

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

(назва факультету)

(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: _____

Виконав: студент (ка) _____ курсу, групи _____
спеціальності (напряму підготовки) _____

(шифр і назва спеціальності (напряму підготовки))

_____ (підпис)

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

_____ (прізвище та ініціали)

Нормоконтроль

_____ (підпис)

_____ (прізвище та ініціали)

Рецензент

_____ (підпис)

_____ (прізвище та ініціали)

Анотація

Дипломна робота складається з пояснювальної записки та графічної частини (ілюстративний матеріал – слайди). Об'єм графічної частини дипломної роботи становить 12 слайдів. Об'єм пояснювальної записки складає 97 друкованих сторінок формату А4 (210×297). Дипломна робота складається з семи розділів, в яких нараховується 52 рисунків та 6 таблиць з даними. В роботі використано 11 літературних джерел. Метою даної магістерської роботи була сконструювати для подальшого використання робота маніпулятора, а також розробити систему управління роботом маніпулятором за допомогою смартфона по протоколу передачі повідомлень MQTT.

Розроблений проект робот маніпулятор на базі контролера ESP8266 Node MCU v3 має ряд переваг: оскільки контролер містить достатню кількість входів і виходів; містить вбудований стек протоколів TCP/IP; містить декілька послідовних портів UART, що будуть необхідні для відлагодження системи; забезпечує задовільною обчислювальною потужністю. За допомогою прошивки NodeMCU можна задіяти весь робочий потенціал для швидкої розробки пристрою. Для керування маніпулятором використовується додаток додаток MQTT dashboard. Він дозволяє підключатися до MQTT брокера Mosquitto і отримувати/відправляти дані на конкретні топіки. Особливість цього додатку полягає в тому що він легко масштабований і звичайний користувач може опублікувати чи отримати повідомлення не володіючи знаннями з програмування.

Ключові слова: РОБОТ МАНІПУЛЯТОР, ПРОТОКОЛ ПОВІДОМЛЕНЬ, КОНТРОЛЕР, ДИСТАНЦІЙНЕ УПРАВЛІННЯ, БРОКЕР ПОВІДОМЛЕНЬ.

Annotation

The thesis consists of an explanatory note and a graphic part (illustrative material - slides). The volume of the graphic part of the thesis is 12 slides. The volume of the explanatory note is 97 printed A4 pages (210 × 297). The thesis consists of seven sections with 52 figures and 6 data tables. 11 literary sources were used in the work. The purpose of this master's thesis was to design a robot manipulator for further use, as well as to develop a robot manipulator control system using a smartphone using the MQTT message transmission protocol.

The designed robot manipulator based on ESP8266 Node MCU v3 controller has several advantages: since the controller contains enough inputs and outputs for us; contains a built-in TCP / IP protocol stack; contains several UART serial ports that will be needed to debug the system; provides satisfactory computing power. With the NodeMCU firmware, you can harness the full potential for rapid device development. The MQTT dashboard is used to control the manipulator. It allows you to connect to MQTT broker Mosquitto and receive / send data to specific topics. The peculiarity of this application is that it is easily scalable and the average user can publish or receive messages without knowledge of programming.

Key words: MANIPULATOR OPERATION, MESSAGE PROTOCOL, CONTROLLER, REMOTE CONTROL, MESSAGE BROKER.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА	8
1.1. Апаратна обчислювальна платформа Arduino.....	8
1.1.1. Загальна інформація про Arduino	8
1.1.2. Історія створення Arduino	10
1.1.3. У чому перевага Arduino	11
1.1.4. Апаратна частина	12
1.2. Arduino Shields - плати розширення для Ардуіно	13
1.2.1. Навіщо потрібні плати розширення?.....	14
1.2.2. Підключення та програмування Arduino Shields	15
1.2.3. Різновиди плат розширення	16
1.3. Брокер повідомлень Mosquitto	19
1.4. Протоколи мережевого обміну CoAP, AMQP , MQTT.....	20
РОЗДІЛ 2. ТЕХНОЛОГІЧНА ЧАСТИНА	25
2.1. Підбір моделі сервопривода.	25
2.1.1. Поняття сервопривода і його будова.....	25
2.1.2. Внутрішній інтерфейс керуючих сигналів. Керування сервоприводом	26
2.1.3. Вибір сервопривода.....	31
2.2. Вибір макета робота маніпулятора	35
РОЗДІЛ 3. КОНСТРУКТОРСЬКА ЧАСТИНА	41
3.1. Робот маніпулятор на Arduino.....	41
3.1.1. Загальний опис проекту.....	41
3.1.2. Основні вузли для виконання проекту робота маніпулятора.....	41
3.2. Збір макету робота маніпулятора.....	42
3.3. Алгоритм роботи маніпулятора	49
3.4. Передача даних по протоколу MQTT.....	51
3.5. Алгоритм роботи mqtt протоколу	52

РОЗДІЛ 4. СПЕЦІАЛЬНА ЧАСТИНА.....	53
4.1. Запуск, налаштування і передача повідомлень через Mosquitto broker на операційній системі Windows	53
4.2. Опис клієнтського додатку на операційній системі Android	54
4.3. Мікроконтролер ESP8266	56
4.3.1. Технічні характеристики модуля ESP8266 NodeMCU:.....	57
4.3.2. Переваги і недоліки модуля NodeMcu v3	58
4.4. Програмування системи керування	60
4.4.1. Опис середовища програмування Arduino IDE.....	60
4.4.2. Розробка програми керування рукою-маніпулятором	62
РОЗДІЛ 5. ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА.....	68
5.1. Вдосконалення організації наукових досліджень	68
5.2. Планування та розрахунок перед виробничих затрат та капіталовкладень на проведення дипломного проекту.....	70
РОЗДІЛ 6. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	83
6.1. Правила безпеки при роботі з маніпулятором.....	83
6.2. Вимоги до робочого місця	85
6.3. Загальні вимоги безпеки при роботі з маніпулятором	87
РОЗДІЛ 7. ЕКОЛОГІЯ	89
7.1. Вимоги до приміщень для експлуатації моніторів і ПЕОМ. Шляхи дотримання цих вимог.	89
7.2. Добування електроенергії за рахунок спалювання мінерального палива. Забруднення довкілля при цьому та шляхи його зменшення.....	90
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	96

Перелік умовних позначень, символів, одиниць, скорочень і термінів

ЕОМ – електронна обчислювальна машина

ПК – персональний комп'ютер

ШІМ – широтно імпульсна модуляція

БЖ – блок живлення

ОС – операційна система

ПЗ – програмне забезпечення

IDE (Integrated Development Environment) — Інтегроване середовище розробки

SD (Secure Digital) – портативна флеш-карта

LCD (liquid crystal display) – рідкокристалічний дисплей

CoAP (Constrained Application Protocol) – Протокол обмеженого застосування

AMQP (Advanced Message Queuing Protocol) – Протокол черги розширених повідомлень

MQTT (Message Queue Telemetry Transport) – протокол обміну даними

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту

UART (universal asynchronous receiver/transmitter) — універсальний асинхронний приймач/передавач

API (A Programming Language) – масивно-орієнтована мова програмування

XML (Extensible Markup Language) – Розширювана мова розмітки

с – секунда

В – вольти

кг·см – кілограм на сантиметр

рис. – рисунок

мА. – міліампер

мм – міліметри

г – грам

ВСТУП

Жодна область приладобудування на сьогодні не може обійтись без автоматизованих систем. Автоматизація є важливим засобом спрощення виробництва, якої можна досягнути завдяки залученню у виробництво та остаточний контроль виробів роботизованих систем. Зараз у виробництві широко застосовується роботи-маніпулятори, які між собою поділяються на безліч різновидів, класів і моделей, проте не всі ці роботи здатні самостійно здійснювати певні операції і все ж таки потребують втручання технічного спеціаліста.

Вирішення задачі управління маніпулятором у складі різноманітних систем та механізмів є актуальним завданням, оскільки маніпулятори досить широко застосовуються у виробництві. Рух маніпулятора здійснюється за допомогою сервопривода. Переважна більшість сервоприводів мають у своєму складі окремий блок керування. А саме керуються здійснюється за допомогою широтно імпульсної модуляції, завдяки чому вони добре підходять для використання у лабораторних роботах як окремо, так і у складі робота-маніпулятора.

Тому, необхідність створення навчальної моделі робота-маніпулятора і керування ним за допомогою віддаленого управління на базі сучасного програмованого контролера є достатньо актуальною задачею на сьогодні.

РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА

1.1. Апаратна обчислювальна платформа Arduino

1.1.1. Загальна інформація про Arduino

Arduino (Ардуіно) — апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та сирововище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider) [1].

Поява перших мікроконтролерів, мініатюрних напівпровідникових приладів ознаменувала початок нової ери в розвитку мікропроцесорної техніки. Факт наявності в невеликому корпусі більшості системних пристроїв зробило мікроконтролер подібним до звичайного комп'ютера. У літературі їх почали називати однокристальними мікро ЕОМ. Практично з їх появою і зявилося бажання використовувати мікроконтролери як персональні комп'ютери. Але різні фактори стримували це бажання. Одним з таких факторів є те що пристрій на мікроконтролері досить важко зібрати, необхідні знання основ схемотехніки, вміння програмувати на асемблері. Також потрібні додаткова допоміжні пристрої і програматори. Одже без певного обсягу знань і досить дорогого обладнання не обійтися. Ця ситуація не дозволяла використовувати мікроконтролери в своїх потребах. Із появою пристроїв, які дають можливість працювати з різними мікроконтроллерами без матеріальної бази і дорогого обладнання, все кардинально змінилося. Таким прикладом є проект Ардуіно.

