Володимир Митрофанович, доктор технічних наук, професор, радник директора Державне науково-виробниче підприємство «Об'єднання Комунар».
Фомовська Олена Владиславівна, кандидат технічних наук, доцент завідувач кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського.
Кухаренко Дмитро Володимирович, кандидат технічних наук, доцент кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського
Демська Наталія Павлівна, кандидат технічних наук, доцент кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харківського національного університету радіоелектроніки.
Фурманова Наталія Іванівна, кандидат технічних наук, доцент, в.о. декана факультета Радіоелектроніки і телекомунікацій, Національного університету «Запорізька політехніка».
- Відповідальний редактор:** **Свесств Владислав В'ячеславович**, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харківського національного університету радіоелектроніки.

УДК 621.396.946

РОЗРОБЛЕННЯ СИСТЕМИ КЕРУВАННЯ МОБІЛЬНИМ ПОШУКОВО-РЯТУВАЛЬНИМ РОБОТОМ

В.С. Головіна

Харківський національний університет радіоелектроніки
Україна, 61166, Харків, пр. Науки 14
Email: vladyslava.holovina@nure.ua

Анотація: Дана стаття спрямована на розробку системи керування мобільним пошуково-рятувальним роботом. В ході дослідження було обрано апаратні модулі, необхідні для реалізації системи керування. На базі обраних модулів було розроблено та описано передаточні функції за допомогою теорії автоматичного керування, що дозволило спроектувати структурну схему керування для перевірки запропонованої системи на стійкість.

Ключові слова: мобільний робот, системи керування, теорії автоматичного керування.

DEVELOPMENT OF MOBILE SEARCH AND RESCUE ROBOT MANAGEMENT SYSTEM

V. Holovina

Kharkiv National University of Radio Electronics
Ukraine, 61166, Kharkiv, Nauky av.,14
Email: vladyslava.holovina@nure.ua

Annotations: This article is aimed at the development of a control system for a mobile search and rescue robot. In the course of the study, the hardware modules necessary for the implementation of the control system were selected. On the basis of the selected modules, transfer functions were developed and described using the theory of automatic control, which made it possible to design a structural control diagram to check the stability of the proposed system.

Key words: mobile robot, control systems, theories of automatic control.

У контексті військової агресії Російської Федерації на Україну актуальність розроблення та використання мобільних пошуково-рятувальних роботів стає надзвичайно важливою. Ці технології можуть врятувати життя, забезпечуючи ефективне виявлення та рятування людей, особливо в небезпечних умовах воєнного конфлікту та під уламками. Використання роботів у ролі засобів моніторингу та розвідки дозволяє отримувати важливу інформацію про розташування противника та інші небезпечні об'єкти. Також вони можуть бути використані для розмінування та виявлення вибухонебезпечних об'єктів, що зменшує ризик для життя людей. Мобільні роботи стають важливим елементом комунікації та розвідки в умовах військових дій, забезпечуючи ефективність та оперативність управління військами та рятувальними операціями [1-4].

Але розробка систем керування мобільними пошуково-рятувальними роботами стикається з великою складністю через ряд технічних та стратегічних факторів. По-перше, інтеграція різноманітних сенсорів, візуальних систем та алгоритмів штучного інтелекту потребує глибоких знань в області комп'ютерного зору та обробки даних. Другий виклик полягає у розробці ефективних стратегій навігації та прийняття рішень в умовах невизначеності та змінного оточення. Крім того, безпека робота в умовах воєнного конфлікту та взаємодія з оператором потребують ретельної уваги до аспектів безпеки та ергономіки. Надійне енергозабезпечення та можливість працювати в обмежених умовах є ще однією важливою технічною трудністю [5-7].

В наслідок чого, на першому етапі розробки системи керування мобільним пошуково-рятувальним роботом, необхідно обрати апаратні модулі. В рамках розробки експериментального зразка мобільного робота, пропонується обрати наступні апаратні модулі:

- система управління на базі ESP32-CAM. Поеднує в собі мікроконтролер ESP32 та камеру, що забезпечує компактну інтеграцію для віддаленого керування та отримання відеопотоку через Wi-Fi [8].

- двигуни постійного струму DC 5В. Використання DC-моторів робить систему ефективною та енергоефективною для мобільного робота. DC-мотори також добре підходять для різних завдань, від керування шасі до маневрових операцій [9].

- драйвер двигуна L298N. Це потужний драйвер двигуна, який забезпечує можливість керування двигунами постійного струму в обидва напрямки, що важливо для точного та динамічного керування рухом робота [10].

- система живлення на базі акумуляторів 18650. Акумулятори 18650 є компактними та мають велику ємність, що робить їх ідеальним вибором для живлення мобільного робота. Забезпечуючи довгий час автономної роботи, вони сприяють ефективності та маневреності робота.

- управління через Wi-Fi з оператором. Використання Wi-Fi дозволяє віддалене управління роботом оператором через мережу, що робить його більш гнучким та дозволяє взаємодіяти в реальному часі з роботом, навіть на відстані [11].

- ультразвуковий сенсор HC-SR04. Може бути використаний для визначення відстані до об'єктів, що дозволяє роботом уникати перешкод та реагувати на зміни в навколишньому середовищі, покращуючи його навігаційні можливості [12-13].

Загальний вид обраних апаратних модулів представлено на рисунку 1.



а) ESP32-CAM; б) Hobby-370 DC 12 V 6600 RPM;
в) Драйвер двигуна L298N; г) Ультразвуковий сенсор HC-SR04

Рисунок 1 – Загальний вид обраних апаратних модулів

Але перед розробкою структурної схеми та схеми підключення, необхідно провести моделювання системи керування мобільним пошуково-рятувальним роботом з використанням теорій автоматичного керування, виконує кілька ключових функцій. По-перше, воно дозволяє аналізувати та оптимізувати взаємодію робота з оточуючим середовищем, враховуючи різні сценарії дій. Моделювання також допомагає визначити оптимальні параметри контролерів та алгоритмів для досягнення поставлених завдань, забезпечуючи ефективність та точність управління. Крім того, цей підхід дозволяє виявляти можливі ризики та недоліки в системі керування, допомагаючи вдосконалити її перед впровадженням у реальних умовах. Використовуючи теорію автоматичного керування, опишемо обрані модулі у вигляді передаточних функцій першого та другого порядку [14-16].

