УДК 004.5
DOI: 10.18523/2617-3808.2024.7.112-119

Волинець Є. А., Хмель С. М., Пєчкурова О. М.

ВІЗУАЛІЗАЦІЯ КОГНІТИВНИХ СТАНІВ НА ОСНОВІ RASPBERRY PI ДЛЯ БІОЛОГІЧНОГО ЗВОРОТНОГО ЗВ'ЯЗКУ В РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ НЕЙРО-КОМП'ЮТЕРНОГО ІНТЕРФЕЙСУ

Нейрокомп'ютерний інтерфейс на основі ЕЕГ (Electroencephalography, електроенцефалографія) — це технологія, яка дає змогу встановлювати взаємодію між мозком людини і зовнішніми пристроями, такими як комп'ютери чи робототехніка. Основні принципи такого інтерфейсу:

1. *Зчитування електричної активності мозку за допомогою електродів, розміщених на поверхні скальпа. Ці сигнали ЕЕГ відображають коливання потенціалів нейронів.*
2. *Оброблення та аналіз ЕЕГ-сигналів за допомогою алгоритмів машинного навчання, щоб визначити певні патерни, пов'язані з конкретними розумовими станами чи намірами користувача.*
3. *Переклад цих розпізнаних патернів у команди для управління зовнішніми пристроями, такими як комп'ютер, протези чи інтерфейси віртуальної реальності.*

Ключові переваги нейрокомп'ютерних інтерфейсів на основі ЕЕГ — це можливість безпосереднього керування пристроями за допомогою думок та намірів, не вдаючись до традиційних методів введення, наприклад клавіатури чи миші. Це відкриває нові перспективи для людей з обмеженими можливостями, а також для інноваційних застосувань у сферах нейрореабілітації, ігор, віртуальної реальності тощо.

У статті описано дослідження можливостей використання ЕЕГ пристрою для збору та візуалізації інформації про когнітивний стан користувача пристрою. Дослідження проводили на електроенцефалографічному приладі Emotiv INSIGHT (5-канальна система електроенцефалографії (EEG) із напівсухими полімерними датчиками), та Raspberry Pi 4B. В рамках дослідження було розроблено прототип, що дає можливість візуалізовувати показники про емоційний і ментальний стан користувача за допомогою LED, що під'єднані до Raspberry Pi.

Ключові слова: нейрокомп'ютерні інтерфейси, BCI, Raspberry Pi 4B.

Вступ

У сучасному світі стрімкого технологічного розвитку спостерігається тенденція до все тіснішого поєднання таких сфер, як штучний інтелект (AI), інтерфейси мозок-комп'ютер (BCI) та апаратне забезпечення. Ці технології мають величезний потенціал, коли використовуються у взаємодоповнювальний спосіб, відкриваючи нові горизонти для інновацій та вирішення складних проблем.

Завданням цієї статті є продемонструвати приклад інтеграції BCI та апаратних рішень на базі одноплатного комп'ютера Raspberry Pi. У статті розглянуто можливості використовувати дані про когнітивні стани людини, отримані з BCI-пристрою, для динамічного керування LED. Такий підхід може бути застосований у широкому спектрі сценаріїв, зокрема у системах «розумного будинку», освітніх і тренувальних застосунках.

Розроблення архітектури системи візуалізації когнітивних станів

Система має кілька компонентів, які працюють разом, щоб забезпечити збір, оброблення і відображення даних, зібраних з EEG-пристрою (рис. 1).

Компоненти системи: Emotiv INSIGHT, Raspberry Pi 4B, 7 LED, 7 резисторів 10 kΩ, дроти для з'єднання.