За допомогою Arduino можна реалізувати практично будь-який задум. Це може бути автоматична система керування поливом в теплицях, веб-сервер, або навіть автопілот для Мультикоптера. Отже, Arduino - це апаратна обчислювальна платформа для розробки пристроїв на базі мікроконтролера, на простій і зрозумілій мові програмування в інтегрованому середовищі Arduino. Це дешева плата яка є у вільному доступі, вона сумісна з усіма популярними операційними системами, а саме Windows, Mac і Linux, до якої можна підключити багато різних додаткових електронних компонентів. Додавши датчики, приводи, динаміки, додаткові модулі та додаткові мікросхеми, ми зможемо використовувати Arduino як "мозок" для будь-якої системи керування. Датчики можуть отримувати інформацію про навколишнє середовище, а також керувати різними виконавчими пристроями під'єднаними до плати ардуіно. Важко навіть перерахувати все, на що здатна ця платформа, тому що можливості обмежені тільки нашою уявою.

Платформа Ардуіно включає в себе електронний блок (плата розробника) і програмне забезпечення, а також інші електронні компоненти, які можна додавати до плати.

Плата Ардуіно це аналог материнської плати сучасного ПК. На ній встановлений мікроконтролер і мінімальна кількість елементів, необхідних для її роботи. Програмування мікроконтролера здійснюється за допомогою роз'єму для зв'язку з комп'ютером. Також є роз'єми для підключення зовнішніх пристроїв. Все, що необхідно для створення нового електронного пристрою, - це плата Arduino (будь якого типу), кабель зв'язку та ПК.

Другою частиною проекту Arduino є програмне забезпечення, що називається IDE. Воно об'єднало в собі середовище розробки і мову програмування, що представляє собою спрощений варіант мови C / C ++ доповненою бібліотеками для мікроконтролерів. В Arduino IDE додані

елементи, які дозволяють створення програм без вивчення апаратної частини, тому для роботи з Ардуіно досить знань основ програмування на мові на C / C ++.

1.1.2. Історія створення Arduino

У 2002 році програміст Массімо Банц (Massimo Banzi) був прийнятий на роботу на посаді доцента в Інститут проектування взаємодій міста Івреа (Interaction Design Institute Ivrea, IDII) для просування нових способів розробки інтерактивних проектів. Однак крихітний бюджет і обмежений час доступу до лабораторної бази зводили його зусилля практично нанівець. У проектах Банц використовував пристрій BASIC Stamp, розроблене каліфорнійською компанією Parallax. Stamp представляв собою невелику друковану плату з розміщеними на ній джерелом харчування, мікроконтролером, пам'яттю і портами введення / виведення для з'єднання з різною апаратурою. Програмування мікроконтролера здійснювалося на мові BASIC. BASIC Stamp мав дві проблеми: брак обчислювальної потужності і досить високу ціну - плата з основними компонентами коштувала близько 100 доларів. І команда Банц вирішила самостійно створити плату, яка задовольняла б усім їхнім потребам. Банц і його співробітники поставили собі за мету створити пристрій, що представляє собою просту, відкриту і доступну платформу для розробки, з ціною - не більше 30 доларів - прийнятною для студентської кишені. Хотіли вони і виділити чимось свій пристрій на тлі інших. Тому на противагу іншим виробникам, що економить на кількості висновків друкованої плати, вони вирішили додати їх якомога більше, а також зробили свою плату синьою, на відміну від звичайних

1.1.3. У чому перевага Arduino

Робота користувача з Ардуіно дуже схожа на роботу з ПК. Ардуіно дозволяє користувачеві зосередитися на розробці проектів, а не на вивченні пристрою і принципів функціонування окремих елементів.

Наявність великої кількості готових модулів і бібліотек прикладів програм дозволяє непрофесіоналам створювати готові працюючі пристрої для вирішення поставлених завдань. А варіанти використання Arduino обмежені лише можливостями мікроконтролера і наявного варіанту плати, ну і, звичайно, фантазією розробника. В результаті доступ до розробки мікропроцесорних пристроїв отримали не тільки професіонали електротехніки, але і прості любителі зробити щось своїми руками.

Існує безліч інших мікроконтролерів і мікропроцесорних пристроїв, призначених для програмування різних апаратних засобів: Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard і багато інших. Всі ці пристрої пропонують схожу функціональність і покликані звільнити користувача від необхідності заглиблюватися в дрібні деталі внутрішнього устрою мікроконтролерів, надавши йому простий і зручний інтерфейс для їх програмування. Ардуіно також спрощує процес роботи з мікроконтролерами, але на відміну від інших систем надає ряд переваг для викладачів, студентів і радіоаматорів:

Низька вартість. У порівнянні зі схожими апаратними платформами, плати Ардуіно мають відносно низьку вартість: готові модулі Ардуіно коштують не дорожче 50 \$, а можливість зібрати плату вручну дозволяє максимально заощадити кошти і отримати Ардуіно за мінімальну ціну.

Кросплатформеність. Програмне забезпечення Ардуіно працює на операційних системах Windows, Macintosh OSX і Linux, в той час, як більшість подібних систем орієнтовані на роботу тільки в Windows.

Проста і зручна середовище програмування. Середовище програмування Ардуіно зрозуміла і проста для початківців, але при цьому досить гнучка для просунутих користувачів. Вона заснована на середовищі програмування Processing, що може бути зручно для викладачів. Завдяки цьому, студенти, які вивчають програмування в середовищі Processing, зможуть легко освоїти Ардуіно.

Розширюване програмне забезпечення з відкритим вихідним кодом. Програмне забезпечення Ардуіно має відкритий вихідний код, завдяки цьому досвідчені програмісти можуть змінювати і доповнювати його. Можливості мови Ардуіно можна також розширювати за допомогою C++ бібліотек. Завдяки тому, що він заснований на мові AVR C, просунуті користувачі, що бажають розібратися в технічних деталях, можуть легко перейти з мови Ардуіно на C або вставляти ділянки AVR-C коду безпосередньо в програми Ардуіно.

Розширюване відкрите апаратне забезпечення. Пристрої Arduino побудовані на базі мікроконтролерів Atmel ATmega8 і ATmega168. Завдяки тому, що всі схеми модулів Ардуіно опубліковані під ліцензією Creative Commons, досвідчені інженери і розробники можуть створювати свої версії пристроїв на основі існуючих. І навіть звичайні користувачі можуть збирати дослідні зразки Ардуіно для кращого розуміння принципів їх роботи і економії коштів.

1.1.4. Апаратна частина

Плата Arduino складається з мікроконтролера Atmel AVR, а також елементів обв'язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором. У

мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен [1].

На концептуальному рівні усі плати програмуються через RS-232 (послідовне з'єднання), але реалізація даного способу різниться від версії до версії. Новіші плати програмуються через USB, що можливо завдяки мікросхемі конвертера USB-to-Serial FTDI FT232R. У версії платформи Arduino Uno як конвертер використовується контролер Atmega8 у SMD-корпусі. Дане рішення дозволяє програмувати конвертер таким чином, щоб платформа відразу розпізнавалася як миша, джойстик чи інший пристрій за вибором розробника зі всіма необхідними додатковими сигналами керування. У деяких варіантах, таких як Arduino Mini або неофіційній Boarduino, для програмування потрібно підключити до контролера окрему плату USB-to-Serial або кабель.[1]

1.2. Arduino Shields - плати розширення для Ардуіно

Ключовою перевагою платформи Arduino є її популярність і простота у використанні. Виробники електронних пристроїв підтримують платформу Ардуіно і випускають спеціальні версії плат, що розширюють базову функціональність Ардуіно. Такі плати називають платами розширення. Їх інша назва arduino shield або Шилд, вони виконують різноманітні завдання і спрощують використання плати. Використовують ардуіно шилди з різноманітними пристроями Arduino: LCD-екранами (Шилд LCD), двигунами (Шилд драйверів двигунів), датчиками (sensor shield), для отримання координат і часу з супутників GPS (модуль GPS), SD-картами (data logger) та іншими.

Плата розширення Ардуіно – це пристрій призначений для використання певних функцій і підключається за допомогою стандартних роз'ємів на корпусі плати. На ардуіно шилді розміщені всі необхідні електронні компоненти, а взаємодія з основною платою відбувається через

стандарстні піни Ардуіно. Зазвичай живлення на плату розштрення подається з плати Ардуіно, но можливий варіант подачі живлення з інших джерел. Інші компоненти підключаються до плати розширення через вільні піни що залишилися. Одночасно підключити можна декілька плат розширення (рис. 1.1).

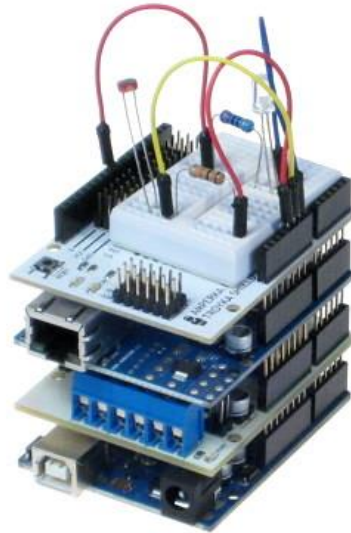


Рисунок 1.1 – Приклад підключення плат розширення до Ардуіно

1.2.1. Навіщо потрібні плати розширення?

Плати розширення (рис. 1.2) призначені для 1) розширення функцій Ардуіно, і 2) економії часу. Навіщо витратити свій час, проектуючи, розміщуючи, припаивая і налагоджуючи те, що можна взяти вже в зібраному, компактному варіанті, і відразу почавши використовувати в своїх проектах? Тим паче що такі плати досить дешеві, продумані і зібрані на якісному обладнанні.



Рисунок 1.2 – Зовнішній вигляд плати розширення

Незважаючи на дешевизну таких плат розширення їх кінцева ціна завжди буде вищою за ціну окремих комплектуючих, і можна зробити аналогічний варіант такому шилду дешевше. Але витрачений час на створення такої плати самостійно не вартує потрачених грошей. Тим паче вартість плат постійно знижується, тому найчастіше вибір робиться на користь використання готових пристроїв.

1.2.2. Підключення та програмування Arduino Shields

Для підключення плати розширення потрібно просто «надіти» її зверху на плату Ардуіно. Контакти плати розширення мають тип гребінки (тато), вони легко встановлюються в роз'єми основної плати.