Представимо ультразвуковий сенсор HC-SR04 у вигляді передаточної функції першого порядку:

$$W_{oc}(s) = \frac{K_d}{1 + T_d s} \quad (1)$$

де: K_d – коефіцієнт; T_d – постійна часу.

Передаточна функція камери OV2640 для ESP32-CAM може бути апроксимована, як ідеальний передавач:

$$W_{ov}(s) = 1 \quad (2)$$

Передаточна функція самого модуля ESP32-CAM для керування мобільним пошуково-рятувальним роботом можна переставити у вигляді:

$$W_{esp}(s) = \frac{K_c}{s} \quad (3)$$

де: K_c – коефіцієнт.

Для двигунів постійного струму Hobby-370 DC можна представити у вигляді передаточної функції першого порядку, яка має наступний вигляд:

$$W_{dc}(s) = \frac{K_p}{1 + T_p s} \quad (4)$$

де: K_p – коефіцієнт; T_p – постійна часу.

Передаточна функція драйвера двигуна L298N може бути апроксимована, як ідеальний передавальний елемент:

$$W_l(s) = 1 \quad (5)$$

Передаточна функція модуля Wi-Fi, яка знаходиться на модулі ESP32-CAM можна представити у вигляді передатної функції першого порядку:

$$W_{wif}(s) = \frac{1}{1 + T_d s} \quad (6)$$

де: T_d – час затримки.

Передаточна функція першого порядку для живлення на базі акумуляторів 18650 можна представити у вигляді передатної функції:

$$W_{ak}(s) = \frac{1}{1 + T_a s} \quad (7)$$

де: T_a – постійна часу.

Операторський інтерфейс для передачі команд від оператора, представимо у вигляді передаточної функції ідеального елемента:

$$W_{mw}(s) = 1 \quad (8)$$

Розробимо структурну схему керування мобільним пошуково-рятувальним роботом з використанням теорій автоматичного керування на базі запропонованих передаточних функцій 1-8. Структурна схема керування представлена на рисунку 2 [17].

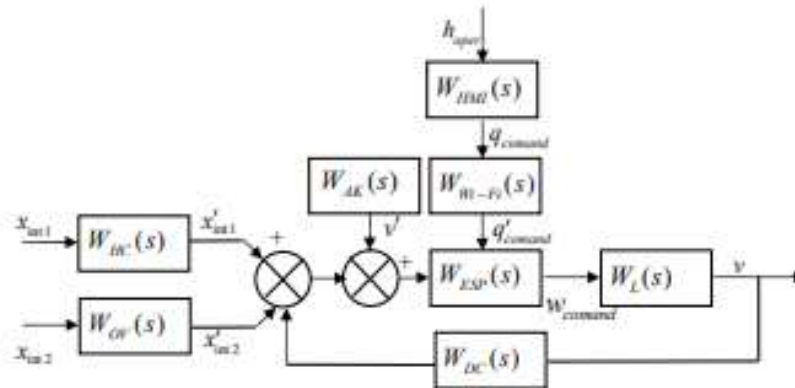


Рисунок 2 – Структурна схема керування мобільним пошуково-рятувальним роботом

На вхід схеми керування подається два сигнали x_{m1} та x_{m2} , які описують навколишнє середовище мобільного робота, оброблена інформація у вигляді x'_{m1} та x'_{m2} поступає на суматор та передається на $W_{ESP}(s)$, також на цей блок подаються команди оператора $q_{command}$ через блок $W_{m1-r}(s)$, які приймають від $q'_{command}$. Відповідно до отриманої інформації з x'_{m1} , x'_{m2} та $q'_{command}$, модуль ESP32-CAM формує керуючу команду для $W_L(s)$ у вигляді v . Цей тип сигналу надходить до $W_{DC}(s)$, що дає можливість керувати двигунами постійного струму. Також, сигнал у вигляді v' з блоку $W_{AK}(s)$ описує живлення на базі акумуляторів 18650 [18].

Висновки: В рамках проведеного дослідження було обрано апаратні модулі для розробки системи керування мобільним пошуково-рятувальним роботом. Це дало можливість, використовуючи теорію автоматичного керування описати обрані модулі у вигляді передаточних функцій першого та другого порядку. На базі яких запропонована структурна схема керування мобільним пошуково-рятувальним роботом. У майбутньому планується провести моделювання розробленої схеми управління та перевірити її на стійкість.

ЛІТЕРАТУРА

1. Bruzzone, Luca, Shahab Edin Nodehi, and Pietro Fanghella. 2022. "Tracked Locomotion Systems for Ground Mobile Robots: A Review" *Machines* 10, no. 8: 648. <https://doi.org/10.3390/machines10080648>.
2. Sasaki, T.; Fujita, T. Gap Traversing Motion via a Hexapod Tracked Mobile Robot Based on Gap Width Detection. *J. Robot. Mechatron.* 2021, 33, 665–675.
3. Yuan, Y.; Xu, Q.; Schwertfeger, S. Configuration-Space Flipper Planning on 3D Terrain. In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, Abu Dhabi, United Arab Emirates, 4–6 November 2020*; pp. 318–325.
4. Kunchev, V.; Jain, L.; Ivancevic, V.; Finn, A. Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review. In *Knowledge-Based Intelligent Information and*

Engineering Systems; Gabrys, B., Howlett, R.J., Jain, L.C., Eds.; KES 2006; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4252.