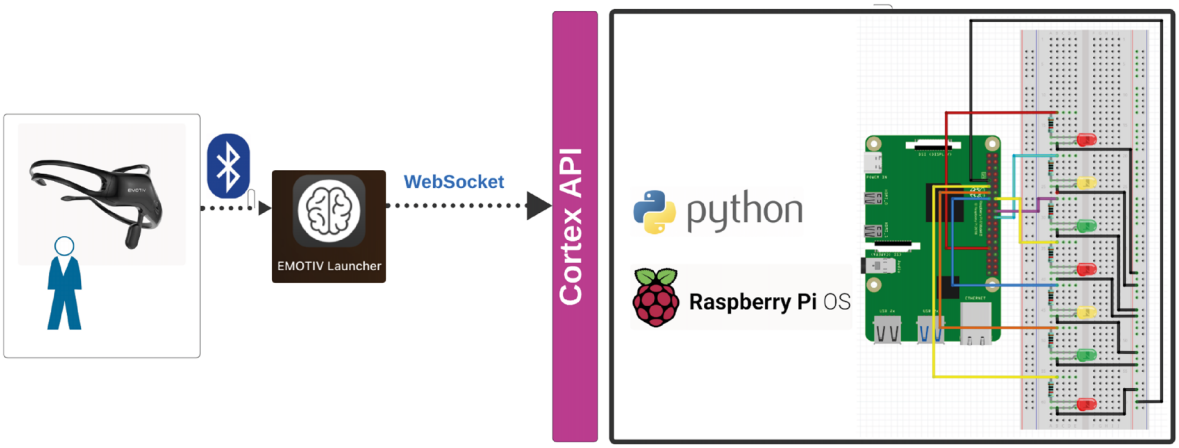

Рис. 1. Загальне архітектурне рішення

Засоби комунікації: EEG прилад з'єднується з RaspberryPi за допомогою BlueTooth, сигнали на Cortex API передаються за домомогою WebSocket.

Візуалізація сигналів відбувається за допомогою LED, що приєднані до GPIO портів RaspberryPi.

Реалізація прототипу

Для збору даних було використано бібліотеку для роботи з пристроєм Cortex, що надає виробник пристрою.

Для роботи з пристроєм має бути встановлено Launcher (програмне забезпечення для приєднання пристрою), що також надає виробник пристрою. Для реалізації прототипу було вибрано Emotiv INSIGHT, оскільки виробник Emotiv надає програмне забезпечення для приєднання пристрою до Raspberry Pi (OS – Debian version 11б 32-bit).

Попередньою умовою для реалізації проєкту є придбання ліцензії, яка дає змогу отримати доступ до електроенцефалографічних (EEG) даних і Performance Metrics — показників когнітивної діяльності.

Performance Metrics — це показники продуктивності EMOTIV, як-от увага, стрес, зацікавленість, хвилювання, розроблені за допомогою власних алгоритмів компанії-виробника, машинного навчання, ШІ та набору анонімних даних ЕЕГ. Розроблені власною командою неврологів і психологів EMOTIV, показники ефективності EMOTIV побудовані на основі даних, зібраних за допомогою встановлених парадигм нейронауки, які викликають бажаний когнітивний стан, наприклад увагу на основі наборів даних ЕЕГ від учасників контрольованих нейро наукових експериментів за допомогою гарнітур EMOTIV EPOC, Insight і MN8 [3].

Для взаємодії з гарнітурою використовують Cortex API [1]. Для розроблення проєкту попередньо створено client_id та client_secret на ресурсі для розробників EMOTIV, що є необхідною умовою.

Реалізовано робочий процес API [2] (рис. 2), створено підписку на отримання даних.

Для відображення отриманих даних за допомогою LED створено схему ланцюга та під'єднання (рис. 3).

Взаємодія з LED відбувається шляхом передавання даних на GPIO порти Raspberry Pi. GPIO (General-Purpose Input/Output) означає введення / виведення загального призначення. Ці контакти дають змогу під'єднувати електричні компоненти до Raspberry Pi і взаємодіяти з ними за допомогою програмного забезпечення.

