

УДК 621.865.8

DOI 10.36910/10.36910/6775-2313-5352-2026-28-10

Смолянкін О. О., ORCID ID: <https://orcid.org/0000-0001-9312-9820>Чорнобай М. О., ORCID ID: <https://orcid.org/0009-0008-4648-1774>Маркіна Л. М., ORCID ID: <https://orcid.org/0000-0003-0735-074>Сацик В. О., ORCID ID: <https://orcid.org/0000-0002-7132-3363>

Луцький національний технічний університет, м. Луцьк, Україна

РОЗРОБКА АНТРОПОМОРФНОГО МАНІПУЛЯТОРА З КЕРУВАННЯМ ВІД РУКАВИЧКИ З ДАТЧИКАМИ ЗГИНУ НА БАЗІ МІКРОКОНТРОЛЕРА ESP32

У статті описано розробку і реалізацію системи керування антропоморфним маніпулятором кисті руки, надрукованої на 3D-принтері, за допомогою рукавички з резистивними датчиками згину. Виконавчими механізмами слугують п'ять сервоприводів Tower Pro MG996R, кожен з яких відповідає за рух окремого пальця. Сигнали від датчиків рукавички Flex Sensor 4.5" зчитуються аналогово-цифровим перетворювачем мікроконтролера ESP32 і через функцію `map()` перетворюються на кути повороту сервоприводів. Описано методику калібрування датчиків, особливості апаратної реалізації на основі дільників напруги з резисторами 22 кОм, а також програмну частину реалізовано на мові C у середовищі Arduino IDE. Наведено електричні схеми підключення, зображення готового прототипу та вихідний код програми. Проведені експериментальні дослідження підтвердили, що запропонований підхід забезпечує плавне та відтворюване керування маніпулятором у реальному часі.

Ключові слова: антропоморфний маніпулятор, ESP32, датчик згину, сервопривід, Arduino IDE, калібрування датчиків, рукавичка керування, реабілітаційна робототехніка, 3D-друк, вбудовані системи.

Постановка проблеми. Сучасна медична реабілітаційна інженерія та промислова робототехніка стикаються зі спільним викликом: необхідністю створення доступних, надійних і інтуїтивно керованих протезів та маніпуляторів кисті руки. Ампутації кінцівок внаслідок травм або захворювань суттєво знижують якість життя людей, тоді як сучасні комерційні рішення залишаються надзвичайно дорогими і технологічно складними для масового застосування. Однією із причин, яка спричинила загострення цієї проблеми стало повномасштабне вторгнення Росії в Україну з 2022 року. Тисячі військових та цивільних отримали важкі поранення, що призвели до втрати верхніх кінцівок внаслідок вибухів мін, обстрілів і бойових дій загалом. За данини благодійних організацій, кількість ампутацій у країні зросла у кілька разів порівняно з довоєнним рівнем. Крім цього, Україна є однією з найбільш замінованих країн у світі, а розмінування є надзвичайно небезпечною роботою, під час якої військові зазнають тяжких поранень і втрачають кінцівки. Тому розробка доступних та функціональних протезів набуває стратегічного значення: відновлення ветеранів та постраждалих цивільних.

Одночасно з цим в промисловій автоматизації зростає попит на легкі маніпулятори, здатні копіювати рухи людської руки в телеопераційних системах. Ключовою проблемою при проектуванні таких систем є точне і синхронне відтворення рухів пальців у реальному часі із мінімальними затримками.

Доступність на сьогоднішній день мікроконтролерів з потужними АЦП, таких як ESP32, поява гнучких друкарських конструкцій та відносно дешевих датчиків згину відкривають можливість для розробки прототипів на рівні навчальних і дослідницьких проектів в межах університетів. Однак залишаються актуальними питання методики калібрування датчиків для досягнення відтворюваного діапазону вимірювань та алгоритмів перетворення аналогового сигналу в кут повороту сервоприводу.

Аналіз останніх досліджень. Напрямок розробки маніпуляторів кисті руки активно досліджується починаючи з 2000-х років. Класичним прикладом є роботизована рука InMoov [1], повністю побудована з 3D-надрукованих деталей і керована від сервоприводів. Цей проєкт встановив стандарт відкритої апаратної платформи для антропоморфних маніпуляторів.

