

Міністерство освіти і науки України
Департамент освіти і науки Кіровоградської обласної військової адміністрації
Кіровоградська Мала академія наук учнівської молоді

Відділення інженерії та матеріалознавства
Секція: автоматизація та робототехніка

АВТОНОМНИЙ БОЙОВИЙ МОДУЛЬ

Роботу виконав:
Колісник Максим Олександрович,
учень 10 класу
комунального закладу
«Центральноукраїнський
науковий ліцей-інтернат
Кіровоградської обласної ради»

Наукові керівники:
Денисов Денис Олександрович,
завідувач кабінетом
Кіровоградської Малої академії
наук учнівської молоді,
вчитель фізики комунального
закладу «Центральноукраїнський
науковий ліцей-інтернат
Кіровоградської обласної ради»;

Саленко Олег Олександрович,
методист Кіровоградської Малої
академії наук учнівської молоді

Кіровоградська Мала академія наук учнівської молоді

Анотація

Колісник Максим Олександрович, учень 10 класу комунального закладу «Центральноукраїнський науковий ліцей-інтернат Кіровоградської обласної ради».

Наукові керівники: Денисов Денис Олександрович, завідувач кабінетом Кіровоградської Малої академії наук учнівської, вчитель фізики комунального закладу «Центральноукраїнський науковий ліцей-інтернат Кіровоградської обласної ради»;

Саленко Олег Олександрович, методист Кіровоградської Малої академії наук учнівської молоді

Автономний бойовий модуль

Науково-дослідницький проєкт присвячений дослідженню та розробці автономного бойового модуля як інноваційного рішення для військової техніки в умовах війни в Україні. В структурі роботи розглядається концепція та функціональні можливості автономного бойового модуля, який інтегрує передові технології мехатроніки, штучного інтелекту та систем автоматизації.

Для створення прототипу бойового модуля пропонуємо використати мінікомп'ютер Raspberry Pi 4, який можна навчити виявляти людей на відео, та систему сервоприводів, яка допоможе навести зброю на ворога.

Результатом дослідження є прототип, який наводить лазер на людину, якщо вона з'являється в полі зору камери.

Висновки науково-дослідницького проєкту вказують на значущі перспективи застосування цих технологічних рішень у сучасних умовах ведення війни та конфліктів, відкриваючи нові горизонти для стратегічного планування та підвищення бойової ефективності.

Ключові слова: автономний бойовий модуль, мехатроніка, штучний інтелект, військова техніка, безпека.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА ОСНОВНІ ПРИНЦИПИ ПОБУДОВИ АВТОНОМНИХ РОБОТИЗОВАНИХ ПЛАТФОРМ	6
1.1. Мінікомп'ютер чи мікроконтролер, порівняння можливостей.....	6
1.2. Мінікомп'ютер Raspberry Pi 4 та його можливості	8
1.3. Принципи побудови автономних роботизованих платформ.....	9
РОЗДІЛ 2 ТЕХНОЛОГІЇ РОБОТИЗОВАНИХ ПЛАТФОРМ У СУЧАСНІЙ ВІЙНІ	11
2.1. Індустрія 4.0 та автономні роботизовані платформи	11
2.2. Війна в Україні та технології майбутнього	12
2.3. Автономні бойові турелі України і світу.....	12
РОЗДІЛ 3 СТВОРЕННЯ ПРОТОТИПУ АВТОНОМНОГО БОЙОВОГО МОДУЛЯ.....	15
3.1. Деталі прототипу автономного бойового модуля	15
3.2. Програмування бойового модуля.....	16
3.3. Моделювання прототипу бойового модуля	20
3.4. Апробація прототипу автономного бойового модуля.....	21
ВИСНОВКИ.....	23
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	24
ДОДАТКИ.....	26

ВСТУП

В сучасному світі стрімкого розвитку технологій військова сфера постійно шукає нові та ефективні рішення для забезпечення високої бойової ефективності та безпеки військовослужбовців. Особливо гостро це питання постає у розрізі сучасної війни з росією. Українська армія прогресує не тільки у військовому плані, але і технологічно. Одним із інноваційних напрямків цього розвитку стає використання автономних бойових модулів – високотехнологічних систем, призначених для автоматизації та підвищення бойових можливостей. В нашій науковій роботі розглядається концепція та розробка автономного бойового модуля, який інтегрує в себе передові технології мехатроніки, штучного інтелекту та систем автоматизації. Досліджуються принципи його функціонування, архітектурні особливості та можливості використання в сучасних умовах ведення війни та конфліктів. Автономні бойові модулі визначають новий рівень військової техніки, відкриваючи перед військовими стратегами та техніками нові перспективи для ефективного використання в бойових операціях.

Мета дослідження: розробити прототип автономного бойового модуля на платформі Raspberry Pi.

Завдання дослідження:

- 1) на підставі аналізу спеціальної літератури та інтернет джерел виявити актуальність побудови роботизованих модулів та інноваційних автономних військових систем;
- 2) уточнити технології створення автономних бойових систем;
- 3) розробити прототип автономного бойового модуля та його компонентів на платформі Raspberry Pi;
- 4) створити програму керування прототипом автономного бойового модуля та його електронними компонентами;
- 5) зібрати 3Д-модель прототипу автономного бойового модуля та апробувати його.

Об'єкт дослідження: роботизовані бойові платформи.

Предмет дослідження: розробка автономного бойового модуля на платформі Raspberry Pi.

Методи дослідження:

- *теоретичні методи:* опрацювання та аналіз наукових джерел з тематики дослідження; системні відомості про роботизовані бойові платформи;
- *математичні методи:* використана математична логіка для доведення коректності програмного коду;
- *спеціальні методи:* тестування і валідація для перевірки взаємодії програмних рішень, компонентів та отримання потрібного результату.

Наукова новизна проєкта полягає в розробці повністю автономного бойового модуля.

Практичне значення. Науково-дослідницький проєкт може бути використаний військовими підприємствами для розробки реального автономного бойового модуля.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТА ОСНОВНІ ПРИНЦИПИ ПОБУДОВИ АВТОНОМНИХ РОБОТИЗОВАНИХ ПЛАТФОРМ

1.1. Мінікомп'ютер чи мікроконтролер, порівняння можливостей

З поступом технологій і розвитком сучасних інформаційних систем виникає необхідність вибору оптимального обладнання для різноманітних завдань. Однією з ключових проблем є вибір між мінікомп'ютерами та мікроконтролерами, оскільки кожен з цих пристроїв володіє своїми унікальними можливостями та характеристиками.

Мінікомп'ютер (одноплатний комп'ютер) – це компактний, потужний комп'ютер, призначений для вирішення складних завдань. Зазвичай, вони використовуються у великих корпоративних системах, наукових дослідженнях та високоточних обчисленнях. Мінікомп'ютери мають велику обчислювальну потужність, великий обсяг пам'яті та можливість працювати з великою кількістю зовнішніх пристроїв [8].