Плати розширення призначені для конкретної версії плати Ардуіно, проте більшість шилдів для Arduino Uno підходять і нормально працюють з платами Arduino Mega. Піни контактів на платі Arduino Mega виконані таким чином що перші 14 цифрові піни і піни з протилежного боку плати збігаються з розташуванням контактів на Arduino Uno, тому в неї легко стає плата розширення від Arduino Uno.

Програмування схем з встановленими шилдами не відрізняється від звичайного програмування плати Ардуіно, тому що з точки зору контролера ми просто підключили пристрої до його пінів. В скетчі програми ініціалізуємо ті піни, які підключіне до плати розширення з відповідними

контактами на основній платі. Як правило виробники плат розширення вказують відповідність контактів на ШІлді або в окремій інструкції.

Читання або запис сигналів на платі розширення проводиться теж звичайним методом: за допомогою стандартних функцій `analogRead()` , `digitalRead()` , `digitalWrite()` та інших.

Плати Arduino дозволяють використовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, у платі Decimila доступно 14 цифрових входів/виходів, 6 із яких можуть генерувати ШІМ сигнал, і 6 аналогових входів.

1.2.3. Різновиди плат розширення

Arduino Sensor Shield – найпопулярніша плата розширення, тому що як правило вона йде в наборах Ардуіно. ШІлд має досить просте завдання - надати більш зручні варіанти підключення до основної плати. Це здійснюється за рахунок додаткових роз'ємів живлення і землі, виведених на плату до кожного з аналогових і цифрових пінів. Також на платі можна знайти роз'єми для підключення зовнішнього джерела живлення (для перемикання потрібно встановити перемички), також є світлодіод і кнопка перезапуску. Варіанти Arduino Sensor Shield зображені на (рис. 1.3).



Рисунок 1.3 – Зовнішній вигляд Arduino Sensor Shield

Існує багато версій сенсорної плати розширення. Вони відрізняються кількістю і видом роз'ємів. Найпопулярнішими на сьогодні є версії Sensor Shield v4 і v5.

Arduino Motor Shield (рис. 1.4) – цей вид плат розширення дуже важливий в робототехнічних проектах, тому що дає можливість підключити до плати Arduino серво двигуни. На платі розширення встановлені всі необхідні регулятори, перемикачі і запобіжники. Загалом, на Motor Shield є все для забезпечення простого керування двигунами і для їх захисту. Основним завданням Arduino Motor Shield - забезпечити керування пристроями які споживають досить високий для звичайної плати Ардуіно струм. Також такий шилд може керувати потужністю двигуна (з допомогою ШІМ) і зміни напрямку обертання. Як правило на в схемі Arduino Motor Shield наявний потужний транзистор, через який підключається зовнішнє навантаження, піни для підключення до Ардуіно, тепловідвідних елемент, схеми для підключення зовнішнього живлення, піни для підключення двигунів.



Рисунок 1.4 – Зовнішній вигляд Arduino Motor Shield

Arduino Ethernet Shield (рис. 1.5) – призначений для забезпечення незалежності ардуіно проекту від персонального комп'ютера, так як він дає

можливість підключити Arduino до локальної мережі через Ethernet. Ще однією перевагою такого щита є наявність на ньому слота для MicroSD карти. Так що якщо є необхідність обробляти великий обсяг інформації, зберігати дані можна саме на SD карті. З використанням Ethernet Shield Можна забезпечити роботу хостингу для веб-сервера.



Рисунок 1.5 – Зовнішній вигляд Arduino Ethernet Shield

Arduino LCD shield – даний тип плат розширення використовується для роботи з LCD-екранами в Ардуіно. Він виводить інформацію з Arduino не на персональний комп'ютер, а безпосередньо на периферійний екран на платі розширення. Також підключення Arduino LCD shield навідрізняє від підключення навіть самого простого 2-сатиричного текстового екрану дозволяє економити кількісь контактів, та як для підключення периферійного текстового екрану потрібно зазвичай 7 і більше контактів, а в Arduino LCD shield використовується протокол передачі даних I2C, тобто для його підключення задіюються лише 2 контакти!

У Arduino LCD shield (рис. 1.6) в додавок до екрану, на плату встановлено 4 "керуючі" кнопки і кнопка "select". що дозволяє організувати і зовнішній інтерфейс для користувача і безпосереднього підключення до ПК можна уникнути.



Рисунок 1.6 – Зовнішній вигляд Arduino LCD shield

Arduino Data Logger Shield (рис. 1.7) – призначений для збереження даних, отриманих з датчиків, з прив'язкою за часом. Така плата розширення дозволяє не лише зберігати дані і отримувати час з вбудованого годинника, але і підключати датчики шляхом пайки або на монтажній платі.



Рисунок 1.7 – Зовнішній вигляд Arduino Data Logger Shield

1.3. Брокер повідомлень Mosquitto

Eclipse Mosquitto - це брокер повідомлень з відкритим кодом (під ліцензією EPL/EDL), що реалізовує протокол MQTT версій 3.1 та 3.1.1. Mosquitto є доволі легкою системою, та внаслідок цього може використовуватись на багатьох видах пристроїв: від простих одноплатних комп'ютерів до повноцінних серверів.

Протокол MQTT забезпечує легкий метод здійснення обміну повідомленнями за допомогою моделі публікації / підписки. Це робить його придатним для обміну повідомленнями Інтернету речей, таких як датчики низької потужності або мобільні пристрої, такі як телефони, вбудовані комп'ютери або мікроконтролери.

Mosquitto також надає бібліотеку на мові програмування C, що дозволяє реалізовувати MQTT-клієнти, а також доволі популярні консольні клієнти: `mosquitto_pub` та `mosquitto_sub`.

1.4. Протоколи мережевого обміну CoAP, AMQP , MQTT

Обмін і передача даних в мережі здійснюється за допомогою прийнятого набору правил і угод. Протоколи визначають тип і формат даних, які передаються, загальні правила взаємодії різних програм, порядок обробки пакетів інформації, створює єдиний простір передачі.

Взаємодія двох терміналів в мережі поділяється на декілька рівнів. До них відносяться транспортний, мережевий, фізичний, прикладної та ін.

Протокол мережевого обміну інформацією – це перелік типів і форматів переданих блоків даних, правила їх обробки і взаємодії термінальних комп'ютерів на одному рівні. Інтерфейсом називають набір правил, що визначають взаємодію сервісів сусідніх рівнів в одному терміналі.

Різні протоколи, узяті разом, утворюють стек протоколів і описують різні сторони одного типу зв'язку. Протоколи реалізують програмне забезпечення на яке вказують назви «протокол» і «стек протоколів».

CoAP - Constrained Application Protocol - це спеціалізований протокол додатків до Інтернету для обмежених пристроїв. Це дозволяє тим обмеженим пристроям, званим "вузлами", спілкуватися з більш широким Інтернетом, використовуючи подібні протоколи. CoAP призначений для використання між пристроями в одній і тій же обмеженій мережі (наприклад, малопотужними, втратними мережами), між пристроями та загальними вузлами в Інтернеті, і між пристроями в різних обмежених мережах, приєднаних до Інтернету. CoAP також використовується за допомогою інших механізмів, таких як SMS у мережах мобільного зв'язку.[0]

Протокол CoAP надає URI для знаходження типів інтернет-медіа для їх опису, а також не зберігаючи стани встановлює відповідності до HTTP дієслів. Протокол сумісний з існуючою IP-інфраструктурою з UDP прив'язкою.

Протокол CoAP призначений для взаємодії простих пристроїв, такими як клапани, вимикачі, датчики якими можна керувати або контролювати їхні покази віддалено через мережу Інтернет. Інформаційний обмін між такими пристроями називається міжмашинна взаємодія (M2M) або пристрої з обмеженими ресурсами. Таку назву вони отримали через обмежений енергоресурс, невеликий обсяг пам'яті і невисоку потужність, тому в роботі з цими пристроями важливо забезпечувати низькі енерговитрати і використовувати передачу повідомлень малого обсягу. CoAP виконує всі необхідні вимоги для забезпечення взаємодії цих пристроїв. Мережа, в якій працюють такі пристрої, називається мережею з обмеженими ресурсами.

Структура протоколу CoAP розроблена відповідно до REST. REST (Representational State Transfer) - це стиль архітектури програмного забезпечення для систем, таких як World Wide Web, який, використовується для побудови веб-служб. REST є дуже простим інтерфейсом керування інформацією без використання додаткових внутрішніх прошарків.

Протокол CoAP має чотири види повідомлень: Confirmable, Non-confirmable, Acknowledgement, Reset. Вони кодуються в бінарному вигляді.

Протокол AMQP: Advanced Message Queuing Protocol

Advanced Message Queuing Protocol (AMQP) - це відкритий протокол, для передачі повідомлень між компонентами системи з низькою затримкою і на досить високій швидкості. AMQP може налаштовується під потреби конкретного проекту і передбачає надійний транспортний протокол, такий як TCP.

Ідея цього протоколу полягає в обміні довільним чином повідомленнями через AMQP-брокер, який здійснює маршрутизацію, гарантує доставку, розподіл потоків даних, підписку на потрібні типи повідомлень.

Протокол AMQP забезпечує комунікацію за допомогою публікації/підписки повідомлення. Головною перевагою AMQP є функція зберігання, а також передачі даних, яка забезпечує надійність навіть якщо в роботі мережі стався збій.

Недоліком протоколу AMQP є низький рівень успішності доставки при низькій пропускній здатності та збільшується при збільшенні пропускної здатності. А перевагою, порівнюючи AMQP з REST, протокол AMQP може відправляти більшу кількість повідомлень у секунду.

В протоколі AMQP, на самому нижньому рівні визначається формат кодування даних в бінарному вигляді, для передачі по TCP-з'єднання, вище формат передачі RPC-запитів між сервером і клієнтом. Семантика роботи протоколу з повідомленнями, описується в XML-специфікації, яка задає RPC-інтерфейс сервера і клієнта. XML є останньою специфікацією протоколу.

