

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра електроніки,
загальної та прикладної фізики

Кваліфікаційна робота магістра

**СУЧАСНІ ВИМІРЮВАЛЬНІ ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ
ФІЗИЧНОГО ЕКСПЕРИМЕНТУ**

спеціальності 171 Електроніка

Здобувач вищої освіти гр. ЕП.м-12

Ю. М. Роман

Науковий керівник,
канд. фіз.-мат. наук, ст.викл.

К. В. Тищенко

Завідувач кафедри ЕЗПФ
д-р фіз.-мат. наук, професор

І. Ю. Проценко

Суми 2022

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра електроніки, загальної та прикладної фізики
Спеціальність 171 – Електроніка, освітньо-наукова програма
«Електронні інформаційні системи»

ЗАТВЕРДЖУЮ
Зав. кафедри ЕЗПФ
І.Ю. Проценко
«14» листопада 2022 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Романа Юрія Михайловича

Тема роботи **Сучасні вимірювальні платформи для автоматизації фізичного експерименту**

затверджена наказом по університету від «14» листопада 2022 р., № № 1067-VI

2. Термін здачі студентом закінченої роботи 20 грудня 2022 року

3. Вихідні дані до роботи (актуальність, мета)

Сьогодні розгортання системи автоматизації фізичного експерименту може бути виконано на базі різноманітних вимірювальних платформ на базі мікроконтролерів та мікропроцесорів. Перспективним є застосування одноплатних комп'ютерів, котрі мають ряд унікальних характеристик, необхідних в цій області: низька вартість, хороша автономність роботи, достатня розрахункова потужність, універсальність. Пристрої такого класу мають операційну систему загального користування, що дозволяє використовувати широкий спектр сучасних програмних засобів та забезпечує загальну захищеність пристрою від несанкціонованого доступу. В системах автоматизації фізичного експерименту часто виникає потреба побудови систем зі зворотнім зв'язком, котрі можуть реалізовувати швидку реакцію на зміну зовнішніх параметрів. Таким вимогам добре відповідає вимірювальна платформа Raspberry Pi на базі мікропроцесора ARM архітектури.

Метою кваліфікаційної роботи магістра є огляд сучасних систем і технологій для проектування і розробки систем автоматизації фізичного експерименту. Розробка схеми пристрою та програмного забезпечення керування фізичним експериментом з використанням одноплатного комп'ютера Raspberry Pi.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить їх розробити)

1. Огляд мікропроцесорних платформ для побудови систем автоматизації

фізичного експерименту.

2. Аналіз методів керування силовим навантаженням за допомогою Raspberry Pi.

3. Опис методів роботи з енкодерами. Підключеними до платформи Raspberry Pi.

4. Підбір компонентів та розробка схеми пристрою автоматизованого керування експериментом.

5. Розробка програмного забезпечення автоматизованого керування експериментом.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди № 1, 2. Актуальність і мета роботи.

Слайди № 3-4. Мікропроцесорні платформи для автоматизації фізичного експерименту.

Слайди № 5-7. Підключення силового навантаження до Raspberry Pi.

Слайди № 8-9. Контроль руху за допомогою енкодерів.

Слайди № 10, 11. Розробка схеми пристрою керування фізичним експериментом.

Слайди № 12-14. Розробка та налаштування програмного забезпечення.

Слайди № 15-16. Висновки.

6. Дата видачі завдання 9.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз літературних даних	до 09.12.2022 р.	<i>вик.</i>
2.	Проведення експерименту, моделювання, розрахунків, обробка результатів	до 16.12.2022 р.	<i>вик.</i>
4.	Оформлення тексту кваліфікаційної роботи.	до 20.12.2022 р.	<i>вик.</i>
5.	Попередній захист роботи	21.12.2022 р., онлайн	<i>вик.</i>
6.	Захист роботи в екзаменаційній комісії	28.12.2022 р., онлайн	<i>вик.</i>

Здобувач вищої освіти

Роман Ю.М.

Науковий керівник

Тищенко К.В.

РЕФЕРАТ

Об'єктом дослідження кваліфікаційної роботи магістра є сучасні вимірювальні платформи для проектування та розробки систем автоматизованого керування фізичним експериментом в режимі реального часу.

Мета роботи полягає в розробці пристрою на базі апаратної вимірювальної платформи Raspberry Pi для збору даних з інкрементального енкодера високої роздільної здатності, а також розробка програмного забезпечення для керування рухом електричного двигуна на основі даних, отриманих з енкодера.

У роботі розглянуто сучасні апаратні вимірювальні платформи, зокрема Arduino, ESP8266/ESP32, STM 32 та Raspberry Pi, які можуть бути використані при розробці електронних систем збору, обробки даних та автоматизації фізичного експерименту.

Розроблено принципову схему електронного пристрою та програмне забезпечення мовою програмування Python, які дозволяють точно керувати положенням ротора електричного двигуна на основі системи зворотнього зв'язку з інкрементальним енкодером високої роздільної здатності.

Робота викладена на 44 сторінках, зокрема містить 22 рисунки, 1 таблицю, список цитованої літератури із 20 джерел.

КЛЮЧОВІ СЛОВА: RASPBERRY PI, СИСТЕМА АВТОМАТИЗАЦІЇ, ІНКРЕМЕНТАЛЬНИЙ ЕНКОДЕР, КЕРУВАННЯ РУХОМ, ЗВОРОТНІЙ ЗВ'ЯЗОК

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1	7
МІКРОКОНТРОЛЕРНІ ТА МІКРОПРОЦЕСОРНІ ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ ФІЗИЧНОГО ЕКСПЕРИМЕНТУ	7
1.1 Апаратні обчислювальні платформи для побудови автоматизованих систем	7
1.1.1. Мікроконтролерна платформа Arduino.....	8
1.1.2 Мікроконтролерні платформи сімейства ESP8266/ESP32	9
1.1.3 Мікропроцесорна платформа Raspberry Pi.....	10
1.1.4 Мікроконтролерні платформи на базі STM 32	11
1.1.5 Порівняння характеристик та можливостей апаратних платформ.....	13
1.2. Початок роботи з Raspberry Pi	16
1.3 Автоматизовані системи на основі Raspberry Pi.....	18
1.3.1. Використання концепції IoT при побудові систем керування науковим експериментом	18
1.3.2. Огляд системи керування науковим експериментом на основі мікроконтролера Raspberry Pi.....	20
1.4 Контроль руху в системах автоматизації фізичного експерименту	23
РОЗДІЛ 2	26
МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ	26
2.1 Керування силовим навантаженням за допомогою Raspberry Pi.....	26
2.1.1. Підключення MOSFET до Raspberry Pi з контакту GPIO.....	26
2.1.2. Керування силовими навантаженнями за допомогою реле.....	30
2.2 Контроль руху за допомогою енкодерів на Raspberry Pi.....	32
РОЗДІЛ 3	37
РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ РУХОМ	37
3.1 Розробка принципової схеми системи керування рухом.....	37
3.2 Розробка програмного забезпечення для платформи Raspberry Pi	39
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43

ВСТУП

При вирішенні задач пов'язаних з автоматизацією фізичного експерименту виникає потреба у проектуванні комплексної системи до складу якої входять апаратні та програмні засоби керування та моніторингу. Технічною складовою забезпечується можливість збору і первинного перетворення фізичних явищ у формат, зручний для подальшого аналізу і обробки у цифровій формі, мікропроцесорною (комп'ютерною) системою також забезпечується ряд мір по роботі із оцифрованими сигналами за необхідним алгоритмом (програмою). Програмні засоби, у свою чергу, дозволяють гнучко і з високим ступенем інтелектуальності виконувати обробку інформації, яка надходить від вимірювальних перетворювачів, і, якщо це необхідно, забезпечують керування апаратним забезпеченням.

Реалізація технічної складової систем автоматизації фізичного експерименту можливе з використанням різних засобів, починаючи з мікросхем логіки, закінчуючи комп'ютером з підключеними до нього модулями керування та збору даних. Проектування високопродуктивної системи автоматизації повинно включати підбір її компонентів виходячи не лише з технічних характеристик, а й економічного обґрунтування.

Сьогодні на ринку доступні вимірювальні платформи, та периферійне обладнання сумісне з ними, різного класу, як за розрахунковими можливостями, так і за ціною, яке дозволяє створювати системи автоматизації будь-якої складності.

Метою роботи є огляд сучасних вимірювальних платформ, які можуть бути використані для побудови автоматизованої системи керування фізичним експериментом; розробка системи аналізу даних на базі мікропроцесорної вимірювальної платформи RaspberryPi, отриманих з сенсора фізичної величини, та керування силовим навантаженням підключеним до неї.

РОЗДІЛ 1

МІКРОКОНТРОЛЕРНІ ТА МІКРОПРОЦЕСОРНІ ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ ФІЗИЧНОГО ЕКСПЕРИМЕНТУ

1.1 Апаратні обчислювальні платформи для побудови автоматизованих систем

Мікроконтролер – це компактна інтегральна схема, призначена для управління певною операцією у вбудованій системі. Типовий мікроконтролер включає процесор, пам'ять і периферійні пристрої вводу/виводу (введення-виведення) на одному чіпі.

Мікроконтролери, які іноді називають вбудованим контролером або мікроконтролерним блоком (MCU), зустрічаються в транспортних засобах, роботах, офісних машинах, медичних пристроях, мобільних радіоприймачах, торгових автоматах та побутовій техніці, серед інших пристроїв. Вони по суті своїй являють собою прості мініатюрні персональні комп'ютери (ПК), призначені для управління невеликими функціями більшого компонента, без складної front-end операційної системи (ОС)[1].

Інструмент автоматизації експерименту - це інструмент, який допомагає автоматизувати потреби в тестуванні програмного забезпечення, зменшуючи потребу в втручанні людини і, таким чином, досягаючи більшої швидкості, надійності та ефективності.

Процес автоматизації має три основні принципи конструкції:

1. Ефективність – проект повинен мінімізувати обчислювальні витрати та максимізувати використання апаратних ресурсів.

2. Гнучкість – процес автоматизації має бути прозорим у всіх своїх операціях, щоб користувачі можуть розширити його відповідно до існуючих або бажаних варіантів використання.

3. Відтворюваність. Проект повинен максимізувати прозорість системи.

З відомих платформ автоматизації були підібрані такі варіанти як Arduino,

ESP8266/ESP32, STM32 та Raspberry Pi.

1.1.1. Мікроконтролерна платформа Arduino

Arduino – це платформа з відкритим кодом, яка використовується для створення проектів електроніки. Arduino складається як з фізичної програмованої плати (яку часто називають мікроконтролером, показана на рис.1.1), так і з частини програмного забезпечення, або IDE (інтегроване середовище розробки), яка працює на вашому комп'ютері, використовується для запису та завантаження комп'ютерного коду на фізичну плату [2].