Значний вплив у розробку рукавичок із датчиками мають роботи Simone та Kamper [2], які запропонували рукавичку для реабілітації з вбудованими тензорезисторами. Sierotowicz та співавтори [3] довели ефективність систем Glove-to-Robot для моторної реабілітації пацієнтів з інсультом.

Застосування платформи Arduino для керування сервоприводами в маніпуляторах описано в роботах Tavakoli та Sgambelluri [4], так із досвіду розробників Луцького національного технічного університету, що описаний у роботі «Автоматизована смарт-рука робота-гуманоїда на платформі Arduino» [8], які зазначають, що 8-бітний АЦП Arduino (1024 рівні) є недостатнім для плавної передачі дрібних рухів пальців. Саме це обґрунтовує перехід на ESP32 з 12-бітним АЦП (4096 рівнів), що описано в даній роботі.

Датчики Flex Sensor 4.5" фірми Spectra Symbol широко використовуються в задачах розпізнавання жестів [5]. Класична схема підключення через дільник напруги з підтягуючим резистором 10–47 кОм добре відома, проте питання точного вибору номіналу підтягуючого резистора та процедури тарування для конкретного діапазону АЦП висвітлено менш детально.

Проблема нестабільності watchdog-таймера на ESP32 при роботі з кількома ШІМ-каналами одночасно описана в технічній документації Espressif [6] і вирішується використанням функції yield() у циклі loop(), що і застосовано в даній роботі.

Метою роботи є розробка і практична реалізація системи повторення рухів пальців людської руки з використанням рукавички з датчиками згину та антропоморфного маніпулятора на базі сервоприводів, що керуються мікроконтролером ESP32.

Для досягнення вказаної мети під час розробки вирішувались наступні задачі:

1. Проектування механічної конструкції маніпулятора кисті з 5 незалежно керованими пальцями;
2. Вибір та обґрунтування мікроконтролера, сервоприводів та датчиків;
3. Розробка схем: електричної принципової та схеми підключення;
4. Проведення калібрування датчиків згину з фіксацією мінімального та максимального значень АЦП;
5. Програмна реалізація системи керування на мові C у середовищі Arduino IDE з усуненням збоїв watchdog-таймера;
6. Верифікація роботи системи та оцінка плавності й точності відтворення рухів.

Виклад основного матеріалу. Як зазначалось раніше, для реалізації поставленої мети було сформовано шість взаємопов'язаних задач, що охоплюють повний цикл розробки – від механічного проектування до експериментальної перевірки системи. Послідовне вирішення кожної з них забезпечує комплексний підхід до створення маніпулятора кисті та дозволяє отримати обґрунтовані результати на кожному етапі роботи.

Отже, спочатку була реалізована конструкція маніпулятора. Механічна частина якого є кистю руки людини в масштабі 1:1, повністю надрукованою на FDM 3D-принтері з використанням матеріалу PLA. Вона включає п'ять пальців з фаланговими шарнірами та систему натяжних ниток (тягових жил), з'єднаних зі шківками сервоприводів Tower Pro MG996R, розміщених у передпліччі. При натягу нитки відбувається згинання відповідного пальця; розгинання – пасивне, за рахунок вбудованих гумових елементів повернення [8]. Загальний вигляд розробленої руки представлено на рисунку 1.

Сервоприводи Tower Pro MG996R обрані через їх характеристики: металевий редуктор, один кульковий підшипник, крутний момент 11 кг·см при напрузі живлення 6 В та кутова стандартних 180°) з метою захисту механіки від перевантажень і уникнення надмірного натягу нитки.

Обґрунтований вибір мікроконтролера, сервоприводів та датчиків визначає як технічні можливості системи, так і практичну доцільність її реалізації. З урахуванням вказаних недоліків плати Arduino, для керування маніпулятором було обрано мікроконтролер ESP32 CH340 Type-C (модуль WROOM-32), який має наступні переваги: тактова частота 240 МГц (проти 16 МГц), 12-бітний АЦП (4096 рівнів проти 1024), вбудований WiFi/Bluetooth для перспективного розширення функціоналу та можливість одночасного формування кількох незалежних ШІМ-сигналів.