AMQP заснований на трьох поняттях:

1. Повідомлення (message) - одиниця переданих даних, ця частина не тлумачиться сервером.
2. Точка обміну (exchange) - в неї відправляються повідомлення і тут розподіляються повідомлення в одну або кілька черг. При цьому в точці обміну повідомлення не зберігаються.
3. Черга (queue) – працює як буфер, тут зберігаються повідомлення до тих пір, поки воно не буде забрано клієнтом з однієї або декількох черг.

Протокол MQTT: Message Queue Telemetry Transport

MQTT або Message Queue Telemetry Transport - це легкий, компактний і відкритий протокол обміну даними створений для передачі даних на віддалених локаціях, де потрібно невеликий розмір коду і є обмеження по пропускній здатності каналу.

Також існує версія протоколу MQTT-SN (MQTT for Sensor Networks), раніше відома як MQTT-S, яка призначена для вбудованих бездротових пристроїв без підтримки TCP / IP мереж, наприклад, Zigbee.

Основними особливостями протоколу повідомлень MQTT є:

- асинхронний протокол
- компактні повідомлення
- Робота в умовах нестабільного зв'язку на лінії передачі даних
- Підтримка декількох рівнів якості обслуговування (QoS)
- Легка інтеграція нових пристроїв

Протокол MQTT працює на прикладному рівні поверх TCP / IP і використовує за замовчуванням 1883 порт.

Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем або підписником (publisher / subscriber) повідомлень, і брокером (broker) повідомлень (наприклад, Mosquitto MQTT).

Видавець відправляє дані на MQTT брокер, вказуючи в повідомленні певну тему, топик (topic). Підписники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки.

Пристрої MQTT використовують певні типи повідомлень для взаємодії з брокером, нижче представлені основні:

Connect - встановити з'єднання з брокером

Disconnect - розірвати з'єднання з брокером

Publish - опублікувати дані в топик на брокера

Subscribe - підписатися на топик на брокера

Unsubscribe - відписатися від топика

Схему взаємодії між підписником, видавцем і брокером можна переглянути на рисунку 1.8.[7]

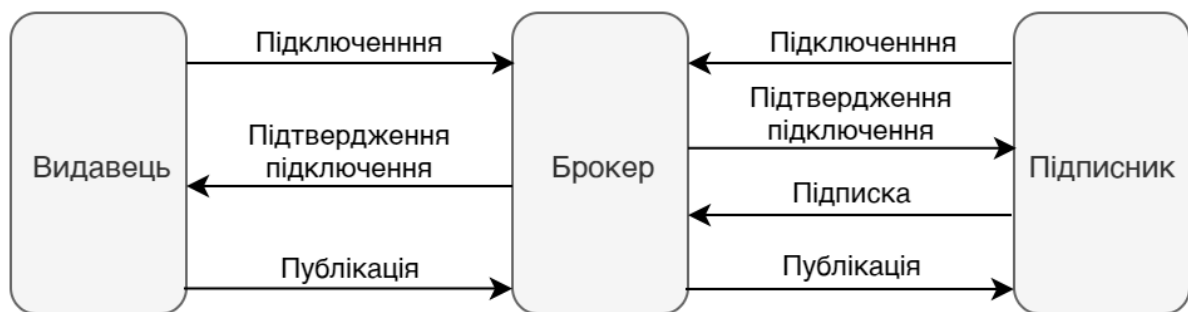


Рисунок 1.8 – Схема взаємодії між підписником, видавцем і брокером

Видавець відправляє дані на MQTT брокер, вказуючи в повідомленні певну тему, топик (topic). Підписники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки.

РОЗДІЛ 2. ТЕХНОЛОГІЧНА ЧАСТИНА

2.1. Підбір моделі сервопривода.

2.1.1. Поняття сервопривода і його будова

Сервопривід – вид приводу, вал якого можна повернути на певний кут і утримувати дане положення або підтримувати обертання з заданим періодом. Керування приводу базується на використанні негативного зв'язку де сигнал на вході і виході в контурі із замкнутою схемою не узгоджений з одним або в деяких випадках кількома системними сигналами. Такий принцип дозволяє керувати параметрами руху. В складі сервопривода є датчик і блок керування приводом, який підтримує всі необхідні параметри на давачі відповідно до заданого значення керуючої системи.

Найбільш розповсюдженими серводвигунами вважаються ті що підтримують сталу швидкість обертання вала і утримують заданий кут повороту.

Будова сервопривода зображена на рисунку нижче (рис. 2.1).

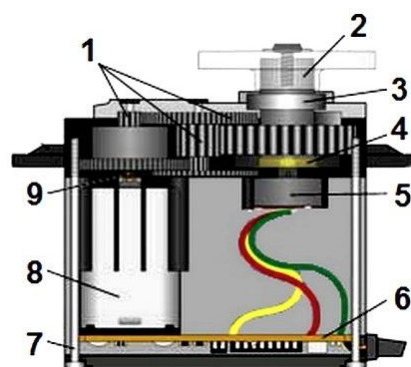


Рисунок 2.1 – Будова сервопривода

1. Редукторні шестерні
2. Вал
3. Шариковий підшипник

4. Втулка
5. Потенціометр
6. Плата керування
7. Корпусний гвинт
8. Сервомотор
9. Шестерня мотора

Електромотор – це пристрій який необхідний щоб перетворити електричну енергію в механічний поворот вихідного валу. Зазвичай швидкість обертання електромотора занадто велика для його практичного використання. Тому для зниження швидкості використовують редуктор.

Потенціометр використовують для зміни поточного стану механізму. Зміна опору проходить при зміні кута повороту бігунка потенціометра, яка пропорційна куту повороту вихідного валу.

2.1.2. Внутрішній інтерфейс керуючих сигналів. Керування сервоприводом

Для зміни положення сервопривода необхідно надсилати по приводу на нього керуючий сигнал. Тобто імпульси сталої частоти але змінної ширини. Важливим параметром є довжина імпульсу, так як визначає положення сервопривода. Задається довжина імпульсу вручну в програмі підбираючи кут повороту або за використанням бібліотеки. Коли на керуючу схему надходить сигнал, імпульсний генератор виробляє власний імпульс а потенціометр визначає його тривалість. Далі тривалість імпульсів порівнюється і якщо вона різна, то включається мотор. Напрямок обертання мотора залежить від того, який з двох імпульсів довший, а якщо вони рівні, мотор зупиняється.

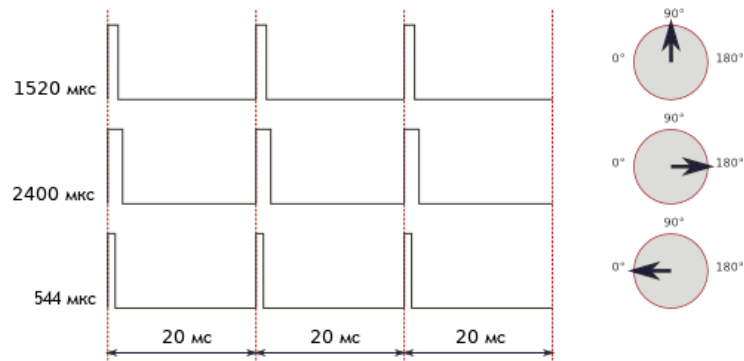


Рисунок 2.2 – Схема залежності положення сервопривода від довжини імпульсів

При частоті імпульсів в 50 Гц, імпульс буде випускатися 1 раз в 20 мілісекунд. При цих значеннях тривалість імпульсу становить 1520 мкс і сервопривід займає положення в 90° . Змінюючи довжину імпульсу міняється кут повороту. В бібліотеці Ардуіно є встановлені межі значення довжини імпульсу при 0° - 544 мікросекунд (нижня межа), при 180° - приблизно 2400 мікросекунд (верхня межа).

Варто звернути увагу, що іноді заводські налаштування відрізняються від прийнятих стандартних налаштувань. На деяких сервоприводах середнім положенням є тривалість імпульсу в 760 мікросекунд.

Іноді спосіб керування сервоприводами називають ШІМ. Но це не зовсім правильно, оскільки для його керування важлива довжина імпульсів і не так важлива частота. Частота 50 Гц вважається нормою, але сервопривід буде працювати і при невеликому їх відхиленні. При великому зниженні частоти, сервопривід буде працювати з ривками, а при великому (наприклад, 150 Гц) сервопривід може перегрітися, а в деяких випадках вийти з ладу.

За типом сервоприводи поділяють на аналогові і цифрові. Зовні немає жодних відмінностей, про те вони відрізняються вбудованою керуючою електронікою. В аналогового сервопривода є мікросхема, а у цифрового мікропроцесор, який керує мотором. Тобто відмінність між ними лише в способі обробки вхідних імпульсів

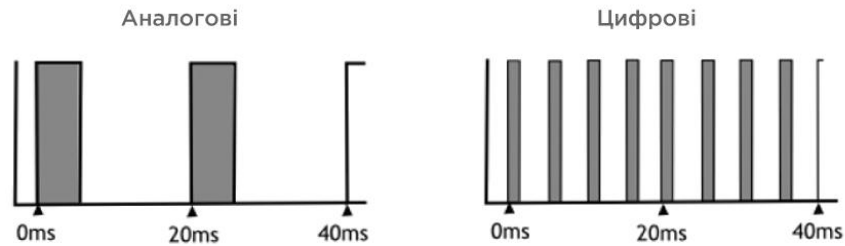


Рисунок 2.3 – Ширина імпульсів в аналогових і цифрових сервоприводах

Обидва типи сервоприводів приймають однакові вхідні керуючі імпульси. Цифровий сервопривід завдяки використанню мікропроцесора який посилає сигнал на електромотор з частотою 200 Гц, швидше реагує на зовнішні впливи, розвиває необхідну швидкість і крутний момент, і значно краще утримує задану позицію. Проте споживає менше електроенергії і простіший у виробництві, а тому коштує дешевше.