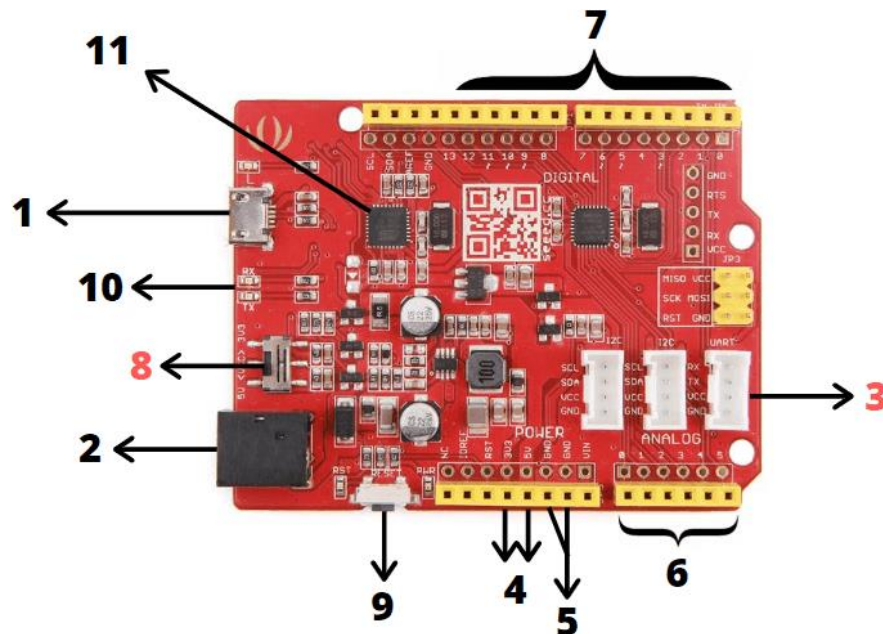


Рис 1.1. – Зовнішній вигляд Arduino Uno, на якій розташовані: 1- USB вхід; 2- вхід для постійного струму; 3- Grove роз'єм; 4- контакти на 3,3 та 5В; 5- контакт GND; 6- аналогові піни; 7- цифрові піни; 8- системний перемикач живлення; 9- кнопка перезапуску; 10- індикатор RX/TX; 11- мікроконтролер[2];

Платформа Arduino стала досить популярною у людей, які тільки починають з електроніки, і не дарма. На відміну від більшості програмованих плат, Arduino не потребує окремого апаратного забезпечення для завантаження нового коду на плату - ви можете просто використовувати USB-кабель. Крім того, Arduino IDE використовує спрощену версію C++, що полегшує навчання програмуванню.

Також, Arduino надає стандартний форм-фактор, який розбиває функції мікроконтролера на більш доступний пакет.

Апаратне та програмне забезпечення Arduino було розроблено для художників, дизайнерів, любителів, хакерів, новачків та всіх, хто цікавиться створенням інтерактивних об'єктів або середовищ. Arduino може взаємодіяти з кнопками, світлодіодами, двигунами, динаміками, GPS-блоками, камерами, Інтернетом і навіть смартфоном або телевізором. Ця гнучкість у поєднанні з тим фактом, що програмне забезпечення Arduino безкоштовне, апаратні плати досить дешеві, а програмне та апаратне забезпечення легко вивчити, призвело до великої спільноти користувачів, які внесли код та випустили інструкції для величезної різноманітності проектів на базі Arduino.

1.1.2 Мікроконтролерні платформи сімейства ESP8266/ESP32

Плати показані на рис.1.2. це ESP8266 і ESP32, які поставляються з ядрами Xtensa і запускають операційну систему реального часу FreeRTOS з платформами, що пропонують середній шлях між енергоефективністю, можливостями плат Arduino в режимі реального часу та повнофункціональним характером Raspberry Pi Linux SBC.

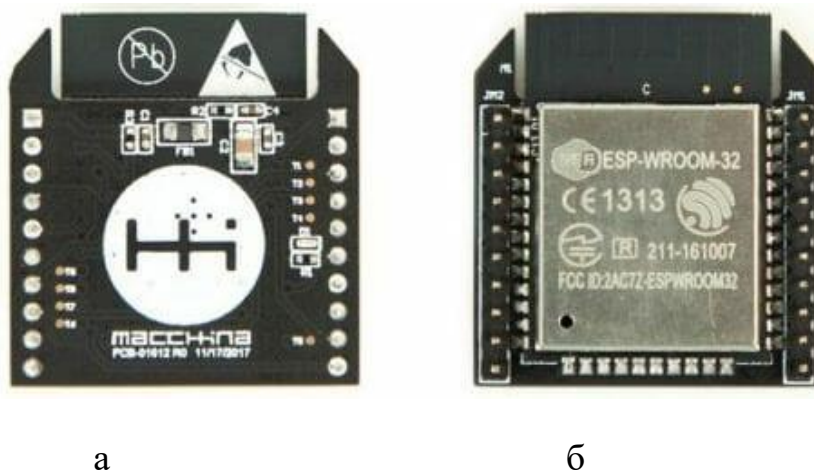


Рис. 1.2 – Зовнішній вигляд мікроконтролерних платформ ESP8266 (а) та ESP32 (б) [3].

Ці процесори розроблені таким чином, щоб бути енергоефективними, добре

інтегрованими з бездротовими технологіями і можуть працювати в грубих умовах з робочою температурою від -40°C до 125°C . Однак є деякі помітні відмінності між ESP32 та ESP8266, які описані в наступному абзаці:

- Швидкість Wi-Fi варіюється до 72,2 Мбіт / с для ESP8266 і до 150 Мбіт/с для ESP32
- ESP32 сумісний з Bluetooth на відміну від ESP8266
- ESP32 має 12-бітний SAR і два 8-бітних канали ЦАП в порівнянні з 10-бітним SAR ESP8266
- Хоча ESP32 має вісім каналів, ESP8266 працює за допомогою одного каналу
- ESP32 має кілька датчиків, включаючи сенсорний датчик з вісьмома каналами, а також датчик температури та ефекту холоду, тоді як ESP8266 не має вбудованих датчиків

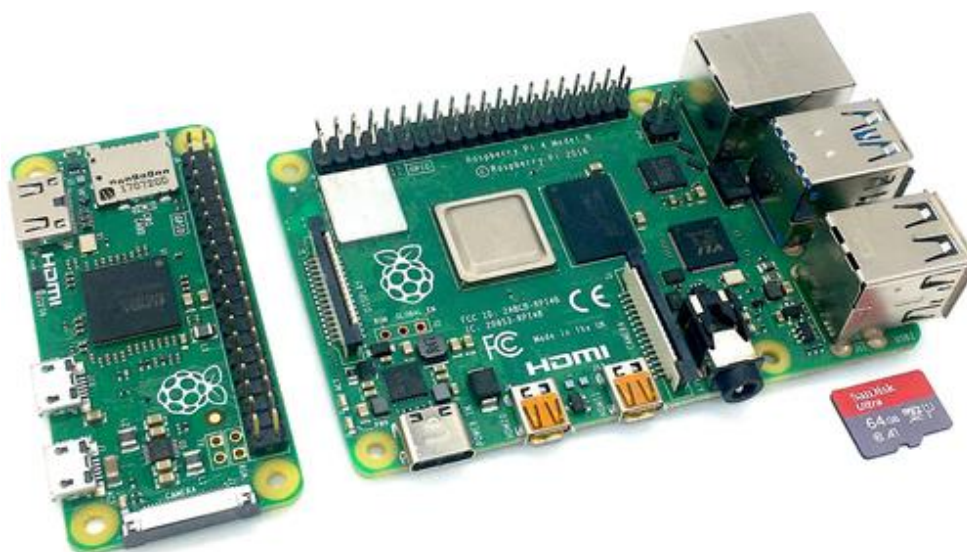
Тому вибір одного з цих SoC залежить від вимог експерименту та особливості процесів автоматизації. ESP8266 та ESP32 призначені для використання в носимій електроніці, промисловості IoT та мобільних пристроях, і знайде застосування в багатьох проектах автоматизації [3].

1.1.3 Мікропроцесорна платформа Raspberry Pi

Raspberry Pi – це лінійка одноплатних комп'ютерів (SBC), які родом з Великобританії і вперше були представлені в лютому 2012 року. Ці невеликі комп'ютери спочатку були розроблені для навчання студентів основам інформатики, але вони знайшли свій шлях у різних проектах, включаючи комерційні продукти [4].

При підключенні до монітора, USB-клавіатури та миші Raspberry Pi схожий на комп'ютер Linux початкового рівня. Raspberry Pi 4, система має 2 ГБ або 4 ГБ оперативної пам'яті, а також порт USB-C для живлення та два порти micro HDMI для підключення до двох дисплеїв. Існують також порти USB 2.0 і USB 3.0 для підключення периферійних пристроїв, а також гігабітний порт Ethernet і модуль

WiFi & Bluetooth для відповідно дротового або бездротового підключення до мережі.



а

б

Рис 1.3 – Зовнішній вигляд Raspberry Pi zero (а) і 4В (б) з карткою microSD для масштабу[4].

Хоча Raspberry Pi іноді продається як частина комплекту під ключ як на рис.1.3. , він часто поставляється з аксесуарами, включаючи блок живлення USB-C потужністю 15 Вт+ і карту microSD. Для початку роботи з платою потрібно встановити операційну систему, бажано Raspbian для початківців, за допомогою інсталятора NOOBS.

1.1.4 Мікроконтролерні платформи на базі STM 32

STM32 – це серія 32-розрядних мікроконтролерів, розроблених і проданих компанією STMicroelectronics. Існують різні типи та різновиди мікроконтролерів STM32, типовий приклад якої зображено на рис.1.3., які належать до сімейства мікроконтролерів ARM-архітектури[5].

ARM розшифровується як машина підвищеного ризику. Це одна з найпопулярніших архітектур, яка використовується в таких пристроях, як камери, мобільні телефони та вбудовані системні пристрої. Вони відомі своїм низьким

енергоспоживанням.

Ці мікроконтролери використовуються в різних сферах застосування, від простих принтерів до складних плат в транспортних засобах. Як результат, технічні знання щодо розробки мікропрограм та вбудованих систем з використанням мікроконтролерів STM32 є важливим набором навичок для інженера з електроніки та зв'язку.

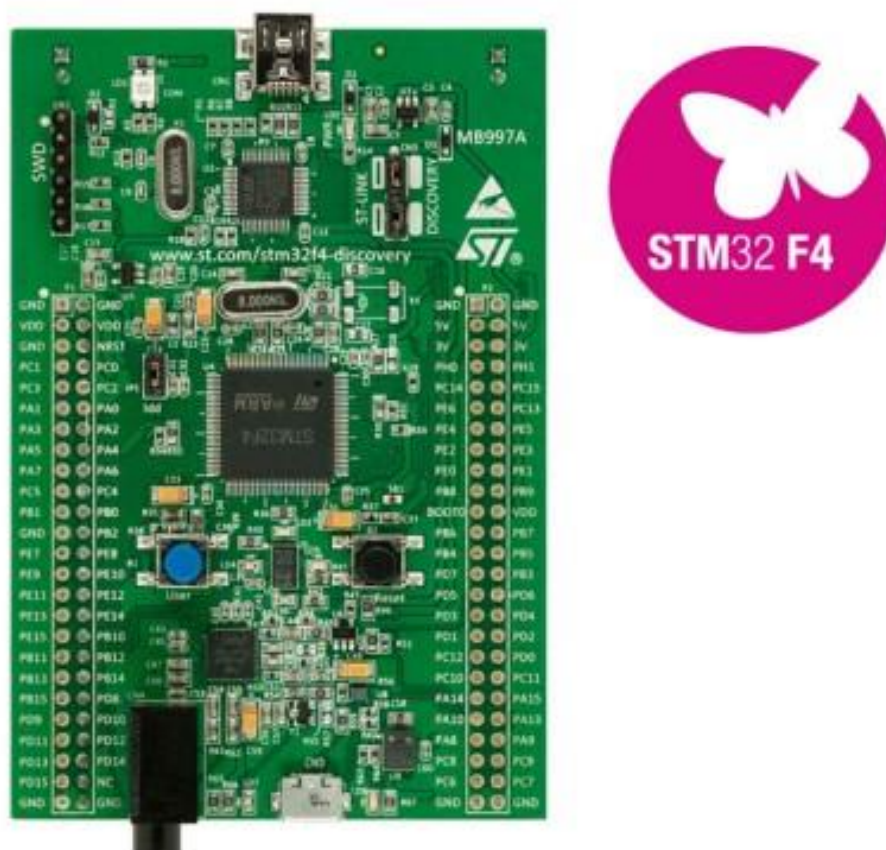


Рис 1.4. – Приклад типової плати STM32 F4 від компанії STMicroelectronics [5].