Мікроконтролер – це малий, спеціалізований обчислювальний пристрій, зазвичай вбудований в електронні системи. Його головним завданням є керування певними функціями або пристроями. Мікроконтролери зазвичай мають обмежену пам'ять та обчислювальну потужність, але вони вигідні у високоспеціалізованих задачах, таких як керування побутовою технікою, автомобільні системи та інші схожі застосування [6].

Одним із ключових критеріїв порівняння є обчислювальна потужність. Мінікомп'ютери, завдяки своїй великій потужності, здатні ефективно вирішувати складні задачі обробки даних, що вимагають великих ресурсів. З іншого боку, мікроконтролери, хоч і мають обмежену потужність, виявляються ефективними в завданнях реального часу, де важлива швидкість реакції.

Також, важливим критерієм є енергоефективність. Мікроконтролери, зазвичай, споживають менше електроенергії, що робить їх ідеальними для вбудованих систем та портативних пристроїв.

В таблиці 1 подано загальний огляд характеристик мінікомп'ютерів та мікроконтролерів. Слід враховувати, що конкретні характеристики можуть змінюватися в залежності від моделі та виробника кожного пристрою. При виборі між ними необхідно зосередитися на власних потребах та конкретних вимогах завдань.

Таблиця 1

Порівняльна характеристика мінікомп'ютера та мікроконтролера
(таблиця автора)

<i>Характеристика</i>	Мінікомп'ютер	Мікроконтролер
<i>Обчислювальна потужність</i>	Висока	Обмежена
<i>Обсяг пам'яті</i>	Великий	Обмежений
<i>Застосування</i>	Комплексні обчислення, наукові	Вбудовані системи, керування
<i>Енергоспоживання</i>	Зазвичай високе	Зазвичай низьке
<i>Швидкість реакції</i>	Залежить від конкретної моделі	Висока (в задачах реального часу)
<i>Розмір</i>	Зазвичай великий	Компактний
<i>Вартість</i>	Зазвичай висока	Зазвичай низька
<i>Система охолодження</i>	Зазвичай потрібна	Зазвичай не потрібна

Вибір між мінікомп'ютером та мікроконтролером повинен ґрунтуватися на конкретних вимогах завдань та потреб користувача. Мінікомп'ютери відмінно підходять для складних обчислень та великих інформаційних потоків, тоді як мікроконтролери є оптимальним вибором для вбудованих систем, де обмежені ресурси та енергоефективність важливі [7].

Враховуючи швидкі темпи технологічного розвитку, необхідно враховувати індивідуальні особливості кожного пристрою та вибирати той, що найкраще відповідає конкретним завданням та умовам використання.

1.2. Мінікомп'ютер Raspberry Pi 4 та його можливості

Відповідно до порівняльної характеристики мінікомп'ютерів та мікроконтролерів при створенні пристрою або проекту грає потужність. Для керування датчиками, двигунами чи світлодіодами достатньо буде мікроконтролеру, але для енергоємних завдань по роботі з відео цього не достатньо – потрібен потужний мінікомп'ютер.

Одним з потужних і надійних мінікомп'ютерів, який гарно зарекомендував себе, виступає Raspberry Pi 4 (Рис. 1.1).

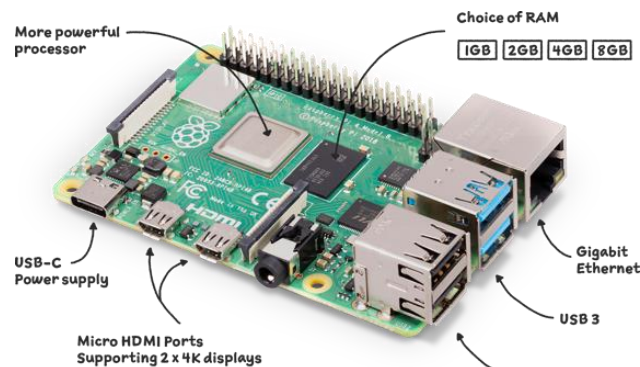


Рис. 1.1 Мінікомп'ютер Raspberry Pi 4 [14]

Raspberry Pi 4 призначений для навчання програмуванню та створення різноманітних проєктів, використовуючи як програмну, так і апаратну базу. Основні компоненти цього пристрою включають процесор, оперативну пам'ять, порти USB, HDMI для виведення відео, а також GPIO для з'єднання з різноманітними електронними компонентами [14].

Характеристики Raspberry Pi 4 [5].

Процесор: Broadcom BCM2711, чотири ядра Cortex-A72 (ARMv8) 64-біт, тактова частота до 1.5 ГГц.

Оперативна пам'ять: 2, 4 або 8 ГБ LPDDR4-3200 SDRAM (залежно від моделі).

Порти USB: 2 порти USB 3.0, 2 порти USB 2.0.

HDMI: 2 порти HDMI для виведення відео до 4K при 60 кадрах на секунду.

GPIO (General-Purpose Input/Output): 40 GPIO-пінів для підключення до різноманітних електронних компонентів та датчиків.

Ethernet: Гігабітний Ethernet (RJ45).

Бездротові технології: Wi-Fi 802.11ac, Bluetooth 5.0.

Пам'ять: Картка microSD для зберігання операційної системи та даних.

Живлення: Мікро-USB або USB-C для живлення (залежно від моделі).

Розміри: 85.6 мм x 56.5 мм x 17 мм.

Операційна система: Raspberry Pi підтримує різні операційні системи, включаючи Raspbian (тепер відому як Raspberry Pi OS), Ubuntu, Windows 10 IoT і багато інших.

Raspberry Pi підтримує різні мови програмування, включаючи Python, C++, Java та інші. Для підключення датчиків існує багато бібліотек для роботи з GPIO, що дозволяють вам керувати підключеними пристроями.

Raspberry Pi може використовуватися як домашній сервер, медіацентр, станція моніторингу, сервер для розробки та багато іншого. Плата має активну спільноту та регулярні оновлення програмного забезпечення для поліпшення функціональності та безпеки.

1.3. Принципи побудови автономних роботизованих платформ

Побудова автономних роботизованих платформ є нетривіальним завданням, що включає в себе інтеграцію різноманітних компонентів і систем для забезпечення роботи пристрою без постійного контролю з боку оператора. Нижче розглянуті основні принципи побудови автономних роботизованих платформ [10]:

1) широке використання різних сенсорів та датчиків для отримання інформації про оточуюче середовище. Це може включати камери, лідари, ультразвукові сенсори, гіроскопи, акселерометри та інші;

2) роботи повинні бути обладнані надійними системами механіки та руху. Це включає в себе мотори, сервоприводи, колеса чи гусениці, щоб забезпечити рухомість та можливість взаємодії з оточуючим середовищем;

3) роботизовані платформи потребують ефективних систем керування. Це може бути реалізовано через мікроконтролери, вбудовані системи або комп'ютерні системи, які обробляють інформацію з сенсорів і приймають рішення щодо подальших дій робота;

4) застосування алгоритмів і методів штучного інтелекту для аналізу отриманої інформації та прийняття рішень. Це може включати в себе використання машинного навчання для вдосконалення робочих функцій робота;

5) можливість взаємодії та обміну даними з іншими системами або зовнішніми джерелами є ключовою. Бездротові та провідні засоби комунікації (наприклад, Wi-Fi, Bluetooth, CAN) дозволяють роботам спілкуватися з іншими пристроями чи серверами;

6) приділяти увагу енергоефективності, особливо якщо робот повинен працювати в автономному режимі. Використання енергоефективних компонентів та оптимізація програмного забезпечення дозволяють подовжити час роботи від джерела живлення;

7) забезпечення безпеки робота та оточуючих осіб є пріоритетом. Використання датчиків для уникнення перешкод, відповідних алгоритмів безпеки та робочого дотримання стандартів безпеки;

8) передбачення можливості оновлення програмного забезпечення та обладнання для вдосконалення функціональності та довговічності робота.