Шестерні та їх матеріал виготовлення

Виготовляють шестерні для сервоприводів з: металу, пластику, карбону. Їх вибір залежить від характеристик які потрібні в установці, і від конкретного поставленого завдання. Варіанти виконання шестерень з металу і пластику зображений на (рис. 2.4).

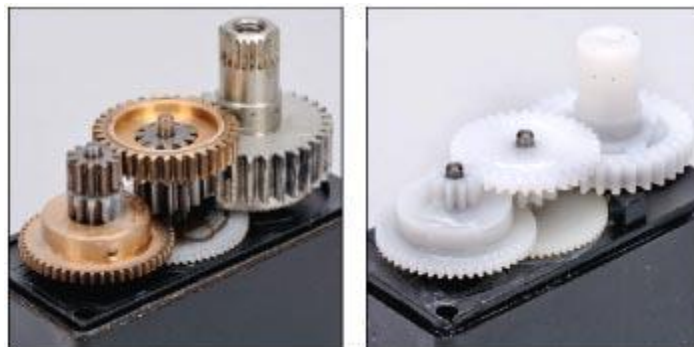


Рисунок 2.4 – Шестерні сервопривода

Пластикові (нейлонові) шестерні чудовий вибір якщо не передбачаються великі навантаження на сервопривід, оскільки вони схильні

до швидкого зносу. Про те вони дуже легкі і найпоширеніші у використання через свою невелику ціну.

Для більш довговічного використання, використовують сервоприводи з карбоновими шестернями, їх недоліком є велика ціна на них. Металеві шестерні витримують великі навантаження, хоча також швидко зношуються.

Підключення сервоприводів до Arduino

На будь-якому сервоприводі є три проводи, які пофарбовані в спеціальні кольори.

- Коричневий або чорний провід - підключається до землі
- жовтий або білий провід – це сигнал, підключається до цифрового піна плати Ардуіно.
- Червоний провід – це живлення, підключається до контакту +5 В або до джерела живлення.

Для підключення сервопривода до плати Ардуіно можна скористатися спеціальною платою розширення або під'єднавши проводи сервоприводів безпосередньо до пінів Ардуіно. Стандартну схему підключення бачимо на (рис. 2.5).

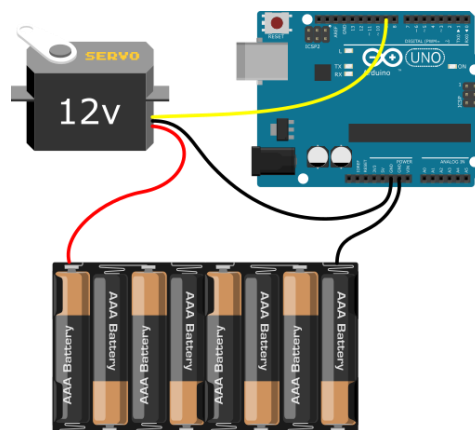


Рисунок 2.5 – Схема підключення сервопривода

Характеристики сервоприводів

Основними характеристиками сервоприводів є: розмір, вага, швидкість, кут повороту, момент на валу.

Розділяють 3 класи розмірів: "маленькі", "стандартні", "великі". Розміри і вага кожного класу можуть дещо змінюватися, переглянути можна на таблиці 1.

Таблиця 1 – Класифікація сервоприводів за розміром і вагою

Клас	Вага	Лінійні розміри
Маленькі	8-25 г	22 × 15 × 25 мм
Стандартні	40-80 г	40 × 20 × 37 мм
Великі	50-90 г	49 × 25 × 40 мм

Швидкість сервопривода це час за який привід робить поворот на кут 60° при поданій на нього напрузі в 4.8В і 6В. Наприклад, сервопривід з вказаним параметром 0.26с / 60° при 6В робить поворот валу на 60 градусів за 0.26с при поданій на нього напрузі живлення 6В. Це дещо посередній результат. Деякі сервоприводи здійснюють таке переміщення за час від 0.06 до 0.09с.

Прийнято вважати що кут повороту сервопривода 180 градусів, но це не так. Сервоприводи, залежно від моделі, можуть повертатись на 60, 90, 180, 270, 360 градусів. Кут повороту таких сервоприводів обмежується внутрішньою електронікою і механікою. Для зміни робочого діапазону сервопривода потрібно змінювати його внутрішню конструкцію.

Для розробки різних проектів, зокрема руки маніпулятора, важливою характеристикою є крутний момент.

Момент сили, або крутний момент - векторна фізична величина, що дорівнює добутку радіус-вектора, проведеного від осі обертання до точки прикладання сили, на вектор цієї сили. Характеризує обертальну дію сили на тверде тіло.[3]

Тобто, момент сили (рис. 2.6) це вага вантажу, яка вимірюється в кілограмах, який сервопривід утримує нерухомо на гойдалці з довжиною плеча 1 см. Ці цифри вказують для вхідної напруги в 4.8В або 6В.

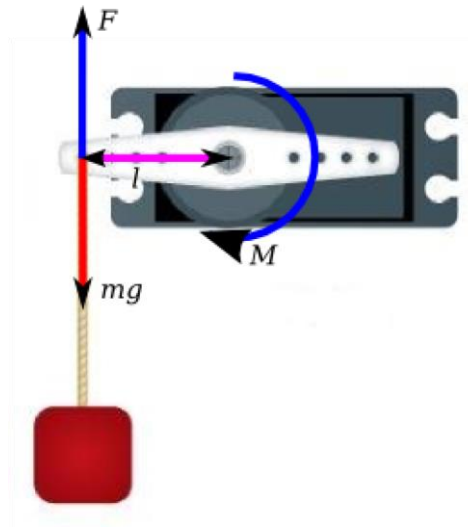


Рисунок 2.6 – Момент сили сервопривода

- l – плече гойдалки;
- M – крутний момент;
- mg – сила яка задається навантаженням;
- F – сила протидії сервопривода;

2.1.3. Вибір сервопривода

Для розробки руки маніпулятора, розглянемо 3 моделі сервоприводів, які найбільше задовільняють наші потреби.

Першим варіантом є сервопривід DS04-NFC

Загальні відомості

Привід постійного обертання DS04-NFC (рис. 2.7) - привід з кутом обертання 360° і зусиллям $5.5 \text{ кг}\cdot\text{см}$.



Рисунок 2.7 – сервопривід DS04-NFC

Може бути використаний при побудові роботів, різних механізмів, де не потрібно відстежувати кут повороту і потрібен привід, здатний здійснювати повний оборот.

Характеристики

- Живлення: 4.8-6В;
- Струм: до 1000мА;
- Діапазон обертання: 360°;
- Швидкість обертання без навантаження при 4,8В: 60° за 0,22с;
- Швидкість обертання без навантаження при 6В: 60° за 0,18с
- Момент сили при напрузі 4.8В: 5.5 кг·см;
- Габаритні розміри: 41x20x40мм;
- Вага: 38г;

Підключення

Чорний провід - GND;

Червоний провід - 5V;

Білий провід - Signal, підключається до будь-якого цифрового виходу плати Arduino;

Детальніше про привід

Основна відмінність приводу постійного обертання від сервоприводу в тому, що він обертається нескінченно довго в тому напрямку, який йому задано, а так само в тому, що привід не має потенціометра або оптопари, отже програмно можливо задати тільки напрямок і швидкість обертання приводу.

З точки зору написання скетчів даними приводом можна керувати за допомогою стандартних бібліотек, що входять до складу Arduino IDE. Основною ж відмінністю є те, що використовується функція `Servo.writeMicroseconds` (ШИРИНА_ІМПУЛЬСУ), яка задає не кут, а напрямок і швидкість обертання приводу:

Сервопривід MG90S

Загальні відомості

Мікросервопривід MG-90S (рис. 2.8) (з металевими шестернями) - аналоговий серводвигун, який використовується для управління невеликими легкими механізмами, кут повороту яких обмежений діапазоном від 0 до 180 градусів.



Рисунок 2.8 - Сервопривод MG90S

Характеристики

- Напруга: 4.8 - 6В;
- Живлення 4.8В: 1.8кг·см;
- Живлення 6В: 2.2 кг·см;
- Час повороту на 60°:
 - Живлення 4.8В: 0.10с;
 - Живлення 6В: 0.08с;
- Матеріал шестерень: Латунь, алюмінієвий сплав;
- Підшипники вихідного валу: 1шт, зверху;
- Електроніка: Аналогова;
- Довжина кабелю: 0.26 м;
- Вага: 13.4гр;
- Розміри: 22.8 x 12.2 x 28.5мм;

Підключення

Коричневий провід - GND;

Червоний дріт - 5V;

Помаранчевий провід - Signal, підключається до будь-якого цифрового виходу Arduino;

Сервопривід MG996R-180 (з підвищеним крутним моментом)

Сервопривід MG996R-180 (рис. 2.9) це покращена версія популярного цифрового сервопривода, моделі MG995R, по точності і швидкості позиціонування. Керування здійснюється на аналоговій мікросхемі, а не як у оригіналі на цифровому мікроконтролері. Що значно знижує ціну сервопривода. Порівняно з попередніми моделями має підвищений крутний момент. Привід здійснений з металу, а перероблена плата і система керування робить сервомотор більш точним. Саме тому ця модель часто використовується для радіокерованих літаків, планерів, гелікоптерів, автомобілів, катерів, шасі роботів та інших. Завдяки редукторі з металевими шестернями на вихідному валу розвивають зусилля якого достатню для застосування в рухомих роботів. Дворядний шарико подібний підшипник і металеві шестерні забезпечують стабільну і надійну роботу приводу без пошкоджень.