П'ять сервоприводів підключено до виходів ШІМ ESP32:

- великий палець: GPIO 18;
- вказівний: GPIO 19;
- середній: GPIO 21;
- безіменний: GPIO 22;
- мізинець: GPIO 23.

П'ять датчиків згину Flex Sensor 4.5" підключено до входів АЦП:

- великий палець: GPIO 32;
- вказівний: GPIO 33;
- середній: GPIO 34;
- безіменний: GPIO 35;
- мізинець: GPIO 36.

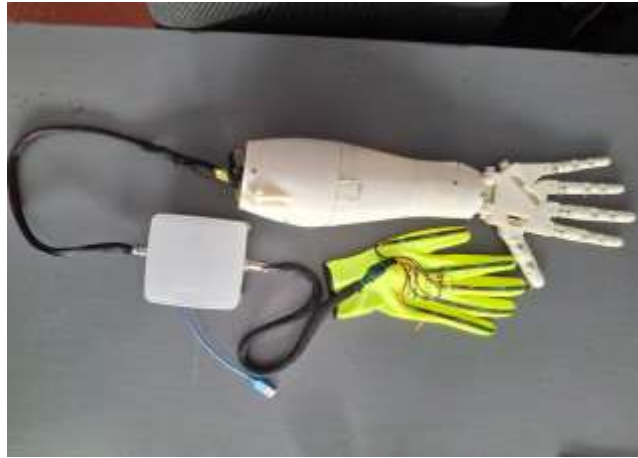


Рисунок 1 – Загальний вигляд розробленої системи: маніпулятор (ліворуч), керуюча рукавичка з датчиками згину (праворуч), блок електроніки ESP32 (центр).

Кожен датчик увімкнено у схему дільника напруги з резистором 22 кОм, підключеним між виходом датчика та землею. При такому включенні вхідна напруга АЦП лінійно залежить від опору датчика, що змінюється у діапазоні 10-22 кОм при згині від 0° до 180°. Живлення логічної частини ESP32 здійснюється через USB Type-C; силова шина сервоприводів підключена до окремого зовнішнього стабілізованого джерела живлення 6 В для запобігання просадкам напруги під час синхронного руху всіх пальців.

Вся електроніка (ESP32, плата дільників напруги, з'єднувальні кабелі) розміщена в готовому промисловому пластиковому корпусі, що забезпечує захист від механічних пошкоджень і надає завершеного вигляду пристрою (рис. 2).



Рисунок 2 – Внутрішній вигляд корпусу з ESP32 та платою дільників напруги

Точно розроблена електрична схема є умовою стабільної та безвідмовної роботи системи, а в нашому випадку руки-маніпулятора в цілому. Електрична принципова схема системи розроблена у середовищі EasyEDA і включає такі компоненти: джерело живлення USB (U2 USB-PWR), мікроконтролер ESP32-CH340C-TYPE-C (U1), п'ять сервоприводів (M1-M5 Motor_Servo), п'ять датчиків згину (U3-U7 Flex Sensor) і п'ять підтягуючих резисторів 22 кОм (R1-R5) та представлено на рисунку 3.

Калібрування датчиків згину є обов'язковим етапом нашої розробки, оскільки кожен датчик Flex Sensor має власний розкид початкового опору та нелінійні характеристики. Метою

калібрування є визначити мінімальне і максимальне значення коду АЦП для кожного пальця в стані повного розгинання і повного згинання.

Методика калібрування полягає в наступному: у налагоджувальний варіант програми виводяться сирі (початкові) значення АЦП через Serial Monitor (115200 бод). Рукавичку надягають і поперемінно фіксують кожен палець у двох крайніх положеннях – повністю випрямлений та максимально зігнутий. Зафіксовані значення АЦП записуються як мінімальний та максимальний рівень для відповідного датчика.