5. Attar, H., & et al. (2022). Control System Development and Implementation of a CNC Laser Engraver for Environmental Use with Remote Imaging. *Computational Intelligence and Neuroscience*, 2022, Article ID 9140156, <https://doi.org/10.1155/2022/9140156>.

6. 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>.

7. Nevliudov, I., Yevsiciev, V., Maksymova, S., Demska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In 2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) PP. 61-64. DOI: 10.1109/MEMSTECH55132.2022.10002906.

8. Євсєєв В.В. Проектування мобільних роботів на базі одноплатних комп'ютерів (Raspberry Pi) і мови Python 3.6 // Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В. Підручник. – Харків : 2020. С. 257.

9. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Новоселов С. П., Демська Н. П. Проектування мобільних маніпуляційних роботів: Монографія. – Х. : 2022. – 427 с.

10. Mykhailo Akorov, Svitlana Maksymova, & Vladyslav Yevsiciev. (2023). Choosing a Camera for 3D Mapping. *Journal of Universal Science Research*, 1(11), 28-38. Retrieved from <https://universalpublishings.com/index.php/jusr/article/view/2486>.

11. Nevliudov Igor, Maksymova Svitlana, Yevsiciev Vladyslav, & Klymenko Oleksandr. (2023). Using Mecanum Wheels for Radio Shuttle. *Multidisciplinary Journal of Science and Technology*, 3(3), 182-187.

12. Yevsiciev V. Some aspects of the development of the BEAM robot control scheme / V. Yevsiciev // In IV International Scientific and Theoretical Conference, Singapore, Republic of Singapore. - P. 79-81.

13. Yevsiciev, V. Comparative Analysis of the Characteristics of Mobile Robots and Collaboration Robots Within INDUSTRY 5.0. / V. Yevsiciev, D. Gurin // In the VI International Scientific and Theoretical Conference, September 8, 2023. Chicago, USA. P.92-94.

14. Yevsiciev, V., & Gurin, D. (2023). COMPARATIVE ANALYSIS OF THE BASIC METHODS USED IN INDUSTRY 4.0 AND INDUSTRY 5.0. *Collection of Scientific Papers «ΑΔΓΟΣ»*, (September 29, 2023; Bologna, Italy), 113-115. <https://doi.org/10.36074/logos-29.09.2023.31>.

15. Vladyslav Yevsiciev, Nikolaj Starodubcev (2023). Development of a control algorithm for a small-sized mobile manipulation robot. *Scientific Collection «InterConf»*, (140), P. 648-651.

16. Yevsiciev V. (2023) Development of a program for modeling the control of a mobile manipulation robot in the unity environment / Yevsiciev V., Starodubcev N. // *Scientific Collection «InterConf»*, (141), P. 331-334.

17. A Small-Scale Manipulation Robot a Laboratory Layout Development / Yevsiciev V., Starodubcev N., Maksymova S., Stetsenko K. // *International independent scientific journal*, №47, 2023. P.18-28.

18. Yevsiciev, V., Maksymova, S., & Starodubcev, N. (2022). A ROBOTIC PROSTHETIC A CONTROL SYSTEM AND A STRUCTURAL DIAGRAM DEVELOPMENT. *Collection of Scientific Papers «ΑΔΓΟΣ»*, (August 12, 2022; Zurich, Switzerland), 113-114. <https://doi.org/10.36074/logos-12.08.2022.33>.

ДОДАТОК Б

Основний файл прошивки

```
#include "esp_wifi.h"
#include "esp_camera.h"
#include <WiFi.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#define CAMERA_MODEL_AI_THINKER
#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 startCameraServer();    const int MotPin1 = 2; // Left front motor
const int MotPin2 = 16; // Left front motor
const int MotPin3 = 14; // Right front motor
const int MotPin4 = 15; // Right front motor
//const int MotPin5 = 1; // Left back motor
```

```

//const int MotPin6 = 3; // Left back motor
//const int MotPin7 = 16; // Right back motor
//const int MotPin8 = 0; // Right back motor
const int detectin = 4; // or 4      int detectState = 0;
void initMotors() {ledcSetup(16, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(15, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(14, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(2, 2000, 8); // 2000 hz PWM, 8-bit resolution/*
    ledcSetup(8, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(9, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(10, 2000, 8); // 2000 hz PWM, 8-bit resolution
    ledcSetup(11, 2000, 8); // 2000 hz PWM, 8-bit resolution*/// ledcAttachPin(GPIO, channel)
    ledcAttachPin(MotPin1, 3); // Right front motor
    ledcAttachPin(MotPin2, 4); // Right front motor
    ledcAttachPin(MotPin3, 5); // Left front motor
    ledcAttachPin(MotPin4, 6); // Left front motor/*
    ledcAttachPin(MotPin5, 8); // Left back motor
    ledcAttachPin(MotPin6, 9); // Left back motor
    ledcAttachPin(MotPin7, 10); // Right back motor
    ledcAttachPin(MotPin8, 11); // Right back motor*/}
void setup() {WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    pinMode(detectin, INPUT);Serial.begin(115200)// Serial.begin(256000);
    Serial.setDebugOutput(true); Serial.println(); 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;
if(psramFound()){config.frame_size = FRAMESIZE_QVGA;
config.jpeg_quality = 10;
    config.fb_count = 2;} else {config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 12; config.fb_count = 1;}
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {Serial.printf("Camera init failed with error 0x%x", err); return;}
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);
s->set_vflip(s, 1);
s->set_hmirror(s, 1); initMotors();
ledcSetup(7, 5000, 8);
ledcAttachPin(4, 7); //pin4 is LED
Serial.println("ssid: " + (String)ssid);
Serial.println("password: " + (String)password);
WiFi.begin(ssid, password); delay(500);
long int StartTime=millis();
while (WiFi.status() != WL_CONNECTED) {
    delay(500);    if ((StartTime+10000) < millis()) break;}
startCameraServer();