Оскільки STM32 є мікроконтролером на основі ARM, він має відмінності залежно від того, чи це Core 7, Core 4, Core 0 тощо. По суті, це плата прориву, яка дає доступ майже до всіх штифтів на STM32 IC, а також має вбудований акселерометр і магнітометр. Таким чином, це чудовий інструмент для розробки та тестування вашої програми / прошивки.

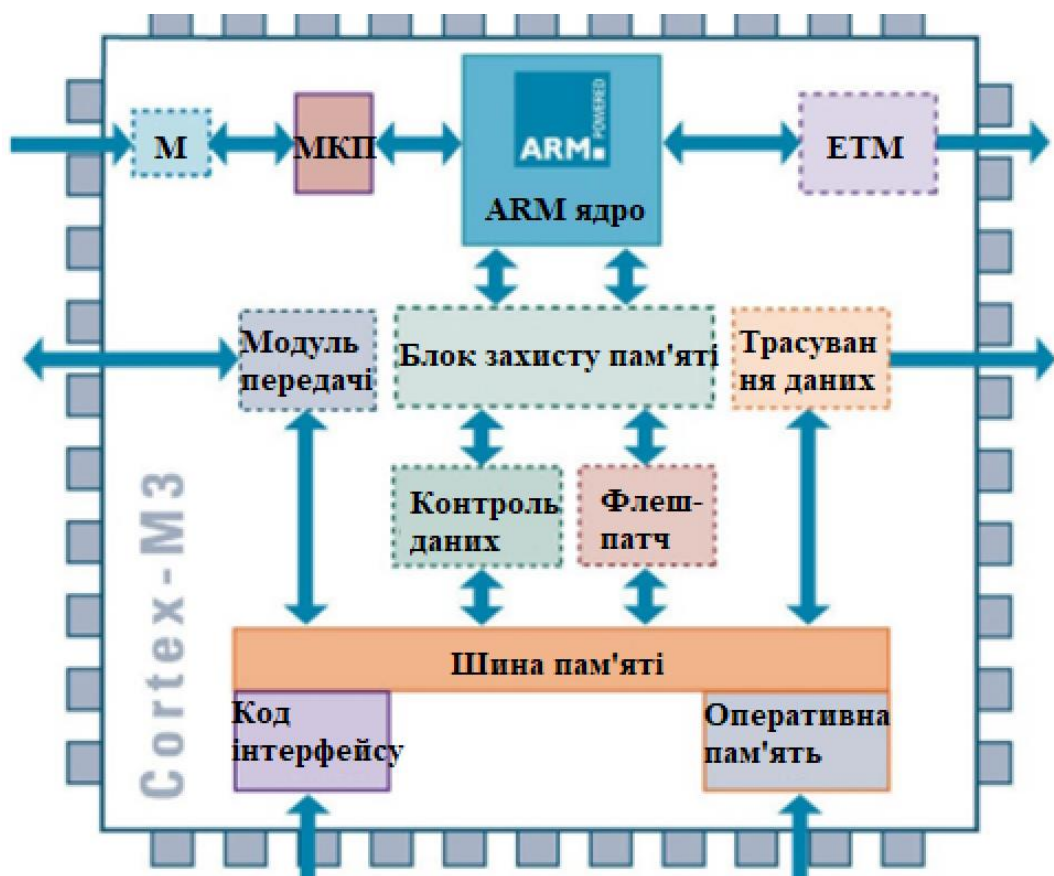


Рис 1.5. – Схема ARM-архітектури. Адаптовано із роботи [6].

ARM – це 32-бітна архітектура, типовий вигляд якої показано на рис.1.5., RISC (Reduced Instruction Set Computing). При маркуванні 32-бітної системи означає , що в адресній шині є 32 рядки, тобто; він може адресувати до 2^{32} місць. Інструкція RISC означає, що апаратне забезпечення виконує багато процесів обробки, що полегшує кодування інструкцій для процесорів ARM.

1.1.5 Порівняння характеристик та можливостей апаратних платформ

Для проведення нашого експерименту проведемо порівняння типових платформ для автоматизації процесів у вигляді таблиці.

Для проведення автоматизації різних експериментів та промислових процесів має значення потужність мікроконтролера, за таблицею 1.1 показано, що всі платформи мають свої плюси та мінуси, але Raspberry Pi – найпотужніша і універсальна платформа. Для освітніх цілей він має найбільше ресурсів, і він

Таблиця 1.1

Огляд характеристик Raspberry Pi, Arduino Uno, ESP32 та STM32

Платформа	Raspberry Pi	Arduino Uno	ESP32	STM32
Тип	Одноплатний комп'ютер	Мікроконтролер	Мікроконтролер	Мікроконтролер
ОС	Linux, Windows	Відсутня	Відсутня	Відсутня
Розміри	85,6 мм × 56,6 мм	68,6 мм × 53,4 мм	7 мм × 7 мм	56 × 43 мм
Вага	46г.	25г.	15г.	30г
Багатозадачність	Так	Ні	Ні	Так
Необхідність налаштування	Так	Ні	Так	Так
Процесор	64-біт	8-розрядна версія	32-розрядна версія	32-розрядна версія
Пам'ять	До 8 Гб	32 Кб	520 Кб	64 Кб
Тактова частота	4 × 1,5 ГГц	16 МГц	240 МГц	72 МГц
Порт камери	Так	Ні	Ні	Ні
HDMI	Так	Ні	Ні	Ні, але є SPI для підключення OLED дисплея
Вхідна напруга	5В	7-12В	3.3В	2.7- 5В
Порт GPIO	40 контактів: 5 В, 3,3 В, Ground Digital I/O	20 контактів: 5 В, 3,3 В, Ground Digital, аналог	48 контактів та 1 тепловий контакт	34 порта GPIO
Споживана потужність	3 000 мВт	<250 мВт	<250 мВт	<250 мВт
Інтерфейс робочого столу	Так	Ні, потрібно C/C++	Ні	Ні

ідеально підходить для проектів, що вимагають потужного комп'ютера або шлюзу Linux.

Переваги використання Raspberry Pi:

1. Велика обчислювальна потужність в компактній платі;
2. Велика кількість виділених інтерфейсів (UART, I2C, SPI, I2S, CSI, DSI) для підключення широкого спектру датчиків і електричних компонентів;
3. Висока можливість підключення (HDMI, USB, Ethernet, Wi-Fi, Bluetooth);
4. Висока простота використання з величезною спільнотою користувачів, великими ресурсами та простими для розуміння підручниками;
5. Працює як безголовий, так і як повноцінний настільний комп'ютер;
6. Не потрібен великий досвід програмування та невелика крива навчання, але слід очікувати деяких проб і помилок;
7. Вбудована графіка з підтримкою HDMI;
8. Низьке енергоспоживання (але вище, ніж у мікроконтролерів) і може житися від широкого спектру зовнішніх батарей і сонячних панелей;
9. Налаштовуваність і гнучкість в порівнянні з комерційними рішеннями;

Недоліки:

1. Не такий потужний, як традиційний ПК;
2. Немає вбудованого аналого-цифрового перетворювача;
3. Немає кнопки живлення або сплячого режиму (можливо з управлінням живленням HATS);

Оскільки на платах Raspberry Pi працює повноцінна операційна система Linux, вони можуть запускати розширені програми, такі як офісні пакети, фоторедактори, а також серверні програми, такі як веб-сервер Apache або nginx. Його велика кількість плюсів легко переважає його обмеження і робить Raspberry Pi чудовим дослідницьким інструментом, який можна використовувати майже для чого завгодно. Використання плати може варіюватися від інтерактивного та автономного моніторингу навколишнього середовища та відеозапису лабораторних експериментів, до станцій польових вимірювань та вдосконалених пристроїв замкнутого циклу, які можуть зчитувати різні вхідні дані, запускати інші

дії (наприклад, вмикати та вимикати світло або сервоприводи) автоматично обробляти дані та надсилати попереджувальні повідомлення.

Raspberry Pi розширюється завдяки апаратним аксесуарам та додатковим платам під назвою HAT (HardwareAttached onTop), а також розгалуженій екосистемі програмних інструментів та пакетів, які можна використовувати для створення більш складних проектів. Деякі з програмних інструментів включають Scratch або Python, щоб навчитися програмувати, тоді як модуль Pi Camera є прикладом апаратного аксесуара, який можна приєднати до Raspberry Pi SBC. Raspberry Pi Foundation співпрацює з третіми сторонами з безкоштовною онлайн-спільнотою під назвою CodeClubпропонує глобальну мережу безкоштовних клубів кодування. Онлайн-платформа має модулі, які допомагають навчитися кодувати за допомогою дошки Raspberry Pi разом з наступними програмними та апаратними компонентами:

- Інструменти візуального програмування Scratch;
- Веб-дизайн з HTML/CSS;
- Мова програмування Python;
- Додаткова плата Sense HAT зі світлодіодами та джойстиком;
- Програма Sonic Pi для створення музики і звуків за допомогою коду;
- BBC Micro:Бітовий кодований комп'ютер;

1.2. Початок роботи з Raspberry Pi

Плата Raspberry Pi містить процесор і графічний чіп, програмну пам'ять (RAM), різні інтерфейси і роз'єми для зовнішніх пристроїв. Деякі з цих пристроїв необхідні, інші є необов'язковими. RPі працює так само, як і стандартний ПК, для команд потрібна клавіатура, дисплей та джерело живлення. Він також потребує «накопичувача», але жорсткий диск такого типу, який є у типовому ПК, не підходить зберігаючи мініатюрний розмір RPі [8].

Для початку роботи з RPі потрібно підключити живлення, дисплей, micro SD-

картку, засоби керування в нашому випадку мишку та клавіатуру та мережевий кабель (не обов'язково при підключенні до Wi-fi), як це показано на рис.1.6.