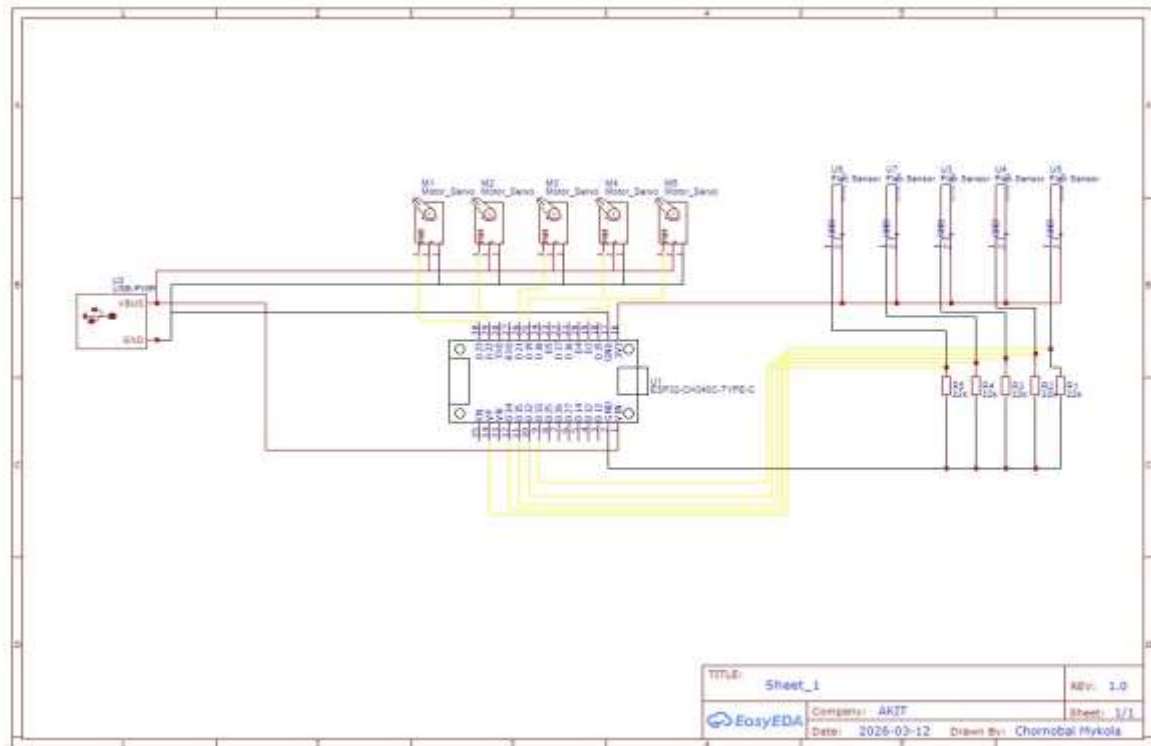


Рисунок 3 – Принципова електрична схема системи керування

За результатами проведеного калібрування було отримано такі діапазони АЦП (мін / макс):

- мізинець: 1000 / 1845;
- великий палець: 1600 / 2611;
- вказівний: 1400 / 2559;
- середній: 1600 / 2302;
- безіменний: 1500 / 2411.

Отримані значення безпосередньо використовуються у функції `constrain()` для обмеження діапазону та у функції `map()` для лінійного перетворення коду АЦП в кут сервоприводу (0-95°).

Програмна реалізація системи керування є головним елементом, що об'єднує апаратну частину в єдиний комплекс. Програма написана мовою C у середовищі Arduino IDE з використанням бібліотеки `ESP32Servo`. Структура програми складається із двох частин: `setup()` і `loop()`.

У функції `setup()` виконується ініціалізація послідовного порту (115200 бод), прив'язка об'єктів `Servo` до відповідних GPIO-виводів та встановлення початкового положення сервоприводів у 0° (пальці розігнуті). Затримка 1000 мс після ініціалізації дозволяє сервоприводам вийти у нульове положення до початку зчитування датчиків.