```

```

if (WiFi.status() == WL_CONNECTED) {Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println(""); } else { Serial.println("");
  Serial.println("WiFi disconnected");
  Serial.print("Self-hosted camera ready! Use 'http://");
  Serial.print(WiFi.softAPIP());
  Serial.println("");
  char* apssid = "ESP32-CAM";
  char* appassword = "Grandway_1";
  WiFi.softAP((WiFi.softAPIP().toString()+"_"+(String)apssid).c_str(), appassword);  }
for (int i=0;i<5;i++) { ledcWrite(7,10); // flash led
  delay(50); ledcWrite(7,0); delay(50);  }}
void loop() {delay(1000);Serial.printf("RSSI: %ld dBm\n",WiFi.RSSI());
  detectState = digitalRead(detectin);
  Serial.println(detectState);// check if the pushbutton is pressed.// if it is, the buttonState is HIGH
  if (detectState == HIGH) { // turn LED on Serial.printf("Someone detected in sight");  }}

```

ДОДАТОК В

Файл с веб-сервером

```

#include "dl_lib_matrix3d.h"
#include <esp32-hal-ledc.h>
int speed = 255;
int speedBalansR = 0; // If the car deviates to the left, then reduce the speed of the right motor.
int speedBalansL = 0; // If the car deviates to the right, then reduce the speed of the left motor.
int noStop = 0;
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "Arduino.h"
typedef struct {
    httpd_req_t *req;
    size_t len;} jpg_chunking_t;
#define PART_BOUNDARY "12345678900000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length:
%u\r\n\r\n";
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;
static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len)
{jpg_chunking_t *j = (jpg_chunking_t *)arg;
if (!index) {j->len = 0;}
if (httpd_resp_send_chunk(j->req, (const char *)data, len) != ESP_OK) {return 0;}
j->len += len;
return len;}

```

```

static esp_err_t capture_handler(httpd_req_t *req) { camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    int64_t fr_start = esp_timer_get_time();
    fb = esp_camera_fb_get();
    if (!fb) { Serial.println("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL; }
    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline; filename=capture.jpg");
    size_t out_len, out_width, out_height;
    uint8_t * out_buf;
    bool s;
    { size_t fb_len = 0;
        if (fb->format == PIXFORMAT_JPEG) { fb_len = fb->len;
            res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
        } else {jpg_chunking_t jchunk = {req, 0};
            res = frame2jpg_cb(fb, 80, jpg_encode_stream, &jchunk) ? ESP_OK : ESP_FAIL;
            httpd_resp_send_chunk(req, NULL, 0);
            fb_len = jchunk.len; }
        esp_camera_fb_return(fb);
        int64_t fr_end = esp_timer_get_time();
        Serial.printf("JPG: %uB %ums\n", (uint32_t)(fb_len), (uint32_t)((fr_end - fr_start) / 1000));
        return res; }
    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, fb->width, fb->height, 3);
    if (!image_matrix) { esp_camera_fb_return(fb);
        Serial.println("dl_matrix3du_alloc failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;}
    out_buf = image_matrix->item;
    out_len = fb->width * fb->height * 3;
    out_width = fb->width;

```

```

out_height = fb->height;
s = fmt2rgb888(fb->buf, fb->len, fb->format, out_buf);
esp_camera_fb_return(fb);
if (!s) {dl_matrix3du_free(image_matrix);
Serial.println("to rgb888 failed");
httpd_resp_send_500(req);
return ESP_FAIL;}
jpg_chunking_t jchunk = {req, 0};
s = fmt2jpg_cb(out_buf, out_len, out_width, out_height, PIXFORMAT_RGB888, 90,
jpg_encode_stream, &jchunk);
dl_matrix3du_free(image_matrix);
if (!s) {Serial.println("JPEG compression failed");
return ESP_FAIL; }
int64_t fr_end = esp_timer_get_time();
return res;}

static esp_err_t stream_handler(httpd_req_t *req) {
camera_fb_t * fb = NULL;
esp_err_t res = ESP_OK;
size_t _jpg_buf_len = 0;
uint8_t * _jpg_buf = NULL;
char * part_buf[64];
dl_matrix3du_t *image_matrix = NULL;
static int64_t last_frame = 0;
if (!last_frame) {last_frame = esp_timer_get_time();}
res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
if (res != ESP_OK) {return res;}
while (true) { fb = esp_camera_fb_get();
if (!fb) {Serial.println("Camera capture failed");
res = ESP_FAIL;} else {{ if (fb->format != PIXFORMAT_JPEG) {
bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
esp_camera_fb_return(fb);
fb = NULL;

```

```

    if (!jpeg_converted) {Serial.println("JPEG compression failed");
        res = ESP_FAIL}} else {jpg_buf_len = fb->len;
        _jpg_buf = fb->buf;}}
    if (res == ESP_OK) {size_t hlen = sprintf((char *)part_buf, 64, _STREAM_PART,
        _jpg_buf_len);
        res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);}
    if (res == ESP_OK) {res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);}
    if (res == ESP_OK) { res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
        strlen(_STREAM_BOUNDARY));}
    if (fb) {esp_camera_fb_return(fb);
        fb = NULL;
        _jpg_buf = NULL; } else if (_jpg_buf) {free(_jpg_buf);
        _jpg_buf = NULL;}
    if (res != ESP_OK) {break;}
    int64_t fr_end = esp_timer_get_time();
    int64_t frame_time = fr_end - last_frame;
    last_frame = fr_end;
    frame_time /= 1000;
    Serial.printf("MJPG: %uB %ums (%.1ffps)\n",
        (uint32_t)(_jpg_buf_len),
        (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time);
    last_frame = 0;
    return res}
enum state {fwd, rev, stp};
state actstate = stp;
static esp_err_t cmd_handler(httpd_req_t *req){char* buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};
    buf_len = httpd_req_get_url_query_len(req) + 1;
    if (buf_len > 1) {buf = (char*)malloc(buf_len);
        if (!buf) {httpd_resp_send_500(req);