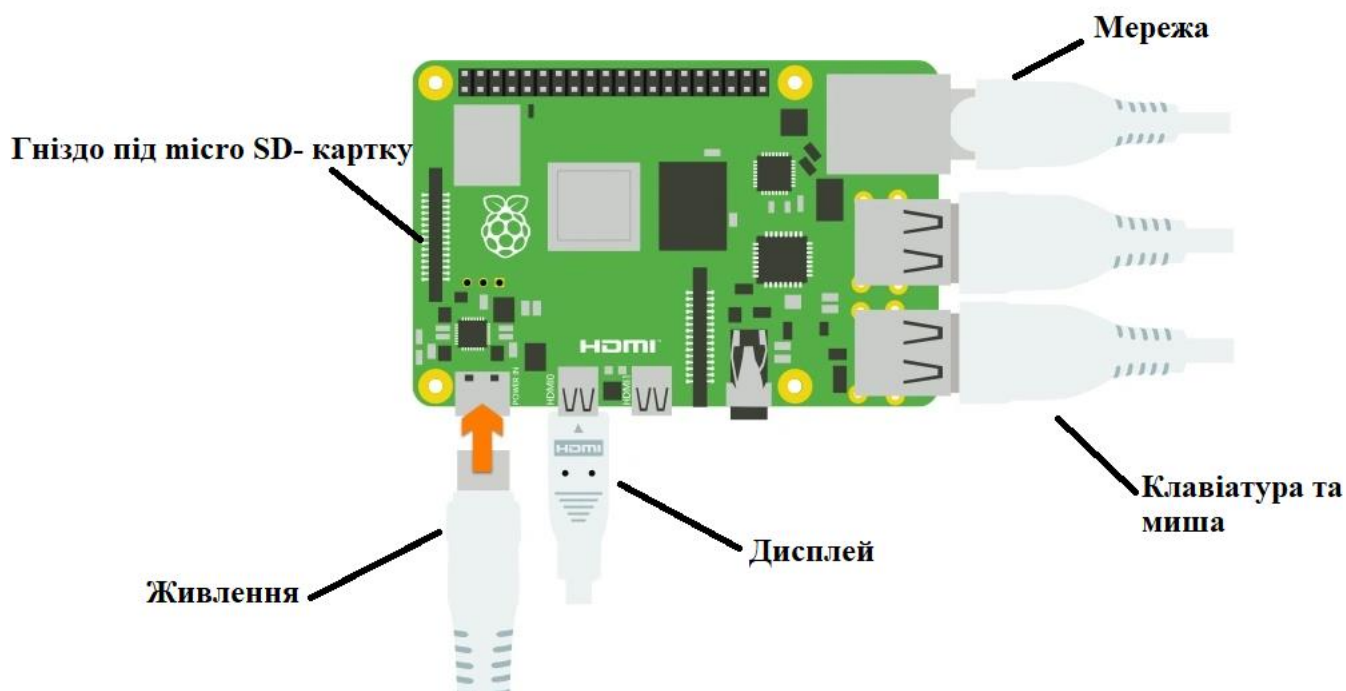


Рис 1.6. – Підготовка плати Raspberry Pi до встановлення ОС. Адаптовано із роботи [8].

Існує більше десятка різних операційних систем для Raspberry Pi, і навіть є спосіб запустити повну Windows 11 на Pi 4. Однак Raspberry Pi OS, спеціальна версія Debian Linux, оптимізована для Pi, є найкращою платформою для більшості випадків використання.

Raspberry Pi не має внутрішньої пам'яті, але замість цього завантажується з карти пам'яті micro SD. Для проведення експерименту потрібен обсяг пам'яті 8 ГБ або більше.

Найшвидший і простий спосіб встановити операційну систему на карту MicroSD - це використання офіційного Raspberry Pi Imager, головне меню якого показано на рис.1.7.



Рис 1.7. – Зовнішній вигляд вікна утиліти Raspberry Pi Imager [9].

Ця програма, доступна для Windows, macOS або Linux, завантажить та встановить останню версію Raspberry Pi OS. Є й інші способи зробити це, а саме завантаживши файл образу Raspberry Pi OS, а потім використовуючи сторонній додаток для його "запису", але Imager полегшує це.

1.3 Автоматизовані системи на основі Raspberry Pi

1.3.1. Використання концепції IoT при побудові систем керування науковим експериментом

Актуальним є автоматизація промислових процесів, які відбуваються без ручного керування або з мінімальним втручанням людини в робочий процес. В підрозділі представлений проект на концептуальній основі IoT [10].

IoT (Internet of Things) — це концепція автоматизації, яка передбачає керування в середовищі різноманітних речей/об'єктів через бездротове з'єднання та через унікальні схеми адресації які здатні взаємодіяти один з одним і

співпрацювати з іншими речами/об'єктами для створення нових додатків, послуг і досягнення спільних цілей.

Існують завдання до IoT і промислової автоматизації, наприклад, безпеки даних і послуг, цілісності даних, конфіденційності інформації. Стаття поєднує концепцію промислової робочої станції на Raspberry Pi і промислової автоматизації з використанням IoT, схема якої зображена на рис.1.8. Запропонована ефективна система промислової автоматизації, яка дозволяє користувачам ефективно контролювати промислові прилади/машини через Інтернет.

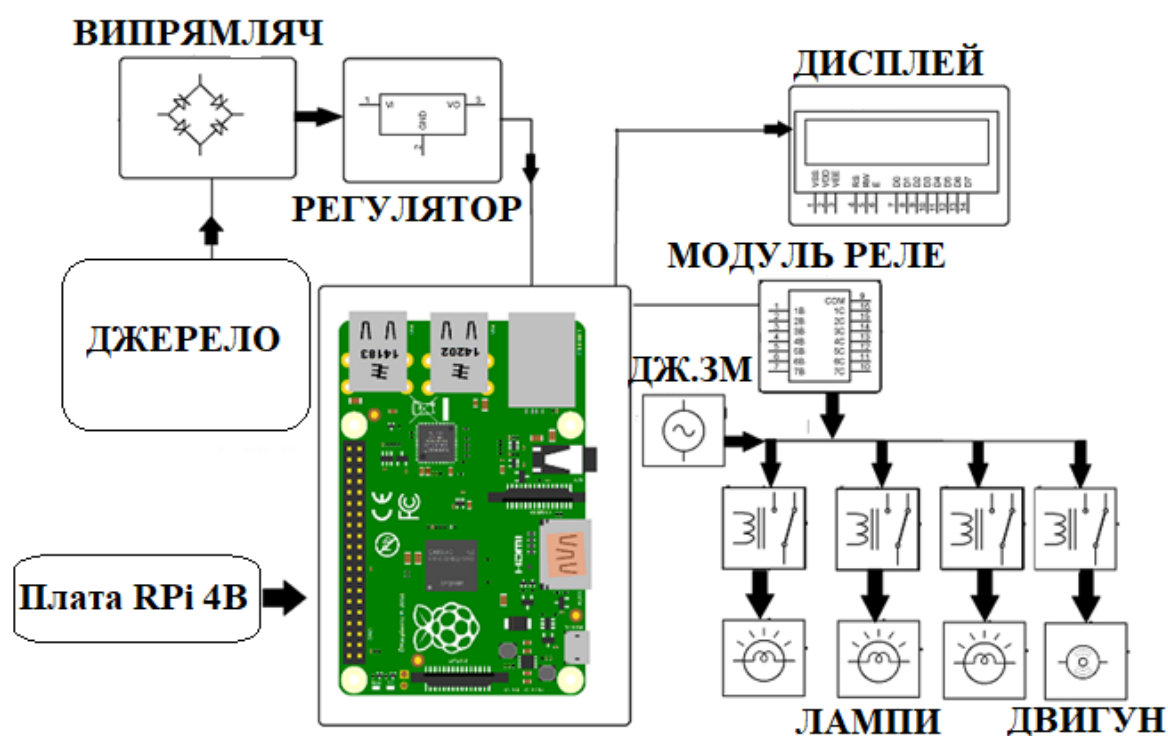


Рис 1.8. – Схема підключення системи автоматизації на основі IoT з використанням Raspberry Pi. Адаптовано із роботи [11]

Метою системи є створення невеликого промислового середовища, яке можливо контролювати лише на основі отриманих даних від датчиків промислового майданчика. Робота системи циклічна, тому втручання людини не допускається. Це означає, що система є адаптивною до навколишнього середовища. При розробці системи використовуємо IoT геско для інтерфейсу веб-

сервера та raspberry pi для обробки та запуску системи. Пристрій працює за наступним планом, користувач надсилає команду для машини через Інтернет за допомогою інтерфейсу IoT Gecko, з будь-якої точки світу. Далі в роботу вступає процесор Raspberry, який захоплює команди через роз'єм Wi-Fi. Raspberry Pi обробляє отримані дані та відображає їх на РК-дисплеї рис.1.9 [12].

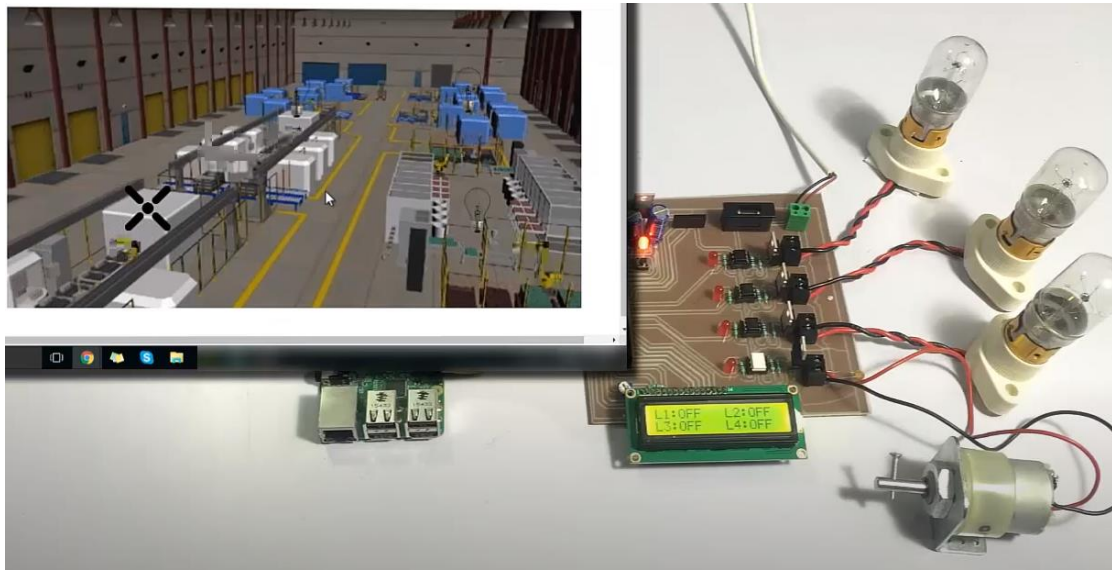


Рис 1.9. – Реалізація системи автоматизації [12].

Крім того, він вмикає/вимикає навантаження на основі отриманих команд для досягнення бажаного користувачем результату. Таким чином, система досягає автоматизації галузі через ІОТ за допомогою Raspberry Pi.

1.3.2. Огляд системи керування науковим експериментом на основі мікроконтролера Raspberry Pi

У роботах [13] та [14] представлено розробку автоматизованої системи керування науковим експериментом. Результатом цих досліджень є автоматизований стенд, за допомогою якого спостерігається методи поширення фронту полум'я повітря та ініціювання горючих газових сумішей. Паралельно використовувався інфрачервоний датчик для керування двотактним соленоїдом який відкриває газову діафрагму підставки. Для отримання вибухонебезпечної

суміші в різних концентраціях використовувався оригінальний програмований змішувач, на основі обчислювальних алгоритмів для точного керування кроковими електродвигунами які дозволяють досягти потоків повітря та легкозаймистих газів, щоб отримати однорідну та безперервну суміш на бажаному рівні концентрації. Проведені експерименти дозволяють краще зрозуміти формування та поширення фронту полум'я, полегшуючи знання та оптимізація часу роботи від редукторів для зменшення наслідків вибухів горючих газоповітряних сумішей.

Переваги використання газозмішувальних установок перш за все це стабільна однорідність газових сумішей. Надійність процесу досягається завдяки постійному якісному потоці змішаного газу. Прилади, які виробляють газові суміші, можуть забезпечити максимальну гнучкість від співвідношення змішування до кількості газу та місця використання та може виробляти саме необхідну газову суміш[13].