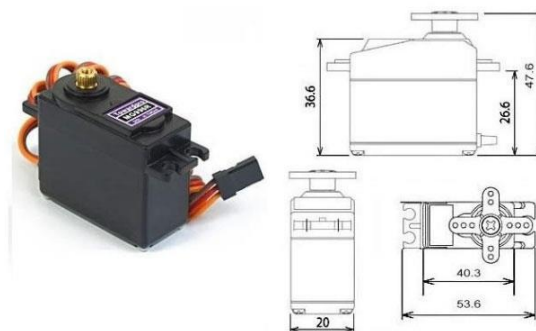


Рисунок 2.9 – Зовнішній вигляд сервопривода MG995R

Характеристики сервопривода:

- Маса: 55 грам;
- Розміри: 40.7 x 19.7 x 42.9;
- Крутний момент:

- 8.5 кг х см (при 4.8 В);
- 10 кг х см (при 6 В);
- Швидкість: 0.2 с / 60° (при 4.8 В), 0.16 с / 60° (при 6 В);
- Робоче живлення: 4.8 - 7.2 В;
- Ширина мертвої зони: 5 мкс;
- Діапазон робочих температур: 0 °С - 55 °С.

Підключення

Чорний провід – GND (земля);

Червоний провід – 5V (живлення);

Білий провід – Signal (підключається до будь-якого цифрового виходу плати Arduino).

Розглянувши ці три моделі сервоприводів, порівнюючи їхні характеристики, робимо висновок що для конструювання робота маніпулятора найбільше підійде сервопривід MG996R. В даній моделі, привід здійснений з металу, а також дворядний шарикопідшипник робить сервопривід більш надійним довговічним і захищеним від пошкоджень. Перероблена плата і система керування робить сервопривід більш точним порівняно з попередніми моделями, а аналогова мікросхема значно знижує ціну сервопривода.

2.2. Вибір макета робота маніпулятора

Для реалізації проекту необхідно вибрати макет робота маніпулятора яким буде здійснене керування через брокер за допомогою мережі WiFi.

На даний момент є велика кількість роботів маніпуляторів різних конструкцій і форм, розглянемо 3 варіанти робота маніпулятора.

На першому варіанті (рис 2.10) бачимо макет робота виконаного з дерева за допомогою досить точної лазерної різки. В даній моделі всі механічні частини досить універсальними. Маніпулятор має 4 рівні свободи.

Маса робота 230г що дозволяє використати дешеві сервоприводи. В задній частині є місце для розміщення плати Arduino. Дана модель має дуже низькі конструктивні і механічні характеристики, а також мала робоча область.

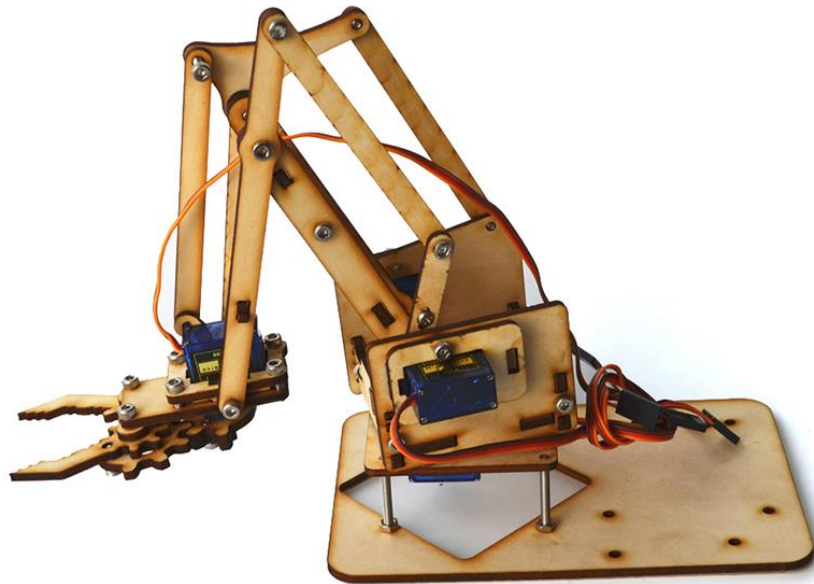


Рисунок 2.10 – Вигляд маніпулятора SNAM1500

На другому варіанті бачимо маніпулятор виконаний з акрилу (рис.2.11).



Рисунок 2.11 – Вигляд маніпулятора виготовленого з акрилу

Це класична машина на 4 ступені свободи, яка існує вже протягом тривалого часу і є натхненням від промислово роботизованого маніпулятора. Це економічний, потужний і дуже простий у використанні набір для роботизованого навчання на базі arduino. В цій можелі досить простий спосіб керування. Чотири ручки використовуються для керування рухом потенціометра. Після підключення робота до комп'ютера через USB роз'єм можна відправляти команди через послідовний порт для керування кута повороту сервопривода. Ця модель сумісна з іншим програмним забезпеченням для керування нею через ПК, а також підтримує програми для виконання фіксованих і повторних дій.

Цю модель легко збирати так як її не потрібно паяти, немає додаткового обладнання і інструментів, а також плата по замовчуванню прикріплена до основи.

- Система тяг, дозволяє розмістити потужні сервоприводи, а також утримує захват паралельно або перпендикулярно основі маніпулятора;
- Простий набір комплектуючих;
- Підшипники є у більшості вузлах маніпулятора
- Простота складання.
- Положення захвату можна міняти на 90 градусів
- Arduino-сумісність.

Третім варіантом є маніпулятор виконаний з алюмінієвих пластин (рис.2.12). Дана модель має 5 рівнів свободи. Всі механічні частини робота є універсальними і можуть бути використані в залежності від мети яку маніпулятор буде виконувати в подальшій експлуатації.

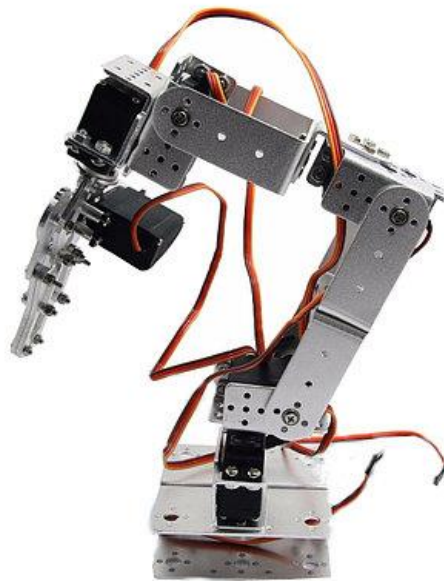


Рисунок 2.12 – Зовнішній вигляд маніпулятора виготовленого з алюмінієвих пластин

Розглянувши всі макети роботів маніпуляторів, для реалізації проекту вибираємо третій варіант. Далі розглянемо більш детально його технічні характеристики.

Даний маніпулятор може бути розцінений як демонстраційна платформа з п'ятьма рівнями свободи. Силова система маніпулятора складається з п'яти сервоприводів які можуть здійснити демонстрацію захоплення, руху вліво, вправо, вверх і вниз. Для стабільності маніпулятора, всі матеріали виготовлені з алюмінієвої пластини товщиною 2 мм. У всіх з'єднаннях використовується високоякісний підшипник для кращої продуктивності.

Перевагою цієї моделі також є великий робочий діапазон. Схематичне зображення робочого діапазону маніпулятора зображено на (рис. 2.13).

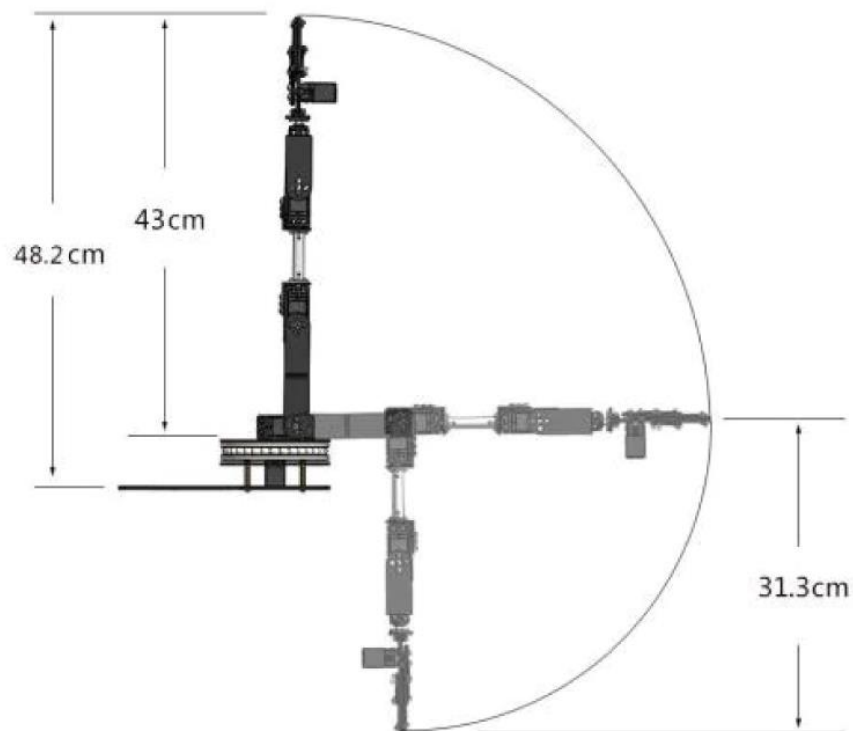


Рисунок 2.13 – Зображення робочого діапазону маніпулятора

В маніпуляторі використовується металічний захват (рис. 2.15) виконаний з твердого алюмінієвого сплаву. До даного типу захвату підходять сервоприводи наступних моделей: HS322 / HS422 / MG995/ MG996R / MG996 / MG996R-180 / MG946 / SG5010.

Характеристики захвату для маніпулятора:

- Маса захвату (без сервопривода) – 68г;
- Максимальний крок відкриття лапи – 50мм;
- Довжина захвату в закритому стані – 118 мм;
- Довжина захвату в відкритому стані – 107 мм;
- Ширина захвату в закритому стані – 55 мм.



Рисунок 2.14 – Металічний захват для маніпулятора

РОЗДІЛ 3. КОНСТРУКТОРСЬКА ЧАСТИНА

3.1. Робот маніпулятор на Arduino

3.1.1. Загальний опис проекту

В проекті робота маніпулятора використано 5 сервоприводів. Для механічної частини я використав набір «5 DOF робот маніпулятор» виконаний з алюмінієвого сплаву товщиною 2 міліметри. Для керування роботом використовуються мікроконтролери Arduino UNO і ESP8266.