```

```

return ESP_FAIL;}

if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
    if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) == ESP_OK &
        httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK) {} else{ free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL;}} else {free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL}
    free(buf); } else {httpd_resp_send_404(req);
    return ESP_FAIL;}

int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;
if (!strcmp(variable, "framesize")){Serial.println("framesize");
    if (s->pixformat == PIXFORMAT_JPEG) res = s->set_framesize(s, (framesize_t)val);}
else if (!strcmp(variable, "quality")){ Serial.println("quality");
    res = s->set_quality(s, val);}
else if (!strcmp(variable, "flash")){ledcWrite(7, val);}
else if (!strcmp(variable, "speed")){if (val > 255) val = 255;
    else if (val < 0) val = 0; speed = val;}
else if (!strcmp(variable, "nostop")){noStop = val;}
else if (!strcmp(variable, "car")) {if (val == 1) {Serial.println("Forward");
    actstate = fwd;
    //ledcWrite(ledChannel, dutyCycle)
    ledcWrite(3, speed - speedBalansL);// лівий перед вперед
    ledcWrite(4, 0); // лівий перед назад
    ledcWrite(5, 0); // правий перед назад
    ledcWrite(6, speed - speedBalansR); //правий перед вперед
// ledcWrite(8, 0);// лівий зад назад
// ledcWrite(9, speed - speedBalansR); // лівий зад вперед
// ledcWrite(12, speed - speedBalansR); // правий зад вперед

```

```

// ledcWrite(13, 0); //правий перед назад
ledcWrite(4, speed - speedBalansR); // pin 12
ledcWrite(3, 0); // pin 13
ledcWrite(5, speed - speedBalansL); // pin 14
ledcWrite(6, 0); // pin 15
delay(200);}
else if (val == 2) { Serial.println("TurnLeft");
ledcWrite(4, 0); // лівий перед вперед
ledcWrite(3, 250); // лівий перед назад
ledcWrite(6, 0); // лівий перед назад
ledcWrite(5, 250); //правий перед вперед
// ledcWrite(8, 250); // правий зад назад
// ledcWrite(9, 0); // лівий зад вперед
// ledcWrite(12, 250); // правий зад вперед
// ledcWrite(13, 0); //правий зад назад
delay(100);}else if (val == 3) {Serial.println("Stop");
actstate = stp;
ledcWrite(4, 0); // лівий перед вперед
ledcWrite(3, 0); // лівий перед назад
ledcWrite(6, 0); // правий перед назад
ledcWrite(5, 0); //правий перед вперед
// ledcWrite(8, 0); // лівий зад назад
// ledcWrite(9, 0); // лівий зад вперед
// ledcWrite(12, 0); // правий зад вперед
// ledcWrite(13, 0); //правий зад назад
}else if (val == 4) {Serial.println("TurnRight");
ledcWrite(4, 250); // лівий перед вперед
ledcWrite(3, 0); // лівий перед назад
ledcWrite(6, 250); // правий перед назад
ledcWrite(5, 0); //правий перед вперед
// ledcWrite(8, 0); // лівий зад назад

```

```

// ledcWrite(9, 250); // лівий зад вперед
// ledcWrite(12, 0); // правий зад вперед
// ledcWrite(13, 250); //правий зад назад
delay(100);
}else if (val == 5) { Serial.println("Backward");
actstate = rev;
ledcWrite(3, 0); // лівий перед вперед
ledcWrite(4, speed - speedBalansR); // лівий перед назад
ledcWrite(5, speed - speedBalansR); // правий перед назад
ledcWrite(6, 0); //правий перед вперед
// ledcWrite(8, 250); // лівий зад назад
// ledcWrite(9, 0); // лівий зад вперед
// ledcWrite(12, 0); // правий зад вперед
// ledcWrite(13, 255); //правий зад назад
delay(200);
}else if (val == 6) { Serial.println("Restart");
ESP.restart();
}else if (val == 11) { Serial.println("leftDiagUP")
ledcWrite(4, 0); // лівий перед вперед
ledcWrite(3, 0); // лівий перед назад
ledcWrite(6, 0); // лівий перед назад
ledcWrite(5, 250); //правий перед вперед
ledcWrite(8, 0); // лівий зад назад
ledcWrite(9, 250); // лівий зад вперед
ledcWrite(12, 0); // правий зад вперед
ledcWrite(13, 0); //правий зад назад
delay(200);
else if (val == 12) { Serial.println("rightDiagUP")
ledcWrite(4, 250); // лівий перед вперед
ledcWrite(3, 0); // лівий перед назад
ledcWrite(6, 0); // правий перед назад

```

```

ledcWrite(5, 0); //правий перед вперед
ledcWrite(8, 0); // лівий зад назад
ledcWrite(9, 0); // лівий зад вперед
ledcWrite(12, 250); // правий зад вперед
ledcWrite(13, 0); //правий зад назад
delay(200);
}else if (val == 9) {Serial.println("leftStraight");
ledcWrite(4, 250); // лівий перед вперед
ledcWrite(3, 0); // лівий перед назад
ledcWrite(6, 250); // правий перед назад
ledcWrite(5, 0); //правий перед вперед
ledcWrite(8, 250); // лівий зад назад
ledcWrite(9, 0); // лівий зад вперед
ledcWrite(12, 250); // правий зад вперед
ledcWrite(13, 0); //правий зад назад
delay(200)}
else if (val == 10) {Serial.println("rightStraight");
ledcWrite(4, 0); // лівий перед вперед
ledcWrite(3, 250); // лівий перед назад
ledcWrite(6, 0); // правий перед назад
ledcWrite(5, 250); //правий перед вперед
ledcWrite(8, 0); // лівий зад назад
ledcWrite(9, 250); // лівий зад вперед
ledcWrite(12, 0); // правий зад вперед
ledcWrite(13, 250); //правий зад назад
delay(200);}
else if (val == 7) {Serial.println("leftDiagDOWN");
ledcWrite(4, 0); // лівий перед вперед
ledcWrite(3, 250); // лівий перед назад
ledcWrite(6, 0); // правий перед назад
ledcWrite(5, 0); //правий перед вперед