Крокові електродвигуни — це двигуни постійного струму, які обертаються з точними кроками, вони дуже корисні, коли потрібно розташувати щось дуже точно. Крокові електродвигуни, типовий показано на рис.1.10, контролюються шляхом застосування імпульсів постійного струму всередині їх внутрішніх котушок. Ці імпульси забезпечують обертання двигуна на один крок. Електродвигуни мають намагнічений редукторний сердечник, оточений певною кількістю котушок, які діють як електромагніти. Для переміщення валу крокового двигуна в дискретному числі кроків, які нам потрібні, щоб точно контролювати струм всередині котушок, подаючи електрику полярності подають під напругу одну котушку, а потім вал двигуна притягується до котушки під напругою та до неї фіксується на місці.

Для роботи біполярних крокових двигунів потрібен складний контролер або драйвер. Схематично стенд автоматизації експерименту виглядає наступним чином рис. 1.11.

Найпростішим способом використання та запуску крокових двигунів було використання драйвера (контролера), використовувався драйвер ТВ6600, сумісний з Raspberry Pi 4. Драйвер крокового двигуна - це а професійний інструмент для використання в проектах, де потрібні двофазні крокові двигуни. Пристрій сумісний

з платою Raspberry Pi 4 та іншими платами розробки, які можуть підтримувати напругу 5 В на виході сигналу. Драйвер, має широкий діапазон споживаної напруги від 9 В до 42В постійного струму. Він здатний видавати піковий струм 4 А, цього має бути достатньо для керування всіма двигунами, також він може підтримувати контроль швидкості та напрямку. Вихідним параметром може бути або тиск полум'я або будь-який інший параметр, змодельований у вигляді 2D/3D польового графіка одна змінна на одному або кількох часових кроках, перехідні сліди тиску на певних відрізках часу.



Рис 1.10. – Кроковий електродвигун для реалізації автоматизованого стенду з змішування вибухових сумішей [13].

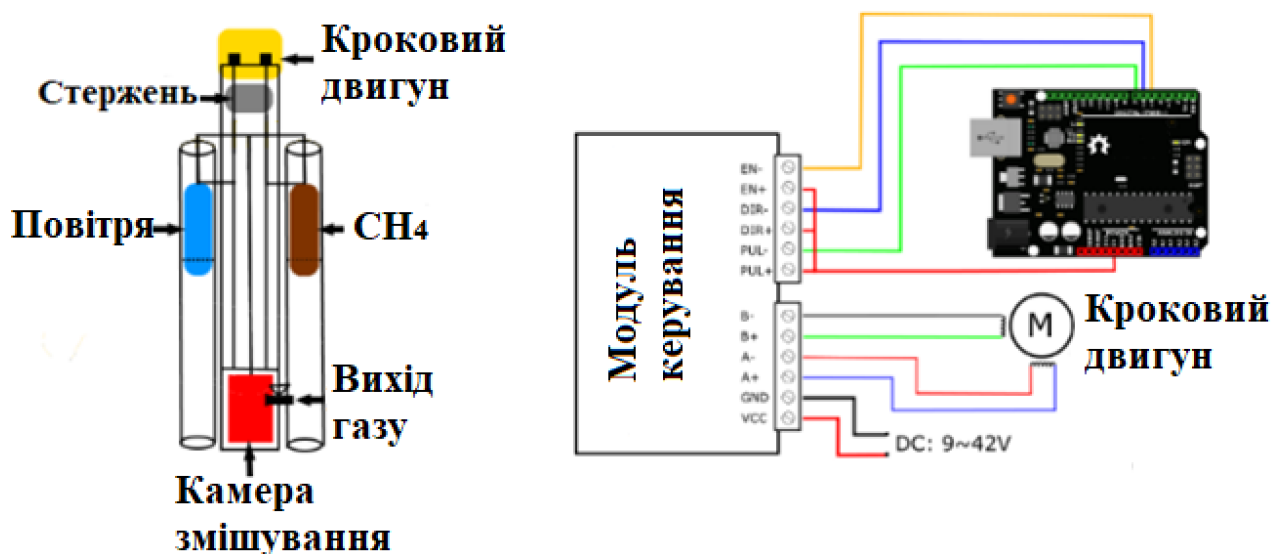


Рис 1.11. – Принцип дії повітряно-метанового змішувача. Адаптовано із роботи [13]

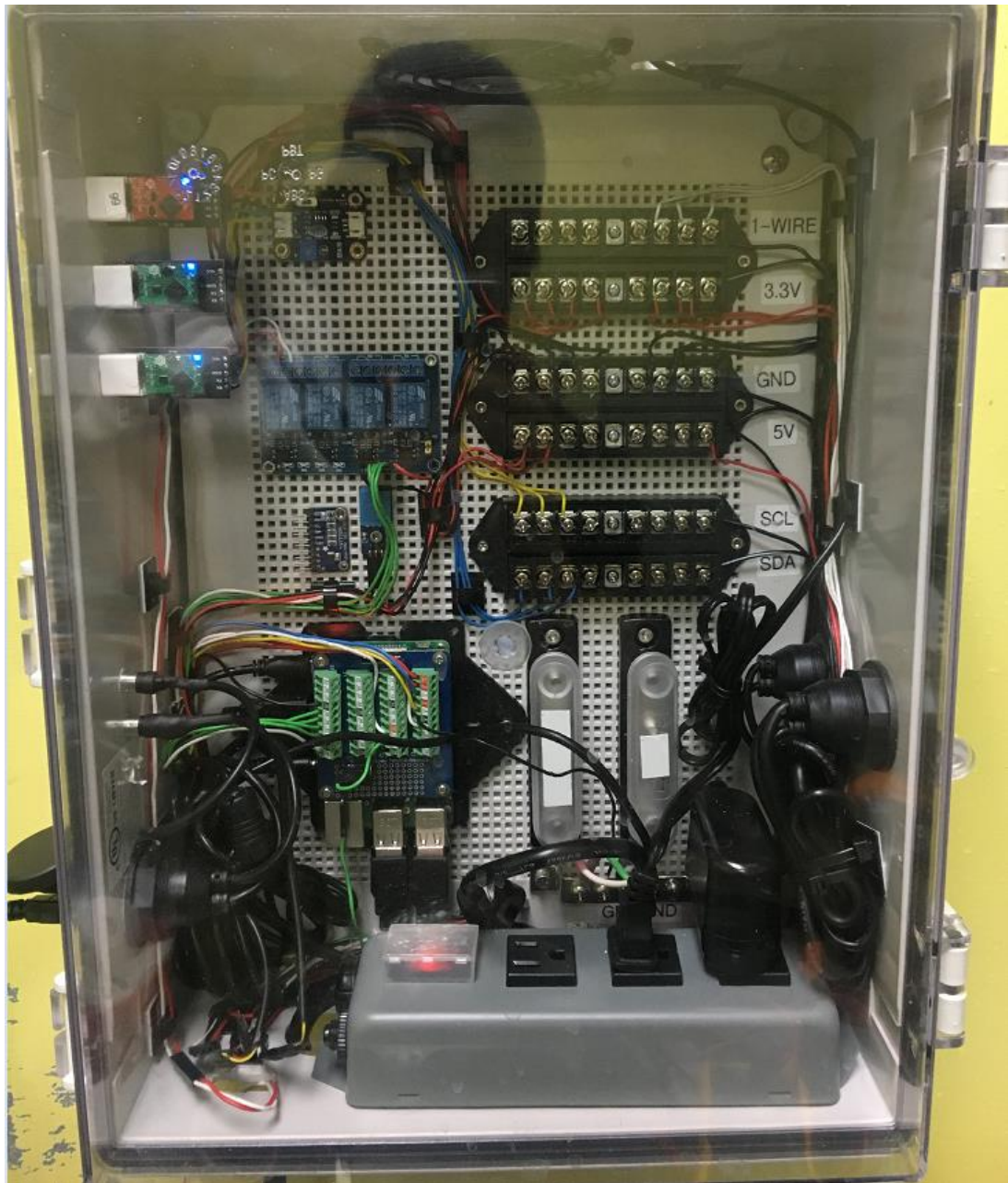


Рис 1.12. – Реалізований стенд для автоматизації експерименту на платі Raspberry Pi 4 в корпусі [14].

1.4 Контроль руху в системах автоматизації фізичного експерименту

В автоматизованих системах на базі мікроконтролерних та мікропроцесорних систем для контролю руху (лінійного та обертального) використовуються спеціальні пристрої кодери.

Кодери – це чутливі пристрої, призначення яких полягає в наданні зворотного

зв'язку про рух об'єктів системам управління [18]. Цей зворотний зв'язок дозволяє системі управління встановити, чи правильно переміщується або позиціонується об'єкт, що контролюється, і дозволяє вносити корективи або дії, які необхідно вжити, виходячи з руху і положення об'єкта.

Кодери зазвичай використовуються для вимірювання одного або декількох конкретних параметрів об'єкта, таких як його швидкість, положення, напрямок або для забезпечення підрахунку об'єкта або якогось пов'язаного з ним значення.

Існує кілька різних способів, за допомогою яких кодери можуть бути охарактеризовані для додатків управління рухом. Найбільш поширеним підходом є характеристика цих пристроїв за типом руху, що контролюється, будь то лінійний (прямолінійний) або обертальний. Три найпоширеніші типи кодерів – лінійні кодери, поворотні кодери та кутові кодери.

Лінійні кодери мають справу з переміщенням об'єктів по контуру або лінії. Цей тип кодера використовує перетворювач для вимірювання руху або відстані між двома точками, іноді використовуючи трос (більші відстані) або невеликий стрижень (менші відстані). У міру руху об'єкта перетворювач генерує вихідний сигнал, який використовується для встановлення руху або положення об'єкта [18].

Роторні кодери використовуються для забезпечення зворотного зв'язку про рух обертового об'єкта або пристрою, наприклад, вала двигуна. Поворотний кодер перетворює кутове положення рухомого вала в аналоговий або цифровий вихідний сигнал, який потім дозволить системі управління встановити положення або швидкість обертання вала.

Роторні кодери можуть містити вали або можуть мати конструкцію, відому як розточувальні кодери, що означає, що вони здатні бути безпосередньо встановлені поверх обертового вала, такого як у двигуна.

Кутові кодери схожі на поворотні кодери тим, що вони відстежують і забезпечують зворотний зв'язок про обертальний рух, але вони відрізняються тим, що кутові кодери, як правило, пропонують більш високу точність.

Абсолютні та інкрементальні кодери. Як лінійні, так і поворотні кодери доступні у вигляді абсолютних або інкрементальних кодерів, що описує потрібний

вихід сигналу для кодера. При абсолютному кодері вихідний сигнал, що генерується пристроєм, призводить до отримання унікального набору цифрових бітів, які відповідають конкретному положенню вимірюваного об'єкта. Навіть якщо потужність втрачена, абсолютний кодер за своєю конструкцією може визначити положення об'єкта, оскільки з кожною позицією пов'язаний певний цифровий сигнал. Роторні абсолютні кодери випускаються як в однооборотних, так і в багатовиткових конструкціях. Однооборотні кодери здатні надавати інформацію в межах будь-якого одного обертання вала. Багатооборотні кодери здатні надавати інформацію про положення над багатьма обертаннями положення вала, навіть великою кількістю обертань.

Абсолютні кодери використовуються в додатках, де важливо знати точно положення об'єкта. Вони також використовуються в ситуаціях, коли машина або процес неактивні протягом великого відсотка часу або рухаються з дуже повільною швидкістю. Інкрементні кодери використовують більш простий метод підрахунку руху і покладаються на встановлення положення об'єкта шляхом підрахунку кількості імпульсів, а потім використання цього підрахунку для обчислення позиції. Оскільки вони покладаються на підрахунок імпульсів, не існує унікального цифрового підпису, який можна використовувати для визначення абсолютної позиції.