У функції `loop()` реалізовано наступний алгоритм: зчитування аналогових значень з п'яти каналів АЦП; обмеження значення у каліброваному діапазоні за допомогою `constrain()`; лінійне відображення в діапазон 0-95° за допомогою `map()` з інвертованими аргументами (від максимуму до мінімуму), що забезпечує відповідність: згин пальця рукавички і збільшення кута сервоприводу; виведення кутів на відповідні сервоприводи; мінімальна затримка 1 мс та виклик `yield()` для передачі керування операційній системі FreeRTOS ESP32 і запобігання спрацюванню watchdog-таймера.

Ключовою особливістю є використання `yield()` замість `delay()`. На відміну від Arduino, де `loop()` виконується монополюсно, ESP32 на базі FreeRTOS потребує явної передачі керування системним задачам. Без `yield()` або достатнього `delay()` watchdog-таймер через ~5 с примусово перезавантажує мікроконтролер.

Вихідний код програми (мова C, Arduino IDE) представлено на лістингу 1.

Лістинг 1 – Програмна реалізація керування маніпулятором (Arduino IDE / C)

```
#include <ESP32Servo.h>
Servo servoBig;
Servo servoIndex;
Servo servoMiddle;
Servo servoRing;
Servo servoPinky;
const int PIN_BIG=18, PIN_INDEX=19, PIN_MIDDLE=21, PIN_RING=22, PIN_PINKY=23;
const int FLEX_PINKY=36, FLEX_BIG=32, FLEX_INDEX=33, FLEX_MIDDLE=34,
FLEX_RING=35;
void setup()
{
  Serial.begin(115200);
  servoBig.attach(PIN_BIG);
  servoIndex.attach(PIN_INDEX);
  servoMiddle.attach(PIN_MIDDLE);
  servoRing.attach(PIN_RING);
  servoPinky.attach(PIN_PINKY);
  servoBig.write(0);
  servoIndex.write(0);
  servoMiddle.write(0);
  servoRing.write(0);
  servoPinky.write(0);
  delay(1000);
}
void loop() {
  int vP=analogRead(FLEX_PINKY);
  vB=analogRead(FLEX_BIG);
  vI=analogRead(FLEX_INDEX);
  vM=analogRead(FLEX_MIDDLE);
  vR=analogRead(FLEX_RING);
  int posP = map(constrain(vP,1000,1845), 1845,1000, 0,95);
  int posB = map(constrain(vB,1600,2611), 2611,1600, 0,95);
  int posI = map(constrain(vI,1400,2559), 2559,1400, 0,95);
  int posM = map(constrain(vM,1600,2302), 2302,1600, 0,95);
  int posR = map(constrain(vR,1500,2411), 2411,1500, 0,95);
  servoBig.write(posB);
  servoIndex.write(posI);
  servoMiddle.write(posM);
  servoRing.write(posR);
  servoPinky.write(posP);
  delay(1);
  yield();}
```

кінець лістингу 1

Зворотне відображення у `map()` (від максимального до мінімального значення АЦП як першого аргументу) є свідомим дизайнерським рішенням: при фізичному згинанні пальця опір датчика зростає, а значення АЦП збільшується – але ми хочемо, щоб кут сервоприводу також

збільшувався. Інверсія аргументів `map()` вирішує цю задачу без додаткових математичних перетворень.

Висновки. В результаті проведеної роботи реалізовано функціональний прототип системи дистанційного повторення рухів пальців людської руки. Основні результати і висновки по завершенню проведеного дослідження наступні:

1. Запропонована архітектура «рукавичка-ESP32-маніпулятор» забезпечує плавне й інтуїтивне керування кожним пальцем незалежно з затримкою не більше 1-2 мс у циклі опитування.

2. Перехід з платформи Arduino на ESP32 дозволив підвищити роздільну здатність АЦП у 4 рази (з 1024 до 4096 рівнів), що безпосередньо позначилось на плавності руху пальців маніпулятора.

3. Запропонована методика калібрування датчиків згину дозволила індивідуально визначити діапазон АЦП для кожного пальця і застосувати його в функції `map()` без складних алгоритмів.

4. Схема дільника напруги з резисторами по 22 кОм підібрані відповідно до максимального опору датчика Flex Sensor 4.5", дозволила отримати стабільний сигнал без перешкод

5. Виклик функції `yield()` у циклі `loop()` є обов'язковим для ESP32 при безперервному опитуванні АЦП – він запобігає скиданню мікроконтролера watchdog-таймером FreeRTOS.