```

```

    ledcWrite(8, 0); // лівий зад назад
    ledcWrite(9, 0); // лівий зад вперед
    ledcWrite(12, 0); // правий зад вперед
    ledcWrite(13, 250); // правий зад назад
    delay(200);}
else if (val == 8) { Serial.println("rightDiagDOWN");
    ledcWrite(4, 0); // лівий перед вперед
    ledcWrite(3, 0); // лівий перед назад
    ledcWrite(6, 250); // правий перед назад
    ledcWrite(5, 0); // правий перед вперед
    ledcWrite(8, 0); // лівий зад назад
    ledcWrite(9, 0); // лівий зад вперед
    ledcWrite(12, 0); // лівий зад вперед
    ledcWrite(13, 0); // правий зад назад
    delay(200);}
if (noStop != 1){ ledcWrite(4, 0); // лівий перед вперед
    ledcWrite(3, 0); // лівий перед назад
    ledcWrite(6, 0); // правий перед назад
    ledcWrite(5, 0); // правий перед вперед
    ledcWrite(8, 0); // лівий зад назад
    ledcWrite(9, 0); // лівий зад вперед
    ledcWrite(12, 0); // правий зад вперед
    ledcWrite(13, 0); // правий зад назад} }
else { Serial.println("variable");
    res = -1;}
if (res) { return httpd_resp_send_500(req);}
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, NULL, 0);}
static esp_err_t status_handler(httpd_req_t *req) {
    static char json_response[1024];
    sensor_t * s = esp_camera_sensor_get();

```

```

char * p = json_response;
*p++ = '{';
p += sprintf(p, "\"framesize\":%u,", s->status.framesize);
p += sprintf(p, "\"quality\":%u,", s->status.quality);
*p++ = '}';
*p++ = 0;
httpd_resp_set_type(req, "application/json");
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, json_response, strlen(json_response));}

static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Control panel</title>
<style>
body{ font-family:Arial,Helvetica,sans-serif;
    background:#ffffff;
    color:#242ba4;
    font-size:16px }
h2{ font-size:18px }
section.main{ display:flex }
#menu,section.main{ flex-direction:column }
#menu{ display:none;
    flex-wrap:nowrap;
    min-width:340px;
    background:#363636;
    padding:8px;
    border-radius:4px;
    margin-top:-10px;

```

```
margin-right:10px }
#content{display:flex;
flex-wrap:wrap;
align-items:stretch }
figure{padding:0;
margin:0;
-webkit-margin-before:0;
margin-block-start:0;
-webkit-margin-after:0;
margin-block-end:0;
-webkit-margin-start:0;
margin-inline-start:0;
-webkit-margin-end:0;
margin-inline-end:0 }
figure img{display:block;
width:100%;
height:auto;
border-radius:4px;
margin-top:8px }
@media (min-width: 800px) and (orientation:landscape){
#content{display:flex; flex-wrap:nowrap; align-items:stretch }
figure img{display:block; max-width:100%;
max-height:calc(100vh - 40px);
width:auto;
height:auto;
margin-left: auto;
margin-right: auto }
figure{padding:0; margin:0;
-webkit-margin-before:0;
margin-block-start:0;
-webkit-margin-after:0;
```

```

margin-block-end:0;
-webkit-margin-start:0;
margin-inline-start:0;
-webkit-margin-end:0;
margin-inline-end:0 }
section#buttons{ display:flex;flex-wrap:nowrap;
  justify-content:space-between;
  margin-left: auto; margin-right: auto }
#nav-toggle{ cursor:pointer;display:block }
#nav-toggle-cb{ outline:0;opacity:0; width:0;height:0 }
#nav-toggle-cb:checked+#menu{ display:flex }
.input-group{ display:flex; flex-wrap:nowrap; line-height:22px; margin:5px 0 }
.input-group>label{ display:inline-block; padding-right:10px;min-width:47% }
.input-group input,.input-group select{ flex-grow:1 }
.range-max,.range-min{ display:inline-block;padding:0 5px }
button{ display:block;margin:7px; padding:0 12px;
  border: 3px solid #242ba4;
  line-height:28px;
  cursor:pointer;
  color:#242ba4;
  background:#ffffff;
  border-radius:5px;
  font-size:16px; outline:0 }
button:hover{ background:#c7c7c7 }
button:active{ background:#9e9e9e }
button.disabled{ cursor:default; background:#a0a0a0 }
input[type=range]{ -webkit-appearance:none;
  width:100%;height:22px;background:#ffffff;cursor:pointer;margin:0 }
input[type=range]:focus{ outline:0 }
input[type=range]::-webkit-slider-runnable-track{ width:100%;height:2px;
  cursor:pointer; background:#242ba4; border-radius:0; border:0 solid #242ba4 }