РОЗДІЛ 2

МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ

2.1 Керування силовим навантаженням за допомогою Raspberry Pi

Потужні електроприлади, що працюють від мережі змінного струму напругою 220В, можуть потребувати управління мікропроцесорними пристроями. Для вирішення подібних завдань необхідно створення потужних схем управління, що перетворюють сигнали стандартного логічного рівня на сигнали управління ланцюгами більшої потужності. Без такої розв'язки експлуатація таких пристроїв була вкрай безпечною. Вирішення проблеми залежить від того, яким чином потрібно керувати навантаженням[15].

2.1.1. Підключення MOSFET до Raspberry Pi з контакту GPIO

Для проведення експерименту були підключені MOSFET в контакти GPIO за схемою з двома- (рис.2.1) та одним GPIO на сегмент (рис.2.2) [16]. Використано розширювач GPIO, який забезпечує 5 В від контактів, а не звичайний 3.3 В.

Для роботи з даною схемою підключення написаний код на мові програмування Python.

Python — це об'єктно-орієнтована мова програмування загального призначення, розроблена в основному для підвищення продуктивності програміста, а не коду, який він пише. Завдяки простоті коду подальше обслуговування програм, написаних на Python, стає легшим і приємнішим, ніж на Java або C++.

```
def display_init():
    GPIO.setmode(GPIO.BOARD)
    for p in PIN_DIGIT + PIN_SEGMENT_SET + PIN_SEGMENT_RESET:
```



```

GPIO.output(p, False)
GPIO.setup(p, GPIO.OUT)

def display_update(text: str=""):
    N_digits = len(PIN_DIGIT)
    text = text.ljust(N_digits)

    for p_digit, char in zip(PIN_DIGIT, text):
        GPIO.output(p_digit, True)
        glyph = FONT.get(char, FONT[' '])
        for seg, p_set, p_reset in zip(glyph,
PIN_SEGMENT_SET, PIN_SEGMENT_RESET):
            GPIO.output(seg and p_set or p_reset, True)
            time.sleep(T1)
            GPIO.output(p_set, False)
            GPIO.output(p_reset, False)
            time.sleep(T2)
        GPIO.output(p_digit, False)
        time.sleep(T3)

def display_shutdown():
    for p in PIN_DIGIT + PIN_SEGMENT_SET +
PIN_SEGMENT_RESET:
        GPIO.setup(p, GPIO.IN)
    GPIO.cleanup()

if __name__=="__main__":
    display_init()
    display_update("4242")
    display_shutdown()

```

Для економії на контактах GPIO можна використовувати триступеневу схему драйверів, так що при перемиканні контакта GPIO на вхід не приводяться в рух ні SET, ні RESET котушки. Таким чином, знадобиться лише 11 виходів GPIO для всього дисплея. З огляду на те, що транзистори і резистори нічого не коштують в порівнянні з розширювачами GPIO, а розширювачів GPIO може не бути в наявності.

```
def display_init():
    GPIO.setmode(GPIO.BOARD)
    for p in PIN_DIGIT + PIN_SEGMENT_SET +
PIN_SEGMENT_RESET:
        GPIO.output(p, False)
        GPIO.setup(p, GPIO.OUT)

def display_update(text: str=""):
    N_digits = len(PIN_DIGIT)
    text = text.ljust(N_digits)

    for p_digit, char in zip(PIN_DIGIT, text):
        GPIO.output(p_digit, True)
        glyph = FONT.get(char, FONT[' '])
        for seg, p_set, p_reset in zip(glyph,
PIN_SEGMENT_SET, PIN_SEGMENT_RESET):
            GPIO.output(seg and p_set or p_reset, True)
            time.sleep(T1)
            GPIO.output(p_set, False)
            GPIO.output(p_reset, False)
            time.sleep(T2)
        GPIO.output(p_digit, False)
```

```

time.sleep(T3)

def display_shutdown():
    for p in PIN_DIGIT + PIN_SEGMENT_SET +
PIN_SEGMENT_RESET:
        GPIO.setup(p, GPIO.IN)
    GPIO.cleanup()

if __name__=="__main__":
    display_init()
    display_update("4242")
    display_shutdown()

```

2.1.2. Керування силовими навантаженнями за допомогою реле

Якщо говорити про релейний модуль, то він є електромагнітним перемикачем і має дуже відому історію [17]. Релейний модуль вимагає меншої потужності для роботи і здатний перемикати електричні прилади. Така особливість релейних модулів робить їх кращими компонентами для автоматизації промислових процесів або наукових експериментів. Інші додаткові компоненти також доступні, але коли справа доходить до проектування недорогих пристроїв, кожен дизайнер завжди віддає перевагу релейним модулям.

Оскільки Raspberry Pi генерує лише 3.3 В на контактах GPIO, це означає, що плата не зможе жити модуль реле 5 В за допомогою Raspberry Pi. Рішенням з цього питання є використання перетворювачів рівня, ці перетворювачі рівнів є не що інше, як MOSFET, які перетворюють низькорівневі сигнали 3.3В в сигнали 5В. На наступному рис 2.3. показано сполучення модуля реле з RPi.

Написання коду для управління модулем реле за допомогою RPi. Перше, що є в цьому коді - імпорт модуля GPIO і модуля часу, за допомогою наступної команди:

```
import RPi.GPIO as GPIO
```

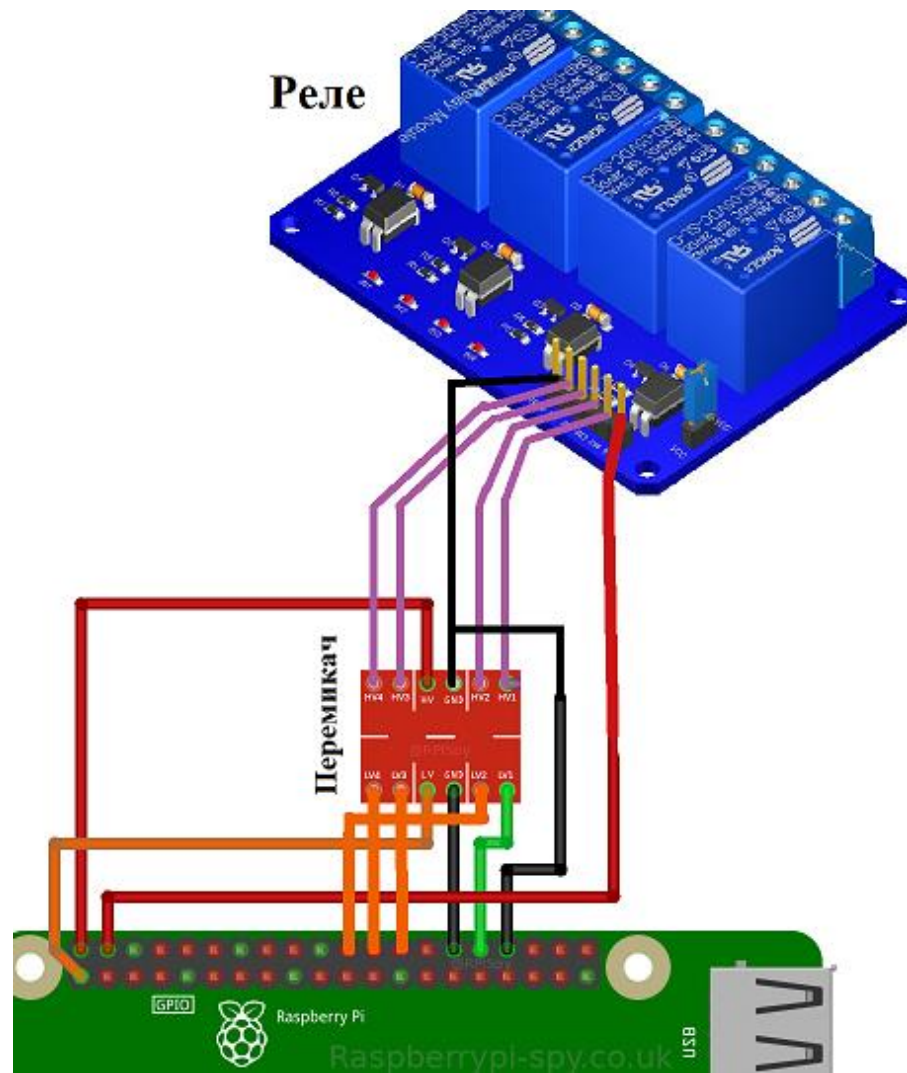


Рис 2.3. – Схема підключення реле до плати Raspberry Pi. Адаптовано із роботи [17]

Далі слід повідомити дощці, який метод нумерації потрібно використати.

```
GPIO.setmode(GPIO.BCM)
```

Наступним кроком є ініціалізація пінів, відповідно до вимог програми, ви можете ініціалізувати будь-який пін GPIO, але в цьому коді було використано наступні піни:

```
Relay1_GPIO = 12
```

```
Relay2_GPIO = 7
```

```
Relay3_GPIO = 8
```

```
Relay4_GPIO = 25
```

Далі потрібно визначити режим дії піна GPIO. У нашому випадку джерелом

живлення є GPIO Raspberry Pi 4. Тому в цьому випадку контакт GPIO був визначений як вихідний пін.

```
GPIO.output(Relay1_GPIO, GPIO.LOW)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.HIGH)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.LOW)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.HIGH)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.LOW)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.HIGH)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.LOW)
sleep(1)
GPIO.output(Relay1_GPIO, GPIO.HIGH)
sleep(1)
```

2.2 Контроль руху за допомогою енкодерів на Raspberry Pi

Для доцільного використання **роторного енкодера** (рис. 2.4), потрібно розібратися з його принципом роботи, маючи 2 внутрішніх контакти, які розривають ланцюг під час повороту ручки. При оберті ручки, можливо відчути, як вона «клацає», що вказує на те, що одна позиція була повернута [19]. Якщо внутрішні контакти спочатку були HIGH (приєднання ланцюга) після одного клацання, тепер вони обидва були б LOW (розрив ланцюга). Турбуватися про будь-які резистори чи інші компоненти не потрібно, оскільки кодер вбудував резистори на 10 кОм, що зводить струм до абсолютного мінімуму.

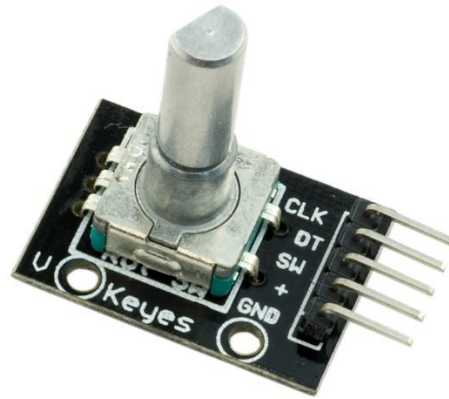


Рис 2.4. – Роторний енкодер[19].