3.1.2. Основні вузли для виконання проекту робота маніпулятора

Для збірки маніпулятора нам знадобляться:

- 5 сервоприводів марки MG996R (в цій моделі підвищений крутний момент, високий термін служби);
- Набір металевих кронштейнів для складання робота. Повний їх склад див. в таблиці 2;
- Робоча підставка для робота;
- Блок живлення;
- Плата Arduino UNO;
- Мікроконтролер ESP8266 Node MCU v3;
- Плата розширення Arduino Motor Shield;
- Виключачель 220В;
- Набір проводів;
- Wi Fi антена;
- Кабелі різних типів а також деякі базові інструменти.

Таблиця 2 – Набір комплектуючих для руки маніпулятора

Кількість	Назва комплектуючих
4	багатофункціональний кронштейн сервоприводу
2	довгий u-подібний сервокронштейн
2	l-подібний сервокронштейн
1	U-кронштейн

Продовження таблиці 2

2	U-подібна балка Кронштейн
3	Кулькові радіальні підшипники
1	Алюмінієвий затиск кіготь
6	металевих рогів
1	фланцевий стрижень
3	подовжувач сервоприводу
1	намотувальний ремінь
1	набір гвинтів і гайок

3.2.Збір макету робота маніпулятора

Маючи всі комплектуючі можемо перейти до конструювання. Їх повний набір зображений на рисунку 3.1.

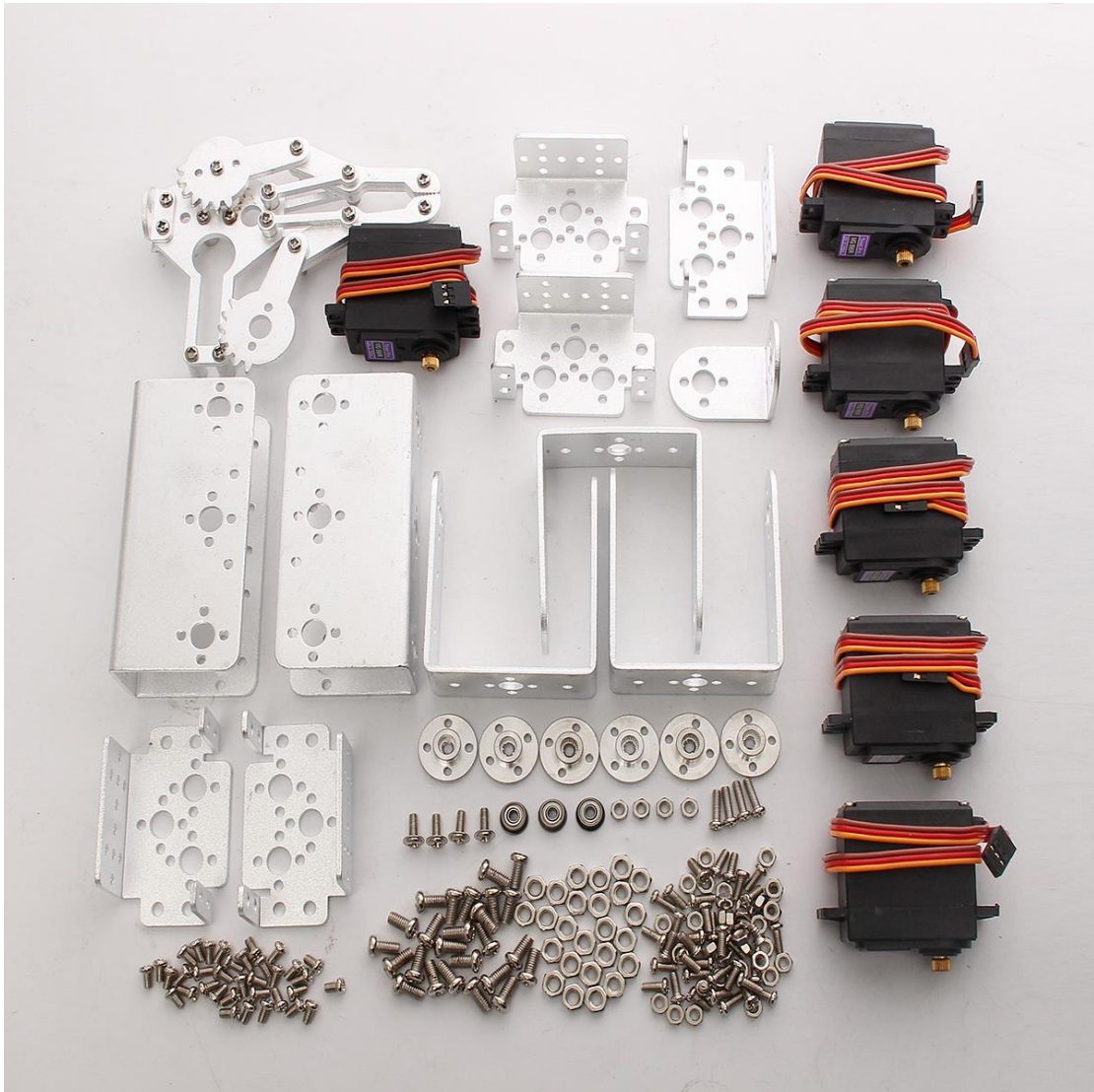


Рисунок 3.1 – Набір комплектуючих робота маніпулятора

Для початку необхідно зібрати маніпулятор. Для цього використовуємо комплектуючі з таблиці 2 і сервоприводи MG996R. Складання маніпулятора проводимо згідно із інструкцією. Особливу увагу хочу звернути на збір захвату для маніпулятора.

Для збирання захвату необхідно трішки вкоротити (рис 3.2) гойдалку від сервопривода до таких розмірів, щоб вона підійшла до захвату, і закрутити її за допомогою болтів.

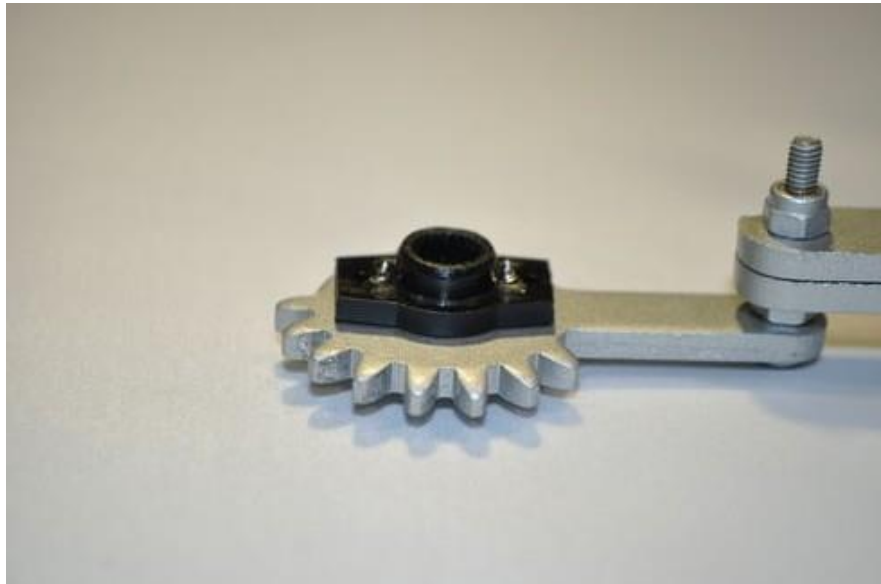


Рисунок 3.2 – Прикріплення гойдалки до сервопривода

Після встановлення сервопривода, повертаємо захват в крайнє положення попередньо стиснувши її (рис. 3.3). Тепер можна закрутити сервопривід на чотири болти тримаючи його при цьому в крайньому положенні і з стисненим захватом. Для перевірки працездатності, підключаємо до плати Arduino.



Рисунок 3.3 – Збирання захвату для маніпулятора

Зібрану руку маніпулятор бачимо на (рис. 3.4).

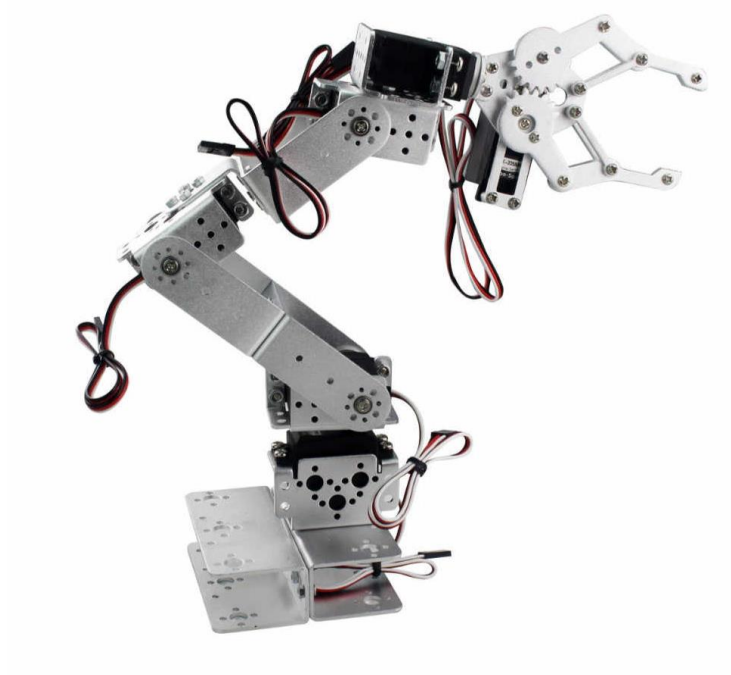


Рисунок 3.4 – Робот маніпулятор

Після того як робот зібраний, прикріплюємо його до робочої підставки. В якості підставки я обрав старий навчальний стенд. Прикріплюємо робота посередині підставки.