```

```

input[type=range]::-webkit-slider-thumb{border:1px solid rgba(0,0,30,0);
    height:22px; width:22px;border-radius:50px;background:#242ba4;
cursor:pointer; -webkit-appearance:none;margin-top:-11.5px }
input[type=range]:focus::-webkit-slider-runnable-track{ background:#EFEFEF}
input[type=range]::-moz-range-track{ width:100%;height:2px; cursor:pointer;
    background:#242ba4;border-radius:0;border:0 solid #242ba4}
input[type=range]::-moz-range-thumb{ border:1px solid rgba(0,0,30,0);
height:22px;width:22px; border-radius:50px;background:#ff3034;cursor:pointer}
input[type=range]::-ms-track{ width:100%; height:2px;cursor:pointer;
background:0 0;border-color:transparent;color:transparent}
input[type=range]::-ms-fill-lower{background:#242ba4;border:0 solid #242ba4;
    border-radius:0}
input[type=range]::-ms-fill-upper{background:#242ba4; border:0 solid #242ba4;
    border-radius:0}
input[type=range]::-ms-thumb{border:1px solid rgba(0,0,30,0);height:22px;
    width:22px;border-radius:50px;background:#ff3034; cursor:pointer; height:2px}
input[type=range]:focus::-ms-fill-lower{background:#242ba4}
input[type=range]:focus::-ms-fill-upper{background:#242ba4}
.switch{ display:block; position:relative;line-height:22px;font-size:16px;
    height:22px}
.switch input{outline:0;opacity:0;width:0;height:0}
.slider{ width:50px;height:22px;border-radius:22px; cursor:pointer;
    background-color:grey}
.slider,.slider:before{ display:inline-block;transition:.4s}
.slider:before{ position:relative; content:"";border-radius:50%;height:16px;
    width:16px; left:4px;top:3px;background-color:#fff}
input:checked+.slider{ background-color:#ff3034}
input:checked+.slider:before{ -webkit-transform:translateX(26px);
    transform:translateX(26px)}
select{border:1px solid #363636;font-size:14px; height:22px;outline:0;
    border-radius:5px}

```

```

.image-container{position:relative; min-width:160px}
.close{position:absolute;right:5px; top:5px; background:#ff3034;width:16px;
  height:16px;border-radius:100px; color:#fff;text-align:center;line-height:18px;
  cursor:pointer}
.hidden{display:none}
</style>
</head>
<body>
  <figure>
    <div id="stream-container" class="image-container hidden">
      <div class="close" id="close-stream">×</div>
      <img id="stream" src="">
    </div>
  </figure>
  <section class="main">
    <section id="buttons">
      <table> <tr>
        <td align="center"><button id="get-still">Photo</button></td></td>
        <td align="center"><button id="restart"
onclick="fetch(document.location.origin+'/control?var=car&val=6');">Restart</button></td>
      </tr><tr>
        <td align="center"><button id="toggle-stream">Start Stream</button></td></tr><tr>
        <td align="center" colspan="3"><input type="checkbox" id="nostop" onclick="var
noStop=0;if (this.checked)
noStop=1;fetch(document.location.origin+'/control?var=nostop&val='+noStop);">No
Stop</td><td></td></tr><tr>
        <td align="center"><button id="leftDiagUP"
onclick="fetch(document.location.origin+'/control?var=car&val=7');">leftDiagUP</button></td>
        <td align="center"><button id="forward"
onclick="fetch(document.location.origin+'/control?var=car&val=1');">Forward</button></td>
        <td align="center"><button id="rightDiagUP"
onclick="fetch(document.location.origin+'/control?var=car&val=8');">rightDiagUP</button></t
d><td></td></tr></tr></table>

```

```

        <td align="center"><button id="turnleft"
onclick="fetch(document.location.origin+'/control?var=car&val=9');">leftStraight</button></td
>

        <td align="center"><button id="stop"
onclick="fetch(document.location.origin+'/control?var=car&val=3');">Stop</button></td>

        <td align="center"><button id="turnright"
onclick="fetch(document.location.origin+'/control?var=car&val=10');">rightStraight</button></
td></tr><tr>

        <td align="center"><button id="leftDiagDOWN"
onclick="fetch(document.location.origin+'/control?var=car&val=11');">leftDiagDOWN</button
></td>

        <td align="center"><button id="backward"
onclick="fetch(document.location.origin+'/control?var=car&val=5');">Backward</button></td>

        <td align="center"><button id="rightDiagDOWN"
onclick="fetch(document.location.origin+'/control?var=car&val=12');">rightDiagDOWN</butto
n></td><td></td></tr><tr>

        <td>Flash</td>

        <td align="center" colspan="2"><input type="range" id="flash" min="0" max="255"
value="0"
onchange="try { fetch(document.location.origin+'/control?var=flash&val='+this.value); } catch(e) {
}"></td></tr><tr>

        <td>Speed</td>

        <td align="center" colspan="2"><input type="range" id="speed" min="0" max="255"
value="255"
onchange="try { fetch(document.location.origin+'/control?var=speed&val='+this.value); } catch(e)
{}"></td></tr><tr>

        <td>Quality</td>

        <td align="center" colspan="2"><input type="range" id="quality" min="10" max="63"
value="10"
onchange="try { fetch(document.location.origin+'/control?var=quality&val='+this.value); } catch(e
) {}"></td></tr><tr>

        <td>Resolution</td>

        <td align="center" colspan="2"><input type="range" id="framesize" min="0" max="6"
value="5"
onchange="try { fetch(document.location.origin+'/control?var=framesize&val='+this.value); } catc
h(e) {}"></td></tr></table></section></section><script>

document.addEventListener('DOMContentLoaded', function() {

function b(B) {let C;

switch (B.type) {case 'checkbox':

```

```

    C = B.checked ? 1 : 0;
    break;
case 'range':
case 'select-one':
    C = B.value;
    break;
case 'button':
case 'submit':
    C = '1';
    break; default:return}
const D = `${c}/control?var=${B.id}&val=${C}`;
fetch(D).then(E => {console.log(`request to ${D} finished, status: ${E.status}`)})}
var c = document.location.origin;
const e = B => {B.classList.add('hidden')},
f = B => { B.classList.remove('hidden')},
g = B => {B.classList.add('disabled'), B.disabled = !0 },
h = B => {B.classList.remove('disabled'), B.disabled = !1},
i = (B, C, D) => {D = !(null != D) || D; let E;
    'checkbox' === B.type ? (E = B.checked, C = !C, B.checked = C) : (E = B.value, B.value
= C), D && E !== C ? b(B) : !D && ('aec' === B.id ? C ? e(v) : f(v) : 'agc' === B.id ? C ? (f(t),
e(s)) : (e(t), f(s)) : 'awb_gain' === B.id ? C ? f(x) : e(x) : 'face_recognize' === B.id && (C ? h(n)
: g(n))));
document.querySelectorAll('.close').forEach(B => {
    B.onclick = () => {e(B.parentNode)}
}), fetch(`${c}/status`).then(function(B) {
    return B.json()
}).then(function(B) { document.querySelectorAll('.default-action').forEach(C => { i(C,
B[C.id], !1)} )});
const j = document.getElementById('stream'),
k = document.getElementById('stream-container'),
l = document.getElementById('get-still'),
m = document.getElementById('toggle-stream'),
n = document.getElementById('face_enroll'),