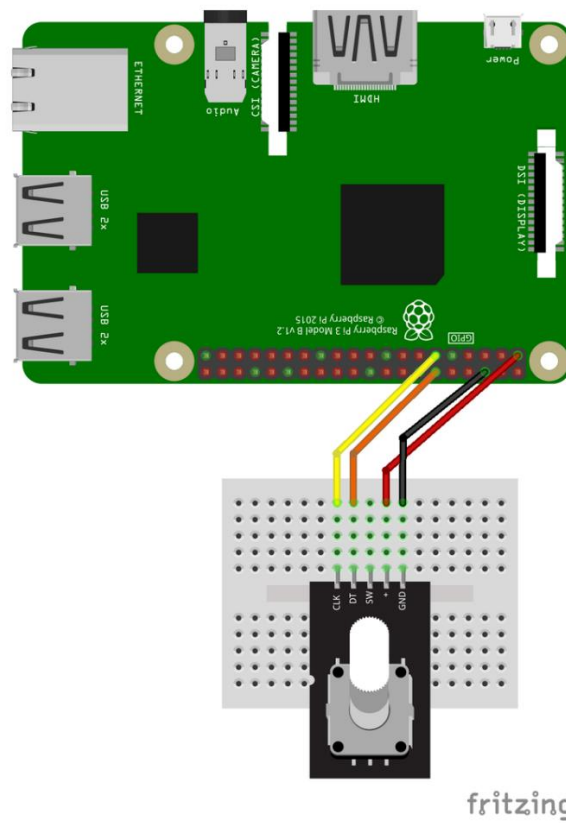


Рис 2.5. – Підключення роторного енкодера до плати RPi[19].

Код підключення енкодера на мові програмування Python:

```

from RPi
import GPIO
from time import sleep
clk = 17
dt = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(clk, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(dt, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
counter = 0
clkLastState = GPIO.input(clk)
try:

```

Це нескінченний цикл, в якому ми перевіряємо стани наших контактів CLK і DT, щоб спочатку побачити, чи обертається кодер, а потім для роботи нашого, якщо він обертається за годинниковою стрілкою або проти годинникової стрілки.

```

while True:
    clkState = GPIO.input(clk)
    dtState = GPIO.input(dt)
    if clkState != clkLastState:
    if dtState != clkState: counter += 1
    else: counter -= 1 print counter
    clkLastState = clkState
    sleep(0.01)

```

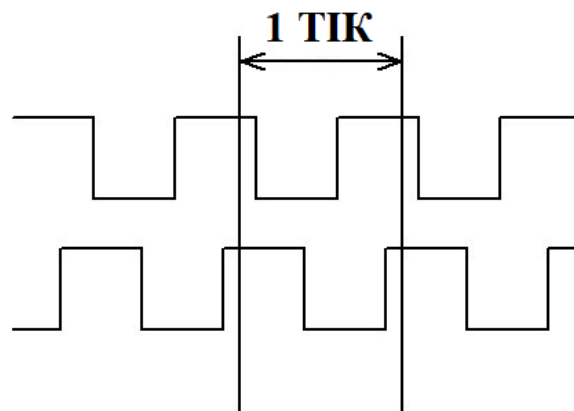
Інкрементальний енкодер на Raspberry Pi. Основою роботи даного енкодера є те що в нього є 3 піни, як показано на рис. 2.6, і якщо подивитися на енкодер зверху, то за годинниковою стрілкою можна ці піни позначити як А, О, В.



Рис 2.6. – Інкременетальний енкодер [20]

Пін О – працює на вихід, в свою чергу, піни А та В працюють на вхід, тобто "читають" стан піна О [22].

При поверненні енкодера за годинниковою стрілкою змінюється стан піна А потім В, далі змінюється стан піна А назад і потім знову В повертає своє первісне значення як на рис.2.7.



За один тік на право
Канал А 10011
Канал В 11001

За один тік на ліво
Канал А 11001
Канал В 10011

Рис 2.7. – Принцип роботи інкрементального енкодера . Адаптовано із роботи [20]

В один відрізок часу піни А та В приймають однакове значення, але на попередньому кроці мають різні положення, таким чином можливо зрозуміти в

яку сторону обертають ручку інкрементованого енкодера.

Код підключення енкодера виглядає наступним чином:

```
-- gpio-encoder.lua
gpio = fr.client.gpio
current = 0
function change_handler( data, pin, states )
  directions //це просто допоміжна таблиця - суть її в тому, що у
разі обертання за годинниковою стрілкою - до лічильника буде додаватися 1, а
проти -1
    local directions = { [false] = -1, [true] = 1 }
    states[pin] = data.value
    if data.value == 1 and states[not pin] == 1 then
      current = current + directions[direct]
      print( current )
    end
end
end
function main( argv )
  fr.run( )
  local states = {[true] = 0, [false] = 0}
  local A = assert(gpio.export( 2, "in" ))
  local O = assert(gpio.export( 3, "out" ))
  local B = assert(gpio.export( 4, "in" ))
  assert(A:set( "edge", "both" ))
  assert(B:set( "edge", "both" ))
  --
  assert(A:subscribe( "on_changed", change_handler,
false, states, A )) -- пин A false
  assert(B:subscribe( "on_changed", change_handler,
true, states, B )) -- пин B true
end
```

РОЗДІЛ 3

РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ РУХОМ

3.1 Розробка принципової схеми системи керування рухом

Нами було розроблено принципову схему системи контролю та керування рухом, яка в подальшому буде інтегрована до автоматизованого комплексу керування науковим експериментом, а саме вимірювання тензорезистивних властивостей тонких плівок. У якості обчислювальної системи, яка виконує усі функції з обробки сигналів та їх обробки була обрана мікропроцесорна апаратна платформа Raspberry Pi версії 4B, котра повністю задовільняє вимогам по функціональності та швидкості роботи для нашої автоматизованої системи.

У загальному вигляді схема установки показана на рис. 3.1. До її складу входить платформа Raspberry Pi, релейний модуль, інкрементальний енкодер та асинхронний двигун. В нашій системі енкодер представляє собою не окремий модуль, а конструкцію (рис. 3.2) з енкодерного диска (позиція 1), який обертається між здвоєним фототранзистором та інфрачервоним світлодіодом (позиція 2). Енкодерний диск має 5 тисяч рисок, що забезпечує точність фіксування руху 0,072 градуса на один імпульс (або ж 13,89 імпульсів на один градус) для кожного з каналів здвоєного оптодіода, чого більш ніж достатньо для поставлених цілей.

Наявність релейного модуля у схемі зумовлена необхідністю зміни полярності постійної напруги 127 В, яка приводить у рух двигун приводу мікрогвинта. Для цього два реле підключені за схемою перехрещених входів, а керування ними здійснюється за допомогою програмованих цифрових виводів (GPIO) модуля Raspberry Pi, як це описано в розділі 2 цієї роботи. Також у схемі є резистор номіналом 100 Ом, який призначений для обмеження струмів на фототранзисторах та світлодіоді.

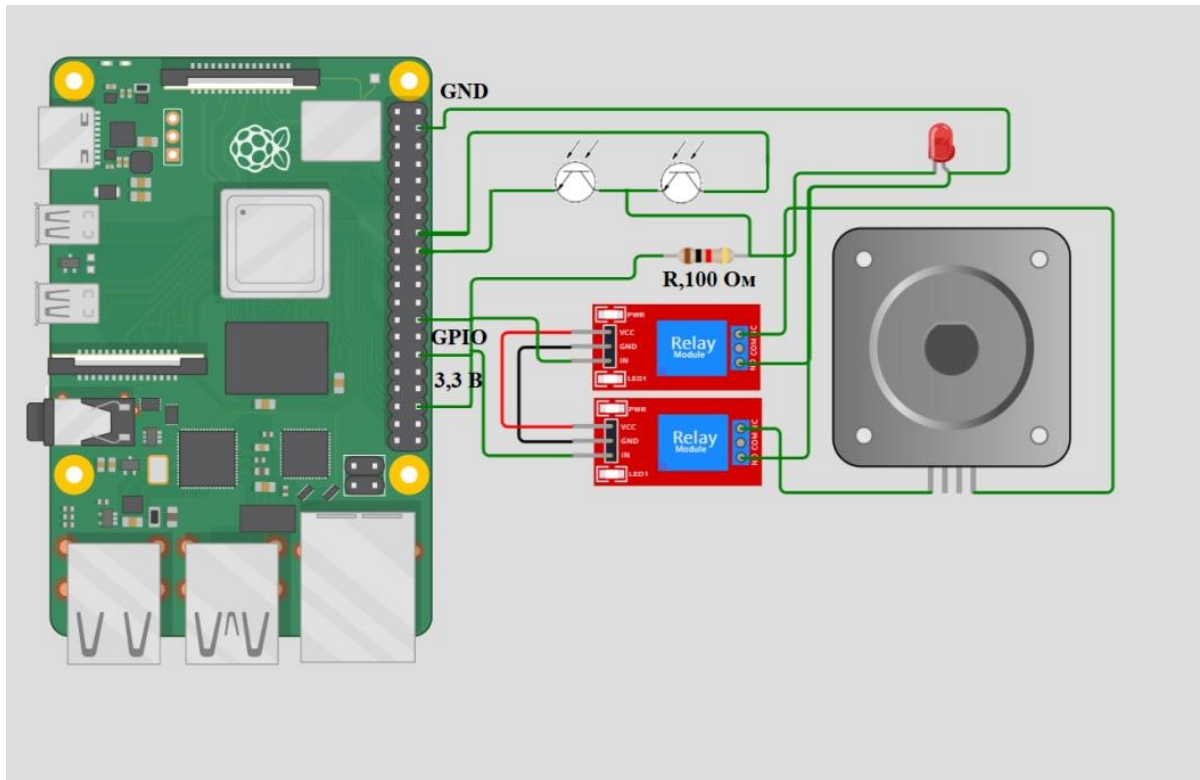


Рис. 3.1 – Принципова схема контролю обертового руху зі зворотнім зв'язком.

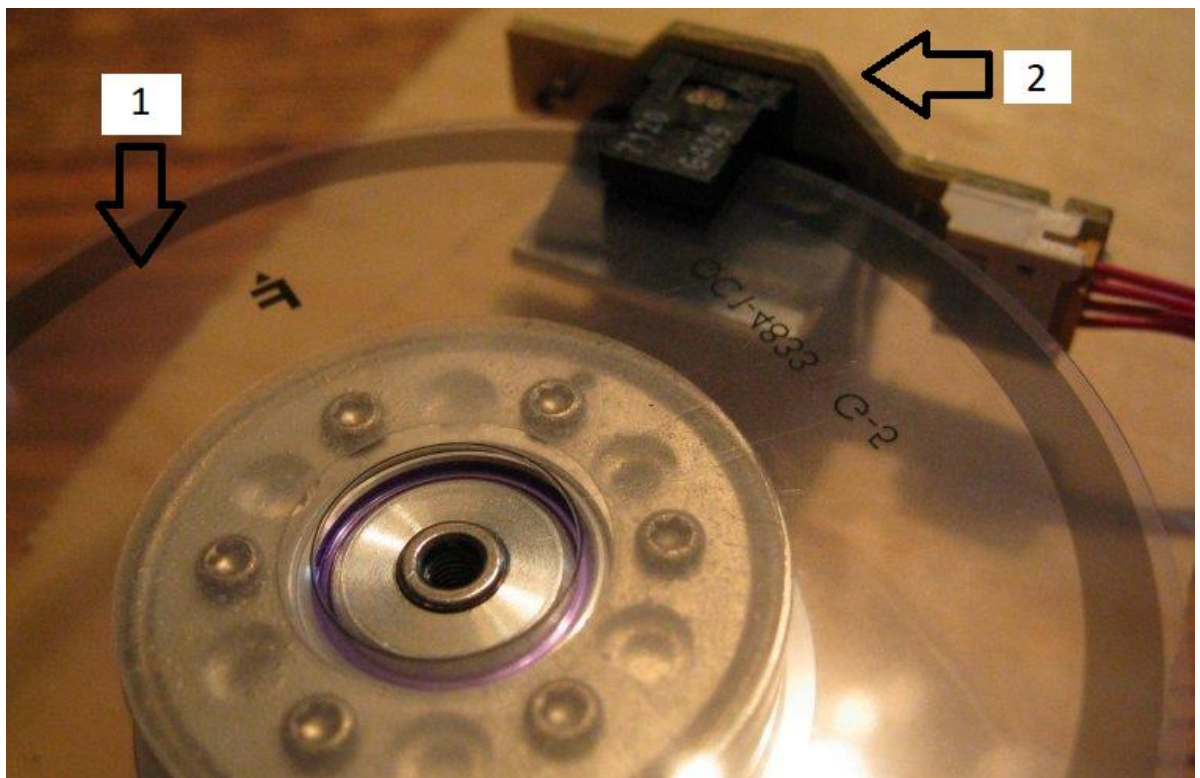


Рис. 3.2 – Зовнішній вигляд інкрементального енкодера

3.2 Розробка програмного забезпечення для платформи Raspberry Pi

Програмне забезпечення для контролю та керування рухом двигуна привода мікрогвинта розроблялось із використанням мультипарадигмної мови програмування високого рівня Python. Після розробки та відладки роботи програма була скомпільована у вигляді виконуваного скрипту, та налаштована таким чином, щоб запускатись разом із операційною системою вимірнувальної платформи Raspberry Pi.