Використано такі порти:

GPIO 17 (Physical Pin 11)
GPIO 27 (Physical Pin 13)
GPIO 22 (Physical Pin 15)
GPIO 23 (Physical Pin 16)
GPIO 24 (Physical Pin 18)
GPIO 25 (Physical Pin 22)
GPIO 6 (Physical Pin 31)

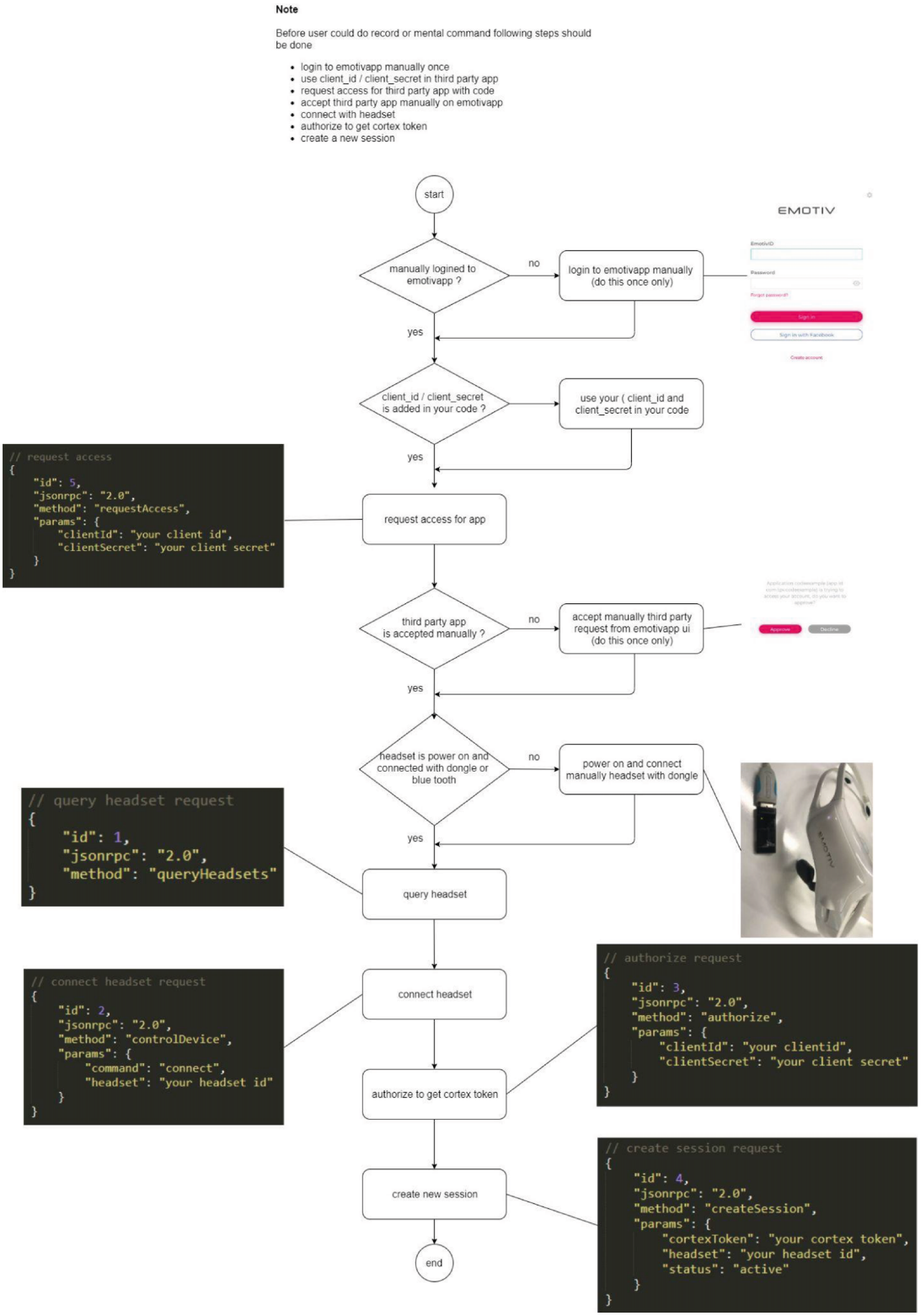

Рис. 2. Схема процесів Cortex API

Інтенсивність сигналу змінюється за допомогою використання методу led_change_duty_cycle.