Ці принципи є основою для побудови якісної роботизованої платформи або автономного бойового модуля.

РОЗДІЛ 2

ТЕХНОЛОГІЇ РОБОТИЗОВАНИХ ПЛАТФОРМ У СУЧАСНІЙ ВІЙНІ

2.1. Індустрія 4.0 та автономні роботизовані платформи

Індустрія 4.0 та автономні роботизовані платформи є тісно пов'язаними, оскільки спрямовані на використання передових технологій для оптимізації та підвищення ефективності виробничих процесів. Взаємодія між цими двома концепціями визначає майбутнє промисловості [4; 11; 15]:

- Індустрія 4.0 включає в себе впровадження автономних роботизованих платформ, які можуть автоматизувати та оптимізувати різноманітні завдання у виробництві;
- Індустрія 4.0 використовує технології штучного інтелекту та машинного навчання для надання автономним роботизованим платформам можливості навчатися від досвіду, адаптуватися до змін і оптимізувати свою діяльність з часом;
- Індустрія 4.0 враховує функціональну безпеку автономних систем та роботів: системи управління роботами можуть включати різноманітні заходи безпеки, такі як виявлення перешкод, системи автоматичної зупинки та інші;
- автономні роботизовані платформи в контексті Індустрії 4.0 розробляються з урахуванням принципів гнучкості та модульності, це дозволяє їм швидко адаптуватися до змін у виробничому середовищі та впроваджувати нові технології без значних затрат на перебудову.

При побудові автономних бойових модулів або роботизованих платформ слід враховувати не тільки їх компоновку відповідно до принципів побудови (підрозділ 1.3), а і їх взаємодію з сучасними технологіями та обладнанням, яке пропонує Індустрія 4.0.

2.2. Війна в Україні та технології майбутнього

Повномасштабна війна викликала збурення в технологічній індустрії, так само, як і в інших галузях. Згідно із дослідженням IT Research Resilience, станом на травня-червня 2022 року за кордон виїхало від 50 до 57 тис. фахівців у галузі IT. Також, приблизно 7 тис інженерів IT приєднались до складу Збройних Сил України чи Територіальної Оборони. Таким чином, на сьогоднішній день загальна кількість IT-професіоналів в Україні оцінюється приблизно в 228 тис. осіб. Це великий потенціал нашої держави до створення нових технологій, нових інноваційних захисних засобів від кривавого нападника [2].

Війна в Україні суттєво вплинула на збільшення використання військових пристроїв Internet of Things (IoT) та технологій військового зв'язку. Військові використовують IoT-пристрої, що дозволяють збирати розвідувальні дані з різних джерел, таких як датчики, камери, радіоелектронні засоби тощо. Це спрощує процес отримання та обробки інформації для прийняття важливих військових рішень. Використання IoT платформ може покращити комунікацію та координацію між різними військовими частинами. Автономні дрони, сенсори, розумна зброя та інші технології можуть підвищити ефективність та точність виконання військових завдань. Комбінація військових пристроїв IoT з кіберфізичними системами дозволяє створювати інтегровані рішення, які забезпечують сполучення фізичних та цифрових чинників для оптимізації військових операцій. І головне, розробка та використання розумних та автономних бойових машин може збільшити можливості військових управлінських структур для реагування на зміни в бойовій обстановці та зменшити втрати людей на війні [12].

2.3. Автономні бойові турелі України і світу

Компанія devDroid разом з військовими Національної гвардії України провели випробування **бойового модуля «ТГП»**, який має дистанційне

керування. На відео, яке опублікували розробники, оператор з пультом керує туреллю. Турель влучно вражає цілі на відстані до 500 метрів.

Модуль «ТГП» здатен підіймати зброю на висоту до 30 сантиметрів. Це значить, що його можна окопати та замаскувати – в потрібний момент турель просто висунеться з окопу (Рис. 2.1) [1].



Рис. 2.1 Бойовий модуль
«ТГП» від компанії devDroid
[1]

Автоматична турель «Шабля» призначена для великокаліберного кулемета і представляє собою рухому платформу, що може обладнуватися автоматичним гранатометом, кулеметом Калашникова або крупнокаліберним кулеметом (Рис. 2.2). Крім того, на цьому модулі може бути встановлено відеокамеру, тепловізор, далекомір та іншу оптику. Основне призначення цього модуля – знищення живої сили та легкої бронетехніки. Оператор може залишатися в безпеці, перебуваючи в броньованому автомобілі або дистанційно керуючи платформою. Конструкція «Шаблі» максимально спрощена та надійна, забезпечуючи швидкий ремонт та заміну [3].



Рис. 2.2 Автоматична турель
«Шабля» [3]

Бойова автоматична турель «СТРАЖ» представляє собою передовий військовий засіб, здатний автоматично виявляти та реагувати на загрози (Рис. 2.3). Турель обладнана передовими системами детекції, такими як радари, камери високої роздільної здатності, інфрачервоні сенсори та інші засоби, що дозволяють виявляти різноманітні загрози. Після виявлення потенційної загрози турель може автоматично спрямовувати свою зброю в необхідному напрямку та відкривати вогонь для негайного реагування на ситуацію. «СТРАЖ» може бути керована дистанційно, що дозволяє оператору ефективно керувати туреллю з безпечного місця та в режимі реального часу [9].