```

```

o = document.getElementById('close-stream'),
p = () => { window.stop(), m.innerHTML = 'Start Stream' },
q = () => { j.src = `${c+':81'}/stream`, f(k), m.innerHTML = 'Stop Stream' };
l.onclick = () => { p(), j.src = `${c}/capture?_cb=${Date.now()}`, f(k)
}, o.onclick = () => { p(), e(k)
}, m.onclick = () => { const B = 'Stop Stream' === m.innerHTML;
  B ? p() : q()
}, n.onclick = () => { b(n)
}, document.querySelectorAll('.default-action').forEach(B => {   B.onChange = () => b(B)});
const r = document.getElementById('agc'),
  s = document.getElementById('agc_gain-group'),
  t = document.getElementById('gainceiling-group');
r.onChange = () => { b(r), r.checked ? (f(t), e(s)) : (e(t), f(s)) };
const u = document.getElementById('aec'),
  v = document.getElementById('aec_value-group');
u.onChange = () => { b(u), u.checked ? e(v) : f(v) };
const w = document.getElementById('awb_gain'),
  x = document.getElementById('wb_mode-group');
w.onChange = () => { b(w), w.checked ? f(x) : e(x) };
const y = document.getElementById('face_detect'),
  z = document.getElementById('face_recognize'),
  A = document.getElementById('framesize');
A.onChange = () => { b(A), 5 < A.value && (i(y, !1), i(z, !1))
}, y.onChange = () => { return 5 < A.value ? (alert('Please select CIF or lower resolution before
enabling this feature!'), void i(y, !1)) : void(b(y), !y.checked && (g(n), i(z, !1)))
}, z.onChange = () => { return 5 < A.value ? (alert('Please select CIF or lower resolution before
enabling this feature!'), void i(z, !1)) : void(b(z), z.checked ? (h(n), i(y, !0)) : g(n)) });
document.addEventListener('keypress', function(event) { // Keyboard control support
  switch (event.charCode) { case 119:
    fetch(document.location.origin + '/control?var=car&val=1');
    break // Forward - W
  case 115:

```

```
    fetch(document.location.origin + '/control?var=car&val=5');
    break // Backward - S
case 97:
    fetch(document.location.origin + '/control?var=car&val=2');
    break // Turnleft - A
case 100:
    fetch(document.location.origin + '/control?var=car&val=4');
    break // Turnright - D
case 112:
    fetch(document.location.origin + '/control?var=car&val=3');
    break // Stop - P
case 108:
    fetch(document.location.origin + '/control?var=flash&val=255');
    break // Flash On - L
case 107:
    fetch(document.location.origin + '/control?var=flash&val=0');
    break // Flash Off - K
case 122:
    fetch(document.location.origin + '/control?var=car&val=7');
    break // leftDiagUP - Z
case 120:
    fetch(document.location.origin + '/control?var=car&val=8');
    break // rightDiagUP - X
case 102:
    fetch(document.location.origin + '/control?var=car&val=9');
    break // leftStraight - F
case 103:
    fetch(document.location.origin + '/control?var=car&val=10');
    break // rightStraight - G
case 99:
    fetch(document.location.origin + '/control?var=car&val=11');
```

```

        break                                // leftDiagDOWN - C
    case 118:
        fetch(document.location.origin + '/control?var=car&val=12');
        break                                // RightDiagDOWN - V}});
</script>
</body>
</html>)>rawliteral"
static esp_err_t index_handler(httpd_req_t *req){httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML))}
void startCameraServer(){httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    httpd_uri_t index_uri = { .uri      = "/",
        .method  = HTTP_GET,
        .handler = index_handler,
        .user_ctx = NULL};
    httpd_uri_t status_uri = { .uri      = "/status",
        .method  = HTTP_GET,
        .handler = status_handler,
        .user_ctx = NULL};
    httpd_uri_t cmd_uri = { .uri      = "/control",
        .method  = HTTP_GET,
        .handler = cmd_handler,
        .user_ctx = NULL};
    httpd_uri_t capture_uri = { .uri      = "/capture",
        .method  = HTTP_GET,
        .handler = capture_handler,
        .user_ctx = NULL };
    httpd_uri_t stream_uri = { .uri      = "/stream",
        .method  = HTTP_GET,
        .handler = stream_handler,
        .user_ctx = NULL};
    Serial.printf("Starting web server on port: %d\n", config.server_port);

```

```
if (httpd_start(&camera_httpd, &config) == ESP_OK) {  
    httpd_register_uri_handler(camera_httpd, &index_uri);  
    httpd_register_uri_handler(camera_httpd, &cmd_uri);  
    httpd_register_uri_handler(camera_httpd, &status_uri);  
    httpd_register_uri_handler(camera_httpd, &capture_uri);}  
config.server_port += 1;  
config.ctrl_port += 1;  
Serial.printf("Starting stream server on port: '%d'\n", config.server_port);  
if (httpd_start(&stream_httpd, &config) == ESP_OK) {  
    httpd_register_uri_handler(stream_httpd, &stream_uri);} }
```

ДОДАТОК Г

Демонстраційний матеріал у вигляді презентації