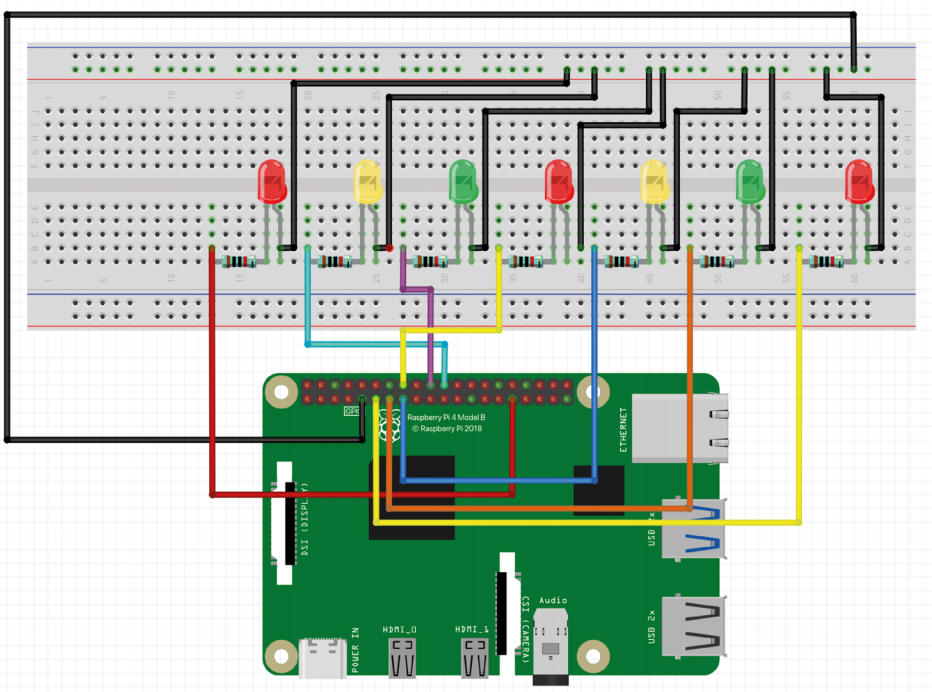

Рис. 3. Схема під'єднання ланцюга до Raspberry Pi

```
import RPi.GPIO as GPIO
import time
from cortex import Cortex
class Subcribe():
  def __init__(self, app_client_id, app_client_secret, **kwargs):
    «»»
    Constructs cortex client and bind a function to handle subscribed data streams
    If you do not want to log request and response message , set debug_mode =
False. The default is True
    «»»
    print("Subscribe __init__")
    self.c = Cortex(app_client_id, app_client_secret, debug_mode=True, **kwargs)
    self.c.bind(create_session_done=self.on_create_session_done)
    self.c.bind(new_data_labels=self.on_new_data_labels)
    self.c.bind(new_met_data=self.on_new_met_data)
    self.c.bind(inform_error=self.on_inform_error)
    # Set the GPIO mode to BCM
    GPIO.setmode(GPIO.BCM)
    # Set up the LED pins
    self.led_pins = [17, 27, 22, 23, 24, 25, 6]
    self.led_pwm = []
    # Set up each pin as a PWM output
    for pin in self.led_pins:
      GPIO.setup(pin, GPIO.OUT)
      globals()[f"pwm_{pin}"] = GPIO.PWM(pin, 100) # 100 Hz PWM
      globals()[f"pwm_{pin}"].start(0) # Start the PWM at 0% duty cycle
  def start(self, streams, headsetId=''):
    self.streams = streams
    if headsetId != '':
      self.c.set_wanted_headset(headsetId)
    self.c.open()
  def sub(self, streams):
    self.c.sub_request(streams)
  def on_new_data_labels(self, *args, **kwargs):
```

```
    data = kwargs.get('data')
    stream_name = data['streamName']
    stream_labels = data['labels']
    print('{} labels are : {}'.format(stream_name, stream_labels))
  def led_change_duty_cycle(self, pin, value):
    pwm = globals()[f"pwm_{pin}"]
    pwm.ChangeDutyCycle(value)
  def on_new_met_data(self, *args, **kwargs):
    """
    To handle performance metrics data emitted from Cortex and control LED brightness.
    """
    data = kwargs.get('data')
    met = data['met']
    print("MET data:", met)
    print("MET engagement:", met[1])
    print("MET lex:", met[4])
    print("MET ex:", met[3])
    print("MET str:", met[6])
    print("MET relax:", met[8])
    print("MET interest:", met[10])
    print("MET foc:", met[12])

    engagement = 0
    excitement = 0
    lex = 0
    stress = 0
    relax = 0
    interest = 0
    focus = 0
    try:
      engagement = met[1]
    except ValueError:
      print("'eng' not found in met data. Using default value.")
    try:
      excitement = met[3]
    except ValueError:
      print("'exc.isActive' not found in met data. Using default value.")
    try:
      lex = met[4]
    except ValueError:
      print("'lex' not found in met data. Using default value.")
    try:
      stress = met[6]
    except ValueError:
      print("'str.isActive' not found in met data. Using default value.")
    try:
      relax = met[8]
    except ValueError:
      print("'relax' not found in met data. Using default value.")

    try:
      interest = met[10]
    except ValueError:
      print("'foc.isActive' not found in met data. Using default value.")
    try:
      focus = met[12]
    except ValueError:
      print("'foc.isActive' not found in met data. Using default value.")
```

```
    # Control LED brightness based on performance metrics
    self.set_led_brightness(engagement, excitement, lex, stress, relax, inter-
est, focus)
  def set_led_brightness(self, engagement, excitement, lex, stress, relax,
interest, focus):
    «»»
    Set the brightness of the LEDs based on the performance metrics.
    «»»
    # engagement
    if isinstance(engagement, (int, float)):
      self.led_change_duty_cycle(self.led_pins[0], engagement)
    # excitement
    if isinstance(excitement, (int, float)):
      self.led_change_duty_cycle(self.led_pins[1], excitement)
    # lex
    if isinstance(lex, (int, float)):
      self.led_change_duty_cycle(self.led_pins[2], lex)
    # stress
    if isinstance(stress, (int, float)):
      self.led_change_duty_cycle(self.led_pins[3], stress)
    # relax
    if isinstance(relax, (int, float)):
      self.led_change_duty_cycle(self.led_pins[4], 100)
    # interest
    if isinstance(interest, (int, float)):
      self.led_change_duty_cycle(self.led_pins[5], 100)
    # focus
    if isinstance(focus, (int, float)):
      self.led_change_duty_cycle(self.led_pins[6], 100)
  def on_create_session_done(self, *args, **kwargs):
    print('on_create_session_done')
    # subribe data
    self.sub(self.streams)
  def on_inform_error(self, *args, **kwargs):
    error_data = kwargs.get('error_data')
    print(error_data)
def main():
  your_app_client_id = 'YOUR_CLIENT_ID'
  your_app_client_secret = 'YOUR_CLIENT_SECRET'
  s = Subcribe(your_app_client_id, your_app_client_secret)
  streams = ['met']
  s.start(streams)
  try:
    while True:
      time.sleep(1)
  except KeyboardInterrupt:
    pass
  finally:
    for pin in led_pins:
      pwm = globals()[f"pwm_{pin}"]
      pwm.stop()
    GPIO.cleanup() # Ensure GPIO cleanup occurs
if __name__ == '__main__':
  main()
```

requirements.txt:

```
certifi==2024.8.30
charset-normalizer==3.4.0
idna==3.10
python-dispatch==0.2.2
requests==2.32.3
RPi.GPIO==0.7.1
urllib3==2.2.3
websocket-client==1.8.0
```

Приклад зібраних даних:

```
met labels are : ['eng.isActive', 'eng', 'exc.isActive', 'exc', 'lex', 'str.
isActive', 'str', 'rel.isActive', 'rel', 'int.isActive', 'int', 'foc.isActive',
'foc']

MET data: [True, 0.070344, True, 0.633367, 0.0, True, 0.224075, True, 0.052446,
True, 0.252325, True, 0.094797]
```

Висновки

У цій роботі було продемонстровано прототип системи, що інтегрує нейрокомп’ютерний інтерфейс на основі ЕЕГ-пристрою, та апаратну платформу Raspberry Pi для динамічної візуалізації когнітивних станів користувача. Цей підхід має низку важливих переваг:

1. Можливість безпосереднього контролю зовнішніх пристроїв, таких як світлодіоди, за допомогою мозкових сигналів, що дає змогу створювати системи біологічного зворотного зв’язку.
2. Гнучкість і масштабованість рішення на базі Raspberry Pi, що робить його придатним для широкого кола застосувань, від розумних будинків до нейроосвітніх тренажерів.
3. Можливість інтеграції з іншими технологіями, як-от штучний інтелект, для глибшого аналізу та класифікації когнітивних станів.