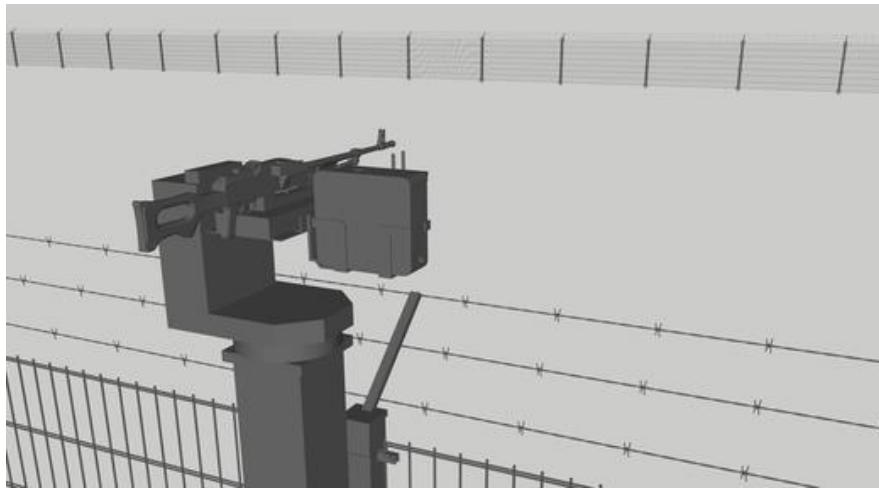


Рис. 2.3 Роботизована турель «СТРАЖ» [9]

Турель «СТРАЖ» призначена для захисту та забезпечення безпеки об'єктів в реальному часі, надаючи комплексний підхід до виявлення та усунення потенційних загроз.

Більшість турелей, про які ми знайшли інформацію дистанційні, та на сьогодні, з розвитком штучного інтелекту, потреба в операторі потроху відпадає. Машинне навчання дозволяє керувати зброєю автономно, безпечно та ефективно.

РОЗДІЛ 3

СТВОРЕННЯ ПРОТОТИПУ АВТОНОМНОГО БОЙОВОГО МОДУЛЯ

3.1. Деталі прототипу автономного бойового модуля

Бойовий модуль буде створений на основі плати керування, а саме Raspberry Pi 4 (Рис. 3.1). Для забезпечення обертання модулю навколо вісей використовуватимуться сервоприводи MG995 Tower Pro 180°. Важливою перевагою використання сервоприводів є наявність вбудованого датчика позиції, що усуває потребу у встановленні «нуля» системи координат кожен раз. Для створення відеозображення буде використана стандартна USB-камера.

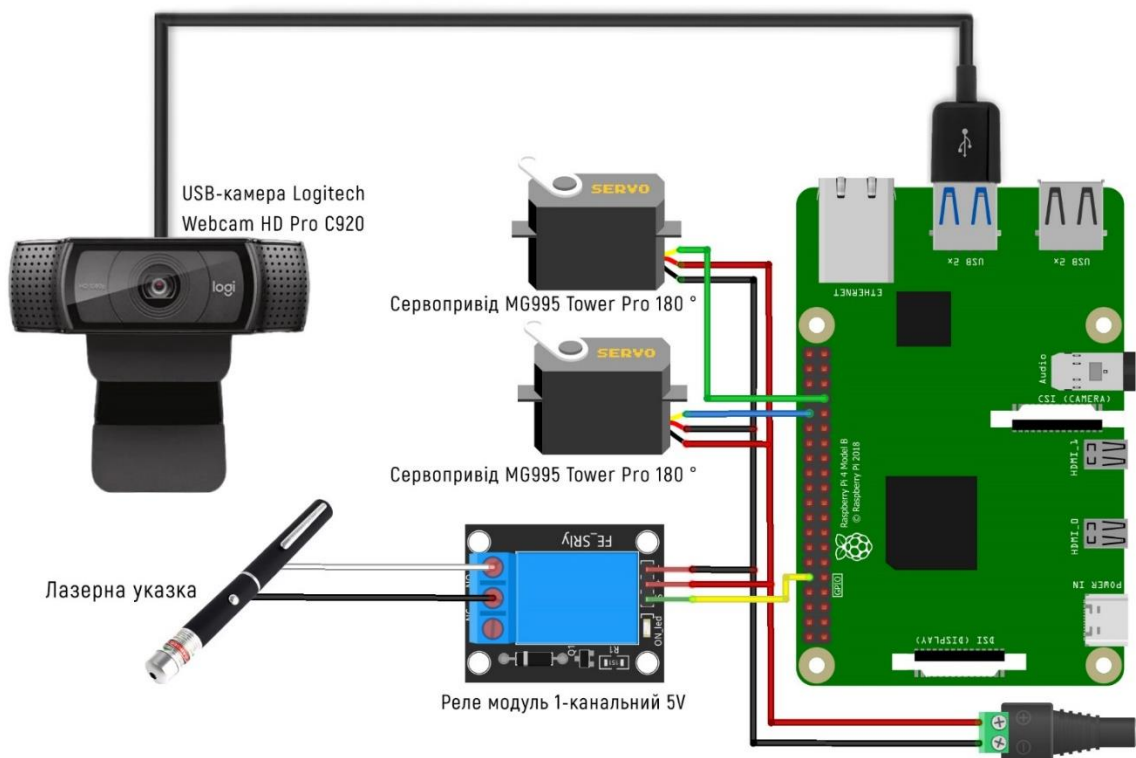


Рис. 3.1 Схема підключення електронних компонентів до Raspberry Pi 4
(схема автора)

Для прототипу використовуємо електронні компоненти відповідно до списку:

- Raspberry Pi 4 2Gb;
- сервопривід MG995 Tower Pro 180° - 2 шт.;
- реле модуль 1-канальний 5V;
- лазерна указка;
- USB-камера Logitech Webcam HD Pro C920.

Початкове збирання виконуємо на макетній платі (Рис. 3.2). Таке прототипування дозволяє легко вносити зміни в початкову конструкцію, полегшує взаємодію програми та датчиків. Первинне налаштування виконуємо для обертання сервоприводу по горизонтальній вісі. Після налаштування коду для руху по горизонтальній вісі екстраполюємо код для руху по вертикальній вісі.

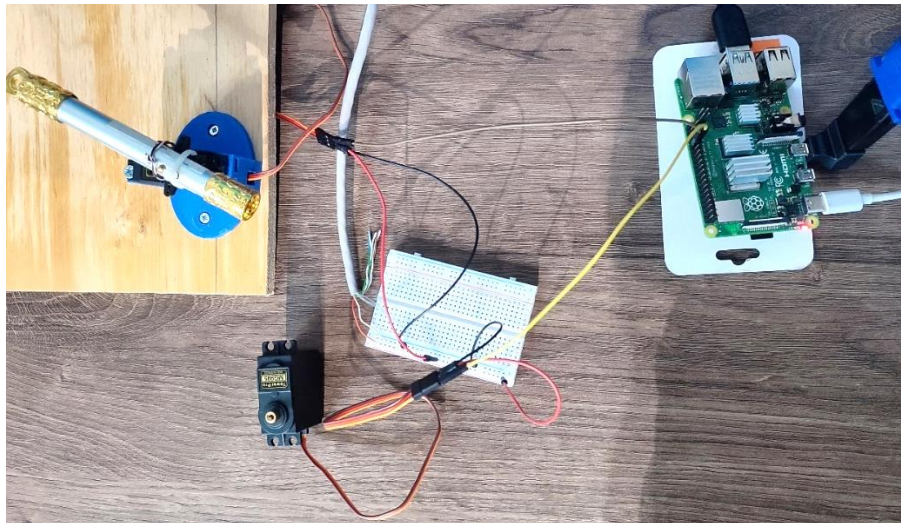


Рис. 3.2 Прототипування на макетній платі (фото автора)

3.2. Програмування бойового модуля

Для програмування будемо використовувати мову Python. Python володіє значною популярністю, як мова програмування, та вважається досить доступною для новачків. Його широке використання науковцями у їх дослідженнях пояснюється наявністю різноманітних науково-спрямованих бібліотек, які частково використовуються і нами. І хоча програмісти нарікають на швидкість роботи – у задачах, пов'язаних з машинним навчанням та штучним інтелектом, розробники вибирають саме Python. Це пов'язано з тим, що більшість бібліотек для машинного навчання написані на Python.