6. Розроблену систему може бути використано як платформу для подальших досліджень у сфері реабілітаційних протезів для верхніх кінцівок ветеранів, цивільного населення, телеоперування, а також як навчальний стенд у галузі робототехніки та мехатроніки.

Перспективами подальшого розвитку є додавання бездротового з'єднання (Bluetooth/WiFi) для збільшення відстані між рукавичкою і маніпулятором, реалізація зворотного зв'язку за силою захоплення та розширення системи до повноцінного протезу з кріпленням до передпліччя.

Інформаційні джерела

1. Gael Langevin. InMoov – Open Source 3D printed life size robot. URL: <https://inmoov.fr> (дата звернення: 01.03.2026).

2. Simone L. K., Kamper D. G. Design considerations for a wearable monitor to measure finger posture. *Journal of NeuroEngineering and Rehabilitation*. 2005. Vol. 2, No. 5. P. 1-10.

3. Sierotowicz M., Lotti N., Carpaneto L., Molteni F., Guanziroli E., Muller-Putz G. et al. EMG-Driven Machine Learning-Based Control of Finger Stiffness. *IEEE Trans. Neural Syst. Rehabil. Eng.* 2019. Vol. 27, No. 5. P. 1038-1046.

4. Tavakoli M., Sgambelluri N., Niccolini M., Mazzoleni S., Filippeschi A., Vitiello N. Adaptive gesture recognition with variation estimation for robotic hand prostheses. *Robotics and Autonomous Systems*. 2017. Vol. 95. P. 52-64.

5. Spectra Symbol. Flex Sensor 4.5 inch – Datasheet URL: <https://www.Spectrasymbol.com/flex-sensor> (дата звернення: 01.03.2026).

6. Espressif Systems. ESP32 Technical Reference Manual. Ver. 5.3. URL: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (дата звернення: 01.03.2026).

7. УНІАН. З початку широкомасштабної війни від 20 до 50 тисяч українців пережили ампутацію – WSJ. URL: <https://www.unian.ua/war/viyna-v-ukrajini-vid-20-do-50-tisyach-ukrajinciv-perezihili-amputaciyu-wsj-12348438.html> (дата звернення: 10.03.2026).

8. Oleksandr Povstanoi, Lyudmila Markina, Viktor Denysiuk, Yurii Lapchenko, Viktor Satsyk, Mykhailo Pylypets. Automated smart humanoid robot arm on the ARDUINO platform / *Technologia i Automatyzacja Montażu*. V. 130, Issue 4 (2025), Pages 43-53. URL: <https://doi.org/10.7862/tiam.2025.4.4>

УДК 621.865.8

Smoliankin O. , Markina L. , Chornobai M. , Satsyk V.

Lutsk National Technical University

DEVELOPMENT OF AN ANTHROPOMORPHIC MANIPULATOR CONTROLLED BY A GLOVE WITH FLEX SENSORS BASED ON THE ESP32 MICROCONTROLLER.

The article describes the development and implementation of a control system for a 3D-printed anthropomorphic hand manipulator, controlled via a glove equipped with resistive flex sensors. The actuators are five Tower Pro MG996R servo motors, each responsible for the independent movement of an individual finger. Signals from the Flex Sensor 4.5" glove sensors are acquired by the analog-to-digital converter of the ESP32 microcontroller and converted into servo rotation angles using the map() function. The sensor calibration methodology is described, along with the hardware implementation based on voltage divider circuits with 22 k Ω resistors, and the software written in C within the Arduino IDE environment. Electrical schematics, images of the finished prototype, and the full program source code are provided. Experimental results demonstrate that the proposed approach ensures smooth and reproducible real-time control of the manipulator.

Keywords: anthropomorphic manipulator, ESP32, flex sensor, servo drive, Arduino IDE, sensor calibration, control glove, rehabilitation robotics, 3D printing, embedded systems.

Дата першого надходження
статті до видання
16.04.2026 р.

Дата прийняття статті
до друку
21.05.2026 р.

Дата
оприлюднення
30.05.2026 р.