Результати проведеного дослідження демонструють, що поєднання BCI-технологій та апаратних рішень на базі Raspberry Pi відкриває перспективи для створення інтелектуальних систем, орієнтованих на потреби людини. Подальший розвиток цього напряму може суттєво вплинути на якість життя, продуктивність праці та розширення можливостей людини.

Серед недоліків виявлено такі: сигнал слабкий і швидко втрачається у міру висихання сенсорів. Досягти 100 % з’єднання вдається рідко, максимальне отримане значення було приблизно 60 %.

Для розроблення та користування прототипом потрібно попереднє отримання clientId.

Для отримання EEG Performance metric потрібна ліцензія. Для усунення недоліків потрібно замінити прилад.

Загалом представлений прототип демонструє перспективність поєднання BCI-технологій та апаратних рішень на базі Raspberry Pi як основи для створення систем людино-комп’ютерної взаємодії.

Список літератури

1. Cortex API Documentation [Electronic resource] // Emotiv. — Mode of access: https://emotiv.gitbook.io/cortex-api/ (date of access: 2.11.2024).
2. Overview of API [Electronic resource] // Emotiv. — Mode of access: https://emotiv.gitbook.io/cortex-api/overview-of-api-flow (date of access: 20.11.2024).
3. Performance metrics [Electronic resource] // Emotiv. — Mode of access: https://www.emotiv.com/pages/performance-metrics?_pos=1&_sid=dbfbb5e33&_ss=r (date of access: 22.11.2024).

References

Emotiv. *Cortex API documentation*. https://emotiv.gitbook.io/cortex-api/.
Emotiv. *Overview of API flow*. https://emotiv.gitbook.io/cortex-api/overview-of-api-flow.
Emotiv. *Performance metrics*. https://www.emotiv.com/pages/performance-metrics?_pos=1&_sid=dbfbb5e33&_ss=r.

Y. Volynets, S. Khmel, O. Pyechkurova

VISUALIZING COGNITIVE STATES ON A RASPBERRY PI PLATFORM FOR REAL-TIME BIOFEEDBACK USING A BRAIN-COMPUTER INTERFACE

This research article presents a prototype system that integrates an EEG-based brain-computer interface (BCI) and the Raspberry Pi hardware platform for the real-time visualization of a user's cognitive states. The key advantages of this approach include:

1. *The ability to directly control external devices, such as LEDs, using brain signals, which enables the creation of biofeedback systems.*
2. *The flexibility and scalability of the Raspberry Pi-based solution, making it suitable for various applications, from smart home systems to educational training.*

The study utilized the Emotiv INSIGHT EEG system, which is a 5-channel EEG devices with semi-dry polymer sensors, alongside a Raspberry Pi 4B microcontroller. As part of the research, a prototype was developed to visualize the user's emotional and mental state indicators through LEDs connected to the Raspberry Pi.

The results of this work demonstrate the potential of combining BCI technologies and Raspberry Pi-based hardware solutions as a foundation for developing human-centered intelligent systems. Further development in this area can significantly impact the quality of life, productivity, and human capability expansion.

One identified limitation in the study is the quality of the EEG signal, which is relatively weak and deteriorates over time as the sensors dry out. Achieving a stable and reliable connection was challenging, with the maximum observed value reaching approximately 60%. Additionally, developing and using the prototype requires prior acquisition of a client ID, and access to EEG data and "Performance Metrics" depends on obtaining a license. Overcoming these limitations may require replacing the EEG device with a more reliable alternative.

Overall, the prototype showcases the potential of integrating BCI technologies and Raspberry Pi-based hardware as a foundation for developing human-computer interaction systems. Future research and enhancements in this field could lead to significant progress in several domains, including neurorehabilitation, smart home applications, and educational training scenarios.

Keywords: *neuro-computer interfaces, BCI, Raspberry Pi.*

Матеріал надійшов 01.12.2024