В цьому контексті, нижче подано короткий огляд функціоналу певних бібліотек для Python, які ми використовували у програмуванні прототипу модулю.

RPi.GPIO – це бібліотека для мови програмування Python, яка дозволяє взаємодіяти з GPIO-пінами (загальний призначення вводу/виводу) на платформі Raspberry Pi. Вона є однією з найбільш використовуваних бібліотек для роботи з GPIO на Raspberry Pi і надає зручний інтерфейс для контролю цих пінів.

Бібліотека **time** в мові програмування Python надає функціонал для роботи з часом та датами. Основні можливості цієї бібліотеки включають:

- затримка – функція **sleep(seconds)** використовується для затримки виконання програми на певну кількість секунд;
- вимірювання часу виконання коду – функції **time()** та **time_ns()** дозволяють вимірювати час виконання певного фрагменту коду;
- отримання поточного часу – функції, такі як **time()** або **perf_counter()**, дозволяють отримати поточний час у секундах або мікросекундах.

OpenCV (Open Source Computer Vision) – це відкрита бібліотека комп'ютерного зору, яка надає широкий спектр інструментів для обробки зображень та відео. OpenCV включає функції для зчитування та запису зображень, виконання операцій обробки зображень (розмір, вирізання, фільтрація), взаємодії з камерами та відеопотоками, виявлення об'єктів, визначення обличчів, аналізу руху, алгоритмів машинного навчання та багато іншого. Використовується в широкому спектрі областей, включаючи комп'ютерний зір, робототехніку, медичні дослідження та інше.

Імпортуємо визначені бібліотеки в коді програми (Рис. 3.3).

```
import cv2
import RPi.GPIO as GPIO
import time

# Инициализация камеры
cap = cv2.VideoCapture(0)

# Загрузка предварительно обученной модели Haarcascades для детекции лиц
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

Рис. 3.3 Імпортування бібліотек та підключення навченої моделі

CascadeClassifier (скріншот автора)

Для виявлення людей на зображенні будемо використовувати **CascadeClassifier** – це клас у бібліотеці OpenCV, який використовується для виявлення облич чи об’єктів на зображеннях за допомогою каскадів класифікаторів (Рис. 3.3). Каскад класифікатора – це набір класифікаторів, які обробляють зображення шляхом послідовної фільтрації, дозволяючи відкидати регіони, що не містять об’єкти, та вибирати потенційно цікаві частини зображення для подальшого аналізу. Клас CascadeClassifier в OpenCV може завантажувати попередньо навчені каскади класифікаторів, такі як класифікатор для виявлення облич чи тіла. Він може бути використаний для виявлення людей на зображеннях або відеопотоках.

Створюємо функцію **main(x)**, яка буде керувати сервоприводом по горизонтальній вісі (Рис. 3.4). Функція визначає пін під’єднання сервоприводу, частоту та генерує ШІМ-сигнал із заданою затримкою, що обертає сервопривід на потрібний кут.

```
def main(x):
    # Налаштування пінів та частоти
    OUT_PIN = 12
    PULSE_FREQ = 50

    # Налаштування роботи плати
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(OUT_PIN, GPIO.OUT)
    servo1 = GPIO.PWM(OUT_PIN, PULSE_FREQ)

    servo1.start(0)

    # Налаштування затримки шим
    servo1.ChangeDutyCycle(x/18+3)
    time.sleep(0.05)

    print(x)

    servo1.stop()
    GPIO.cleanup()
```

Рис. 3.4 Функція для керування
сервоприводом
(скріншот автора)

В головному циклі програми окрім виклику функції **main(x)** підключаємо детекцію обличчя та малюємо приціл (Рис. 3.5). Весь код програми можна для наведення прототипу автономного бойового модуля на обличчя переглянути в додатку А.

```

center_x = 0
# Малювання перехрестя у вигляді прицілу на кожному виявленому обличчі
for (x, y, w, h) in faces:
    # Знаходження центру особи
    center_x = x + w // 2
    center_y = y + h // 2

    coordinates = f"x={center_x}, y={center_y}"
    cv2.putText(frame, coordinates, org: (10, 30), cv2.FONT_HERSHEY_SIMPLEX, fontScale: 0.7, color: (255, 0, 0), thickness: 2)