При розробці програмного забезпечення використовувався модульний підхід, коли частина коду програми, що відповідає за виконання певного набору операцій, реалізовується у вигляді окремого програмного модуля, що значною мірою структурує програму, чітко розділяючи її на окремі бібліотеки.

Для реалізації поставленої задачі нам потрібно було розробити програмний код, який би керував роботою двигуна, а саме міг вмикати і вимикати його, а також змінювати напрямок обертання. Друга логічна частина програми відповідає за обробку сигналу з двохканального енкодера, а саме приймати сигнал з фототранзисторів, обробляти їх як переривання, та на основі цих даних розраховувати такі параметри руху як напрямок, кут повороту та швидкість обертання.

По-перше, потрібно отримати доступ до пінів GPIO за допомогою Python. Для цього використано модуль RPi.GPIO.

Для початку роботи з двигуном було створено теку в терміналі RPi 4B

```
sudo nano motor.py
//Написання коду починається з контактів GPIO
import RPi.GPIO as GPIO
import time
channel = 21

# GPIO setup
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(channel, GPIO.OUT, initial=GPIO.HIGH)
```

При подачі енергії на електродвигун використано наступні команди

```
def motor_on(pin):
    GPIO.output(pin, GPIO.LOW)
```

Та при відключенні

```
def motor_off(pin):
    GPIO.output(pin, GPIO.HIGH)
```

Далі формується цикл для налаштування параметрів двигуна

```
if __name__ == '__main__':
    try:
        motor_on(channel)
        print("ON")
        time.sleep(5)
        motor_off(channel)
        print("OFF")
        GPIO.cleanup()
    except KeyboardInterrupt:
        GPIO.cleanup()
```

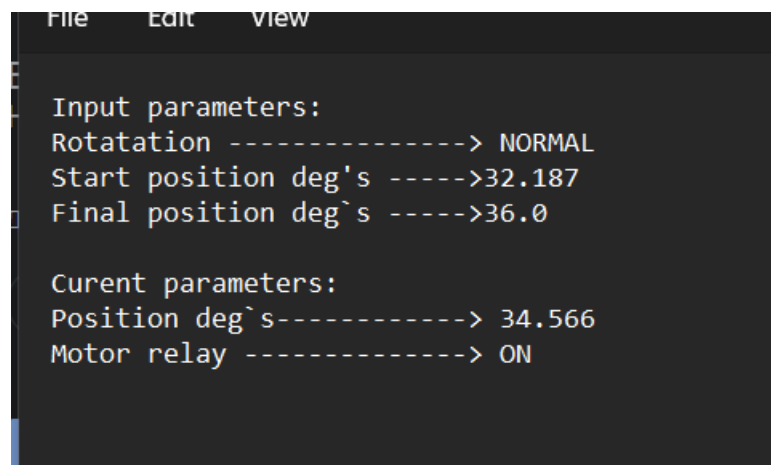
Код для керування рухом за допомогою реле представлений у підрозділах 2.1.1 та 2.1.2 цієї роботи, він був скомпільований нами в окрему бібліотеку із забезпеченням можливості зовнішнього виклику команд. Також у цій частині коду в режимі піна встановлено формат нумерації GPIO. Для обертання вперед керуючий вхід 1A встановлюється як «High» (високий логічний рівень сигналу), а інший вхід (1B) «Low» (низький логічний рівень сигналу). Для зміни напрямку обертання встановлюються інвертовані логічні рівні на керуючих виходах Raspberry Pi.

У нашій автоматизованій системі для контролю руху використано сигнали від двох каналів інкрементального двохканального енкодера. Оформлений у вигляді програмного модуля код аналізу сигналів енкодера (представлений у підрозділі 2.2

цієї роботи) реалізовує контроль параметрів руху, а саме фіксує зміну кута повороту та напрямку руху.

Останнім програмним модулем є система зворотнього зв'язку, яка синхронізує роботу двох попередніх частин програми, а саме в залежності від наперед заданих параметрів (необхідний кут повороту і напрям руху) задає керуючі сигнали для двигуна, контролює процес виконання завдання за сигналами з енкодера, а по досягненні запланованих кінцевих параметрів зупиняє двигун і чекає наступної команди.

Розроблена програма працює в термінальному режимі (рис. 3.3), приймаючи як вхідні параметри масив з трьох елементів: напрямок руху, поточне і кінцеве положення ротора двигуна. Вихідними параметрами є поточне положення у градусах та статус живлення двигуна. Ці параметри можуть бути передані до іншої програми у вигляді масиву.



```
File Edit View
Input parameters:
Rotatation -----> NORMAL
Start position deg's ----->32.187
Final position deg`s ----->36.0

Curent parameters:
Position deg`s-----> 34.566
Motor relay -----> ON
```

Рис. 3.3 – Результат виконання розробленого програмного забезпечення

Розроблений нами автоматизований застосунок призначений для використання як модуль керування рухом у програмному забезпеченні контролю науковим експериментом з дослідження тензоефекту в тонких плівках. Також воно може бути використано і в інших системах, які потребують керування і контролю обертового руху.

ВИСНОВКИ

1. Розглянуто сучасні апаратні платформи, які можуть бути використані при проектуванні автоматизованих систем керування фізичним експериментом. Проаналізовано їх технічні характеристики та можливість підключення до них перетворювачів фізичних величин і модулів керування.

2. Проаналізовано можливості мікропроцесорної вимірювальної платформи Raspberry PI для реалізації систем автоматизації фізичного експерименту різного ступеню складності. Виконано огляд існуючих систем автоматизації на базі цієї апаратної платформи.

3. Показано, що Raspberry PI добре підходить для проектування систем автоматизації, де необхідно реалізовувати програмний аналіз отримуваних результатів, для чого традиційно використовувались системи на основі комп'ютера і модулів керування-збору даних.

4. Виконано підбір апаратного забезпечення для побудови автоматизованої системи керування обертальним рухом та контролю зі зворотнім зв'язком на базі апаратної платформи Raspberry PI. Розроблено програмний алгоритм роботи з ротаційними інкрементальними енкодерами високої роздільної здатності.

5. Розроблено схему та програмне забезпечення автоматизованої системи контролю та керування рухом зі зворотнім зв'язком на основі інкрементального енкодера та мікропроцесорної платформи Raspberry PI. Показано, що розроблена система дозволяє фіксувати положення з роздільною здатністю 13,89 імпульсів на один градус та керувати ним з точністю ± 1 імпульс ($\pm 0,072$ градуса).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sean McManus., Raspberry Pi For Dummies .Artificial Intelligence Evolution // Adv. Mater. – V. 14(1). – 2019. – P. 880 – 885.
2. Огляд Arduino. Основи роботи з мікроконтролером [Електронний ресурс] – Режим доступу: <http://arduino.ua/about/>.
3. Schwartz M. Internet of Things with ESP8266. International Conference on Physical Experiments // Microcontrollers Journal. – V. 27(1). – 2019. – P. 450 – 500.
4. Мікроконтролер ESP8266/ESP32. Частина 1. [Електронний ресурс] – Режим доступу: <https://esp8266.ua/>.
5. Мікроконтролер STM32. Початок роботи з структурою ARM. [Електронний ресурс] – Режим доступу: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>
6. Використання портів GPIO в Raspberry Pi. Частина 1: [Електронний ресурс] – Режим доступу: <http://eah.me/raspberry-pi-gpio/>
7. Raspberry Pi 4В – перший запуск. Частина 2: [Електронний ресурс] – Режим доступу:<https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html>
8. Mason B. Microcontroller automation projects for conducting physical experiments //Journal of Analytical Science and Technology. – V. 27(1). – 2020. – P. 123 – 125.
9. Sanil P. Raspberry Pi based energy management system. Artificial Intelligence Evolution// Institute of Computer Engineering. – 2020. – V. 1(2). – P.63 – 144.
- 10.Joan M., Shahriar Alamgir Md., Mahabubur Rahman Md., Bhuiyan M. Raspberry Pi projects. International Journal of Engineering & Technology // International Journal of Engineering And Science. – V. 10(2). – 2019. – P. 1 – 12.
- 11.Brandt P., Brandt D. Science and Engineering Projects Using the Arduino and Raspberry Pi. Journal of Engineering // Automatic Control And Information Sciences. – V. 31(2). – 2021. – P. 1 – 97.

12. Tae-Kook K. Raspberry Pi: Amazing Projects from Scratch. International Conference on Physical Experiments // Microprocessor Engineering & Applications. – V. 33(2). – 2022. – P. 1 – 34.
13. Paul L., Deepak G. Automatic system of control of production processes. Experimental systems // American Journal Of Embedded Systems And Applications. – V. 21(2). – 2021. – P. 1 – 97.
14. Muck Y., Homam P. The project of automating the experiment of explosive mixtures using a microcontroller. International Journal of Engineering & Technology // J. Matr. Chem. – V. 14(2). – 2020. – P. 1 – 43.
15. Hachemi C., Sahu S.. EMG & EIMG measurement for Arm & Hand motions using custom made instrumentation based on Raspberry Pi // International Journal of Engineering And Science. – V. 10(2). – 2021. – P. 3 – 43.
16. Mohan H. M., Anitha S., Chai R.. Edge Artificial Intelligence: Real-Time Noninvasive Technique for Vital Signs of Myocardial Infarction Recognition Using Jetson Nano // Journal Of Electrical And Electronics Engineering. – V. 32(1). – 2021. – P. 3 – 10.
17. Kurniawan A. Microcontroller-Based Experimental Setup and Experiments for SCADA Education. Experimental systems // Physical Review B. – V. 15(7). – 2020. – P. 1 – 67.
18. Енкодери. Характеристики та типи енкодерів [Електронний ресурс] – Режим доступу: <https://www.azosensors.com/article.aspx?ArticleID=1730>
19. Використання роторного енкодера. Налаштування та підключення [Електронний ресурс] – Режим доступу: <https://thepihut.com/blogs/raspberry-pi-tutorials/how-to-use-a-rotary-encoder-with-the-raspberry-pi>
20. Злиття інженерії, контролю, кодування, машинного навчання та науки. Інкрементні ротаційні кодери з Raspberry Pi [Електронний ресурс] – Режим доступу: <https://aleksandarhaber.com/incremental-rotary-encoders-with-raspberry-pi-lpd3806-600bm/>