    # Малювання вертикальної лінії
    cv2.line(frame, pt1: (center_x, y), pt2: (center_x, y + h), color: (0, 255, 0), thickness: 2)
    # Малювання горизонтальної лінії
    cv2.line(frame, pt1: (x, center_y), pt2: (x + w, center_y), color: (0, 255, 0), thickness: 2)
    cv2.circle(frame, center: (center_x, center_y), radius: 40, color: (0, 0, 255), thickness: 2)
    cv2.circle(frame, center: (center_x, center_y), radius: 5, color: (0, 255, 255), thickness: 2)

```

Рис. 3.5 Налаштування малювання прицілу (скріншот автора)

Результат виконання програми на мінікомп'ютері Raspberry Pi подано на Рис. 3.6. Лазер слідкує за обличчям по горизонтальній вісі та наводить на нього сервопривід.

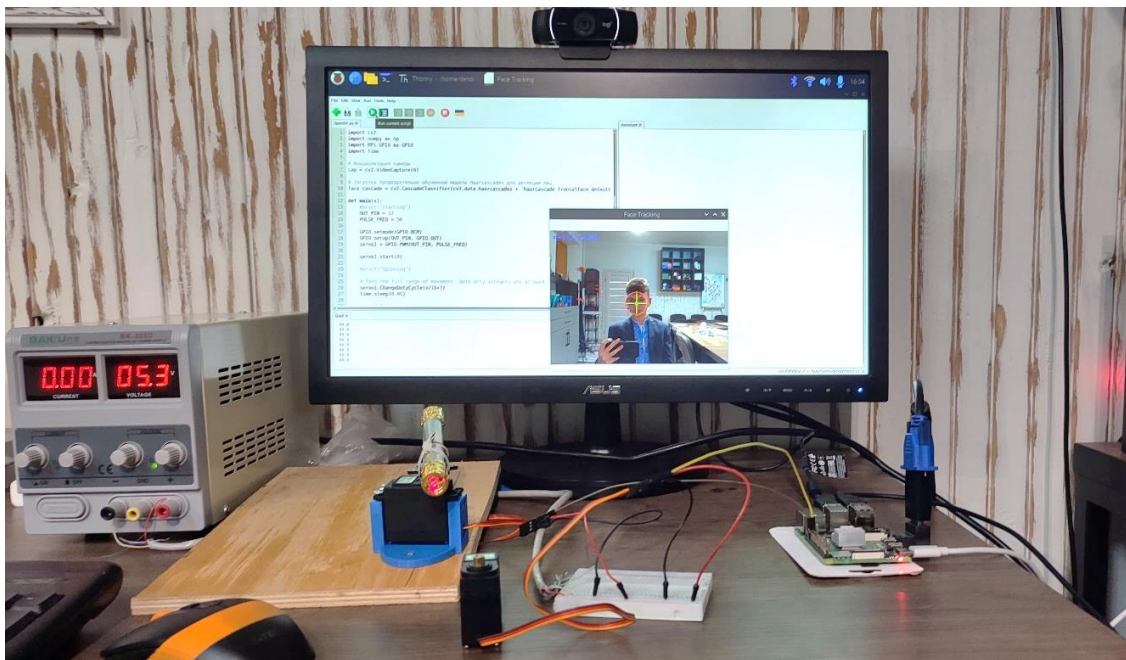


Рис. 3.6 Процес наведення лазера на обличчя (фото автора)

Екстраполюємо програмний код для вертикальної вісі, для цього створюємо функцію **main(y)** та викликаємо її у головному циклі, але виникає проблема: «Як одночасно рухати лазер по горизонталі та вертикалі?».

Макетна плата для цього не підходила, тоді було вирішено створити та надрукувати 3D-модель для двох сервоприводів і лазера.

3.3. Моделювання прототипу бойового модуля

Використовуючи відкрите джерело з безкоштовними моделями thingiverse.com було підібрано 3D-модель, яка була спроектована під розміри сервоприводів MG995 Tower Pro 180°, оберталася по двом вісям (горизонтальній і вертикальній) та мала платформу для встановлення обладнання [13].

Для кріплення лазера не вистачало платформи з отвором, цю платформу ми спроектували у безкоштовній програмі Blender. Зібравши 3D-модель в програмному середовищі Blender ми ще раз перевірили відповідність всіх кріплень, деталей та розмірів (Рис. 3.7).

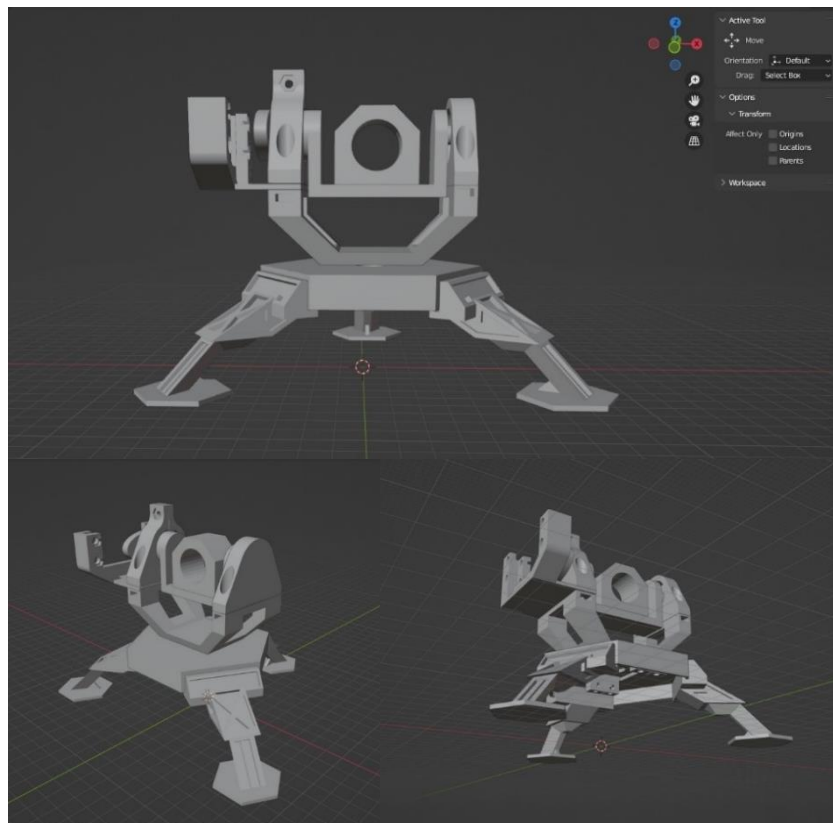


Рис. 3.7 Модель прототипу автономного бойового модуля (скріншот автора)

Друкували модель на 3D-принтері Creality Ender 3 v2 пластиком PLA.

Виконуємо збирання деталей та електроніки, лазер фіксуємо на платформі клеєм. Отримуємо такий результат (Рис. 3.8).

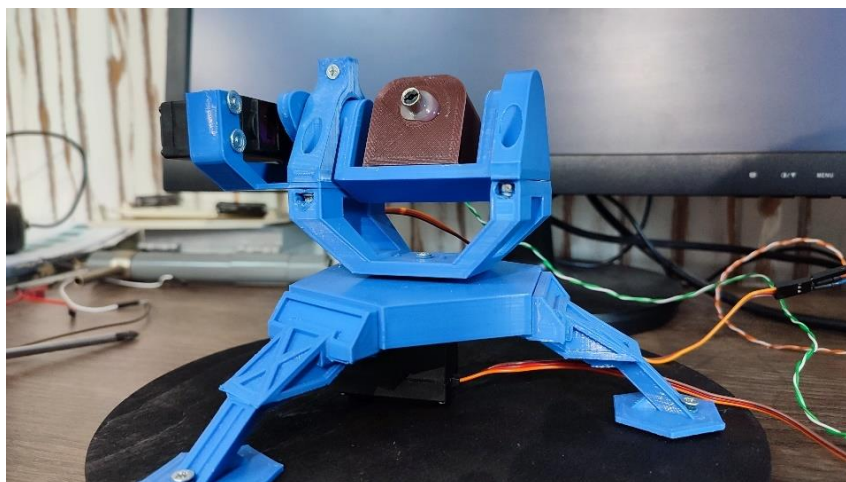


Рис. 3.8 Прототип автономного бойового модуля (фото автора)

3.4. Апробація прототипу автономного бойового модуля

Надрукований та зібраний прототип дозволив нам провести ряд дослідів для налаштування мехатроніки прототипу автономного бойового модуля та верифікації програмного коду (фото в додатку Б).

Роздрукований прототип дав можливість переключатися між моделями пошуку різних частин тіла людини: обличчя, тулубу та всього тіла. Навчені моделі класифікації можна взяти за посиланням: <https://github.com/opencv/opencv/tree/master/data/haarcascades>.

Останню версію коду ви можете переглянути за посиланням: https://github.com/dend406/combat_module.git. Тут містяться коди, які ми використовували при апробації, налаштовані на пошук обличчя, силуету людини та силуету верхньої частини тіла.

Апробація прототипу автономного бойового модуля дозволила сформулювати такі висновки:

1) бойові автономні модулі ефективні, коли використовують сучасні електронні та мехатонні компоненти, це дозволяє гнучко керувати та налаштовувати їх;

2) застосування готових алгоритмів і методів штучного інтелекту для аналізу отриманої інформації та прийняття рішень дозволяє аналізувати відеосигнал у режимі реального часу;

3) бездротові та провідні засоби комунікації дозволяють віддалено керувати модулем та спілкуватися з іншими пристроями чи серверами.

4) прототип автономного бойового модуля стає частиною Індустрії 4.0 через принципи своєї побудови та компоновки.

Верифікація коду в процесі апробації прототипу автономного бойового модуля дозволила окреслити напрямки розвитку проєкту:

1) підключення декількох моделей розпізнавання людей дозволить краще визначати людину на різних відстанях;

2) краща апаратна частина дозволила б підключити більш потужну платформу для розпізнавання людей на основі фреймворків Tensorflow або Pytorch;

3) якісніша елементна база двигунів дозволила б точніше наводити приціл на людину;

4) додаткове устаткування у вигляді платформи, що піднімається, дозволило б приховати установку за перешкодами, і виїзжати її з укриття тільки коли камера фіксувала людину.

ВИСНОВКИ

Реалізація концепції автономного бойового модуля стає важливим кроком вперед у використанні передових технологій для забезпечення безпеки українського війська та ефективності бойових підрозділів. Застосування високотехнологічних систем мехатроніки, штучного інтелекту та автоматизації дозволяє створювати адаптивні та самостійні рішення, здатні ефективно реагувати на різні виклики на полі бою та, головне, замінити людей на складних та небезпечних ділянках фронту.

Автономні бойові модулі об'єднують в собі точність, швидкість та надійність, мають потенціал змінити стратегію ведення війни та надати військовим командам нові можливості для успішного виконання завдань.

В роботі виконані наступні завдання:

1. Аналіз спеціальної літератури та інтернет джерел дозволив підтвердити актуальність побудови роботизованих платформ та інноваційних автономних військових систем;
2. Уточнено технології та принципи, які використовуються для створення автономних бойових систем;
3. Розроблено прототип автономного бойового модуля та його компонентів на платформі Raspberry Pi;
4. Створена програма керування прототипом автономного бойового модуля та його електронними компонентами дозволила перевірити взаємодію плати керування та електронних компонентів;
5. 3Д-модель автономного бойового модуля дозволила надрукувати макет прототипу на 3Д-принтері та апробувати його.

За умови подальших досліджень та вдосконалення технічних характеристик, автономні бойові модулі можуть стати ключовим елементом військової технології, впроваджуючи нові стандарти у сфері безпеки та стратегічного планування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В Україні створили «розумну» турель – що про неї відомо? [Електронний ресурс] – URL: <https://mil.co.ua/turel/>.
2. Всупереч рецесії та війні [Електронний ресурс] – URL: <https://ain.ua/special/despite-recession-and-war/>.
3. Дистанційний бойовий модуль «Шабля» [Електронний ресурс] – URL: <https://www.peoplesproject.com/bojovij-modul-shablya/>.
4. Індустрія 4.0 як інноваційний тренд України [Електронний ресурс] – URL: <https://interfax.com.ua/news/blog/799334.html>.
5. Микрокомпьютер Raspberry Pi 4 Model B 2GB [Електронний ресурс] – URL: <https://evo.net.ua/ru/mikrokomputer-raspberry-pi-4-model-b-2gb/>.
6. Мікроконтролер [Електронний ресурс] – URL: <https://uk.wikipedia.org/wiki/Мікроконтролер>.
7. Мікроконтролери проти комп'ютерів: який найкращий варіант для ваших електронних проєктів? [Електронний ресурс] – URL: <https://polaridad.es/uk/microcontroladores-vs-computadoras-cual-es-la-mejor-opcion-para-tus-proyectos-electronicos/>.
8. Одноплатний комп'ютер [Електронний ресурс] – URL: https://uk.wikipedia.org/wiki/Одноплатний_комп%27ютер.
9. Роботизована турель СТРАЖ [Електронний ресурс] – URL: <https://temerland.com/rishennya/robotizovana-turel-strazh/>.
10. Робототехнічні системи: проєктування і моделювання [Електронний ресурс]: навч. Посіб. для студ. спеціальності 126 «Інформаційні системи та технології» / М. М. Поліщук, М. М. Ткач; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 41,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
11. Світові тенденції Індустрії 4.0 на 2021 та їх вплив на Україну [Електронний ресурс] – URL: <https://industry4-0-ukraine.com.ua/2021/01/07/worldwide-trends-in-industry-4-0-and-their-impact-on-ukraine-in-2021/>.

12. Як війна вплинула на розвиток тех-індустрії [Електронний ресурс] –
URL: <https://lvivtech.com/uk/how-war-has-impacted-the-tech-industry/>.
13. Mini Turret – Working [Електронний ресурс] –
URL: <https://www.thingiverse.com/thing:3181636/files>.
14. Raspberry Pi 4 [Електронний ресурс]
URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
15. Substantial capabilities of robotics in enhancing industry 4.0 implementation
[Електронний ресурс] –
URL: <https://www.sciencedirect.com/science/article/pii/S2667241321000057>.

ДОДАТКИ

Додаток А

Програмний код для наведення прототипу автономного бойового модуля на обличчя «Combat_module_faces.py»

(актуальні версії коду для обличчя, тулубу та силуету людини)

[https://github.com/dend406/combat_module.git]

```
import cv2
import RPi.GPIO as GPIO
import time

# Ініціалізація камери
cap = cv2.VideoCapture(0)

# Завантаження попередньо навченої моделі Haarcascades для детекції облич
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

# Відключення виведення помилок
GPIO.setwarnings(False)

# Функції обробки кута в дані DutyCycle для керування сервоприводами
def main_x(x):
    OUT_PIN = 12
    PULSE_FREQ = 50
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(OUT_PIN, GPIO.OUT)
    servol = GPIO.PWM(OUT_PIN, PULSE_FREQ)
    servol.start(0)
    servol.ChangeDutyCycle(x/18+3)
    time.sleep(0.05)
    servol.stop()

def main_y(y):
    OUT_PIN = 13
    PULSE_FREQ = 50
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(OUT_PIN, GPIO.OUT)
```

```
servo1 = GPIO.PWM(OUT_PIN, PULSE_FREQ)
servo1.start(0)
servo1.ChangeDutyCycle(y/18+3)
time.sleep(0.05)
servo1.stop()

# Основний цикл програми
while True:
    # Захоплення кадру з камери
    ret, frame = cap.read()

    # Перетворення кадру у відтінки сірого
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Детекція обличч
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5,
minSize=(30, 30))

    # Початкове положення обличчя для сервоприводів
    center_x = 280
    center_y = 210

    # Малювання перехрестя у вигляді прицілу на кожному виявленому обличчі
    for (x, y, w, h) in faces:
        # Знаходження центру обличчя
        center_x = x + w // 2
        center_y = y + h // 2

        # Виведення на екран координат центра обличчя
        coordinates = f"x={center_x}, y={center_y}"
        cv2.putText(frame, coordinates, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255, 0, 0), 2)

        # Малювання вертикальної лінії
        cv2.line(frame, (center_x, y), (center_x, y + h), (0, 255, 0), 2)
        # Малювання горизонтальної лінії
        cv2.line(frame, (x, center_y), (x + w, center_y), (0, 255, 0), 2)
        cv2.circle(frame, (center_x, center_y), 40, (0, 0, 255), 2)
        cv2.circle(frame, (center_x, center_y), 5, (0, 255, 255), 2)

    # Умова спрацювання лазера при виявленні обличчя
```

```
if len(faces) == 0:
    GPIO.setmode(GPIO.BCM)
    pin_relay = 21
    GPIO.setup(pin_relay, GPIO.OUT)
    GPIO.output(pin_relay, GPIO.HIGH)
else:
    GPIO.setmode(GPIO.BCM)
    pin_relay = 21
    GPIO.setup(pin_relay, GPIO.OUT)
    GPIO.output(pin_relay, GPIO.LOW)

# переведення координат обличчя в кут для сервоприводів
ServX = (550-center_x)/6
ServY = (center_y)/4+37.5

# переведення координат обличчя в кут для сервоприводів
main_x(ServX)
main_y(ServY)

# Відображення результату на екрані
cv2.imshow('Face Tracking', frame)

# Переривання циклу при натисканні клавіші 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Звільнення ресурсів
cap.release()
cv2.destroyAllWindows()
GPIO.cleanup()
```

Процес налаштування мехатроніки прототипу автономного бойового модуля та верифікації програмного коду

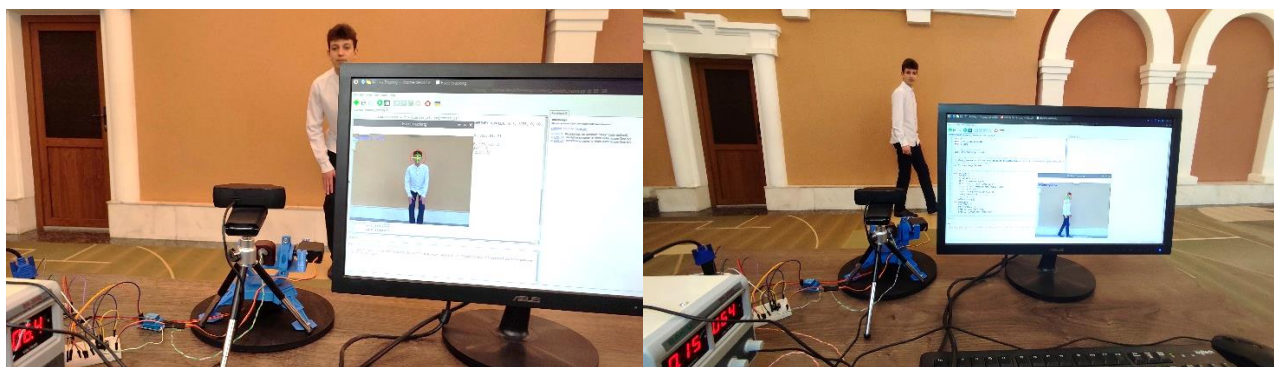


Рис. 1 Ідентифікація людини по обличчю (фото Денисова Д.О.)

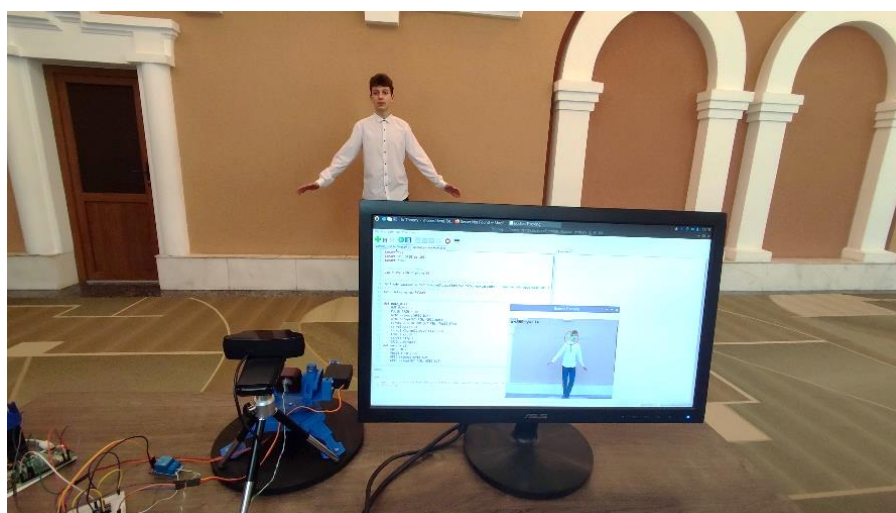


Рис. 2 Ідентифікація людини по верхній частині тіла (фото Денисова Д. О.)

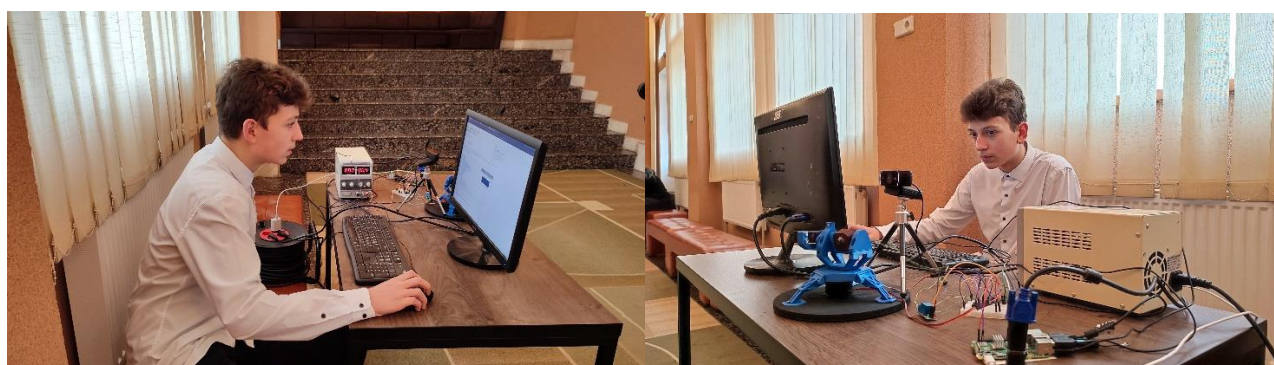


Рис. 3 Зміна налаштувань програмного коду (фото Денисова Д. О.)