Позаду робота можна прикріпити блок живлення. В якості блоку живлення я обрав Gametax GM-400 8CM (рис. 3.5). Це чудовий вибір для офісних ПК або ігрових комп'ютерів початкового рівня. Потужність БЖ складає 400 Вт. Напруга на лініях 3,3 В і 5 В. Для охолодження силових вузлів застосований вентилятор діаметром 120-мм.



Рисунок 3.5 – Блок живлення

Наступним етапом збірки буде збирання електричної схеми всіх комплектуючих.

Оскільки в середині блока живлення багато вільного місця, всі електричні комплектуючі будуть розміщатися там. Спочатку прикріпимо плату Arduino UNO і ESP8266 Node MCU v3 на кришку блока живлення (рис. 3.6). Для цього відмічаємо і просвердлюємо дирки в кришці. Прикріплюємо плати за допомогою болтів з гайками. Оскільки в середині металевого корпусу блока живлення буде послаблюватися WiFi сигнал до плати ESP8266, виведемо з корпусу антену для кращого сигналу.

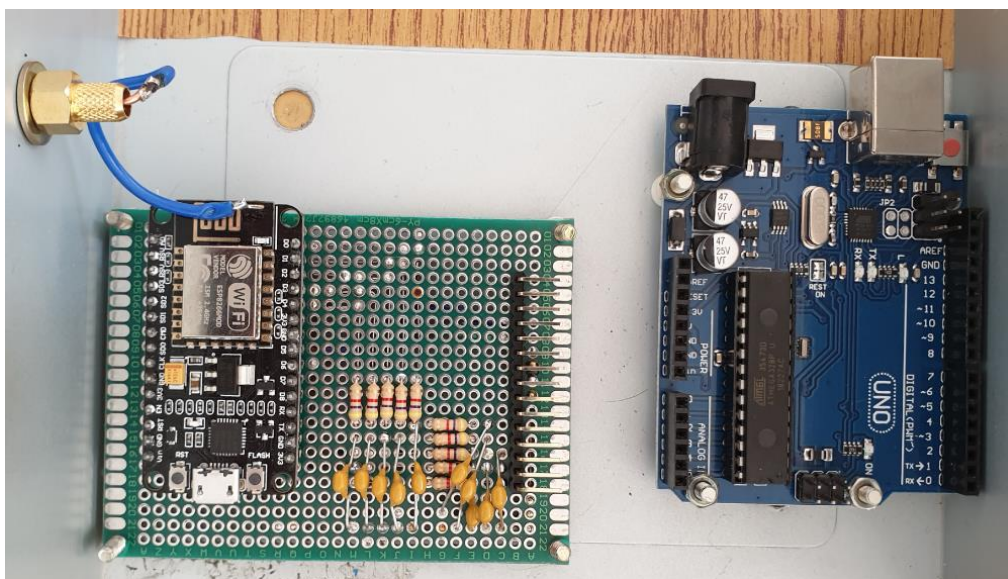


Рисунок 3.6 – Прикріплені плати на кришці БЖ

Поверх Arduino UNO встановлюємо плату розширення shield sensor v4. До цієї плати розширення під'єднаємо сервоприводи і мікроконтролер ESP8266 Node MCU v3 (рис. 3.7). Виходи від сервоприводів підключаються до пінів з широтно імпульсною модуляцією плати Arduino. Використовуючи умовні позначення на платі і сервоприводах, під'єднаємо сервоприводи наступним чином

- Чорний провід до піна «G» - земля;
- Червоний провід до піна «V» - 5В;
- Помаранчевий провід до піна «S» - сигнал.

5 виходів ESP8266 Node MCU v3 під'єднуємо з аналоговими виходами на платі розширення. Ще 1 вихід до цифрового виходу. Він буде служити в якості включателя для ESP8266. При цьому не забуваємо в програмі коректно вказати назви пінів, які ми використовували під'єднуючи плати.

Червоний і чорний проводи від блоку живлення і плати ESP8266 також під'єднуємо до shield sensor v4.

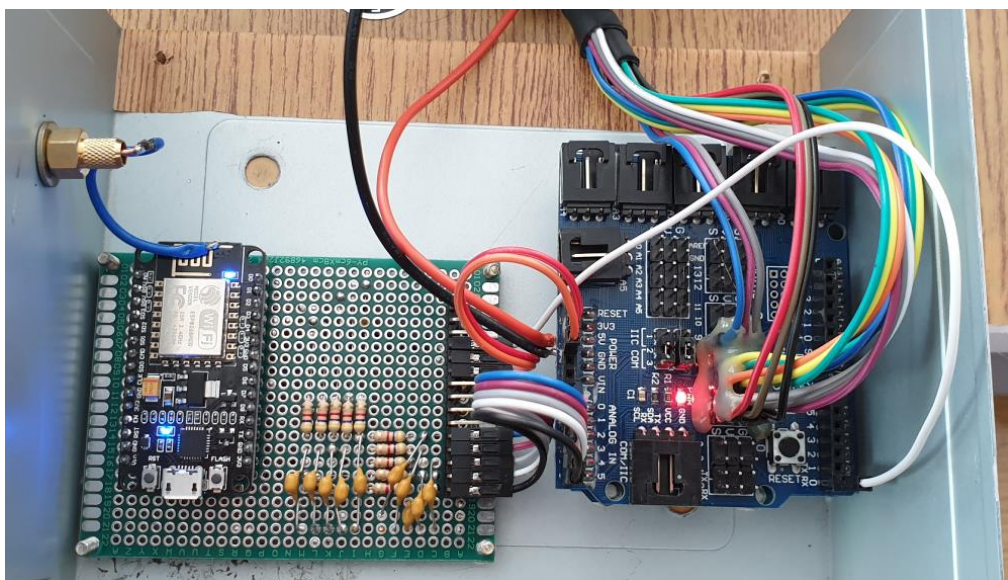


Рисунок 3.7 – під'єднання всіх комплектуючих

Таблиця 3 – Під'єднання пінів плати ESP8266 до плати розширення

ESP8266 Node MCU v3	Shield sensor v4
D0	1 (ANALOG IN)

Продовження таблиці 3

D1	2 (ANALOG IN)
D2	3 (ANALOG IN)
D3	4 (ANALOG IN)
D4	5 (ANALOG IN)
D5	0 (DIGITAL IN)
G	GND
VIN	5V

Повну електрично схему можна побачити на (рис. 3.8).

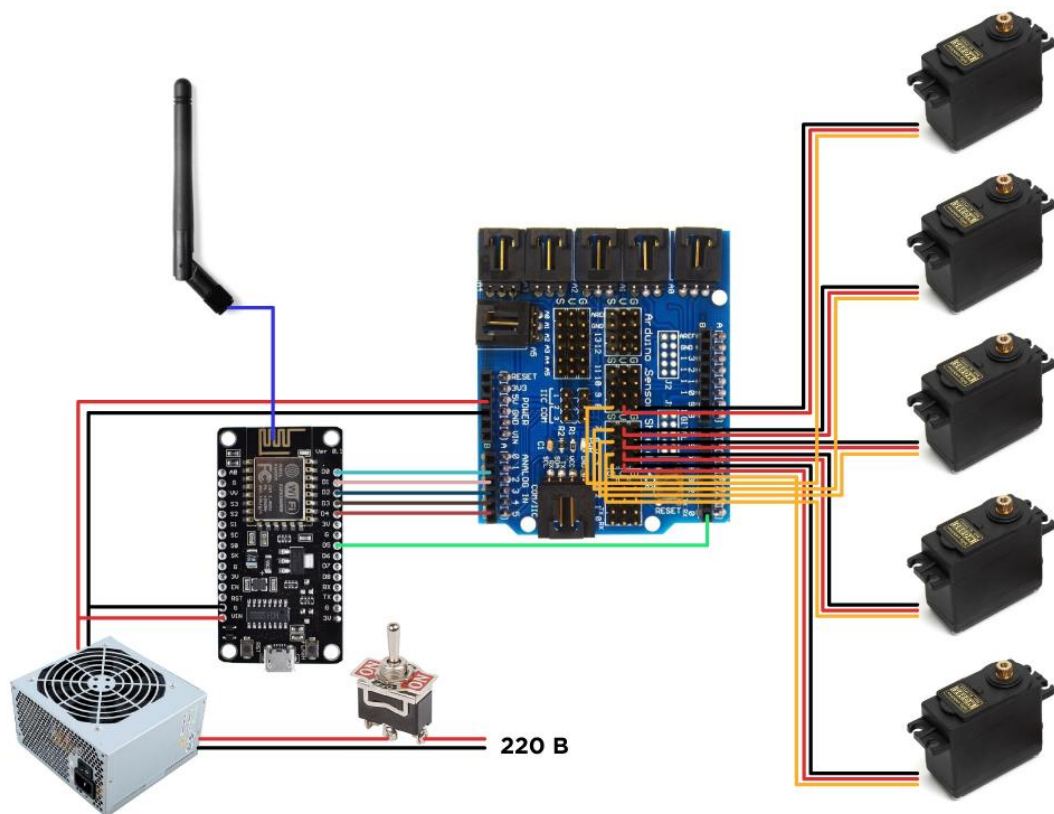


Рисунок 3.8 – Схема підключення

Перед встановленням кришки на місце необхідно прошити плати Arduino UNO і ESP8266 Node MCU v3. Плату Arduino UNO прошиваємо програмою яка буде при поданні живлення на плату задавати початкове положення руки маніпулятора. А на плату ESP8266 Node MCU v3 буде

прошита програма яка буде получати дані з брокера і передавати їх на Arduino UNO.

Перед тим як підключити плату Arduino через usb кабель до персонального комп'ютера, відключаємо живлення. Під час тесту програми, також відключаємо живлення руки. Навіть коли живлення вимкнуте, Arduino біде отримувати струм 5 В від usb. Відповідно, потужність від ПК яка передається через usb перейде на джерело живлення і він трішки "просяде".

Після прошиття плат, встановлюємо кришку на блок живлення. Для зручності в користуванні роботом встановлюємо виключатель «тумблер», який буде розривати подання напруги на блок живлення з мережі.

Робот маніпулятор в зібраному вигляді зображений на рисунку 3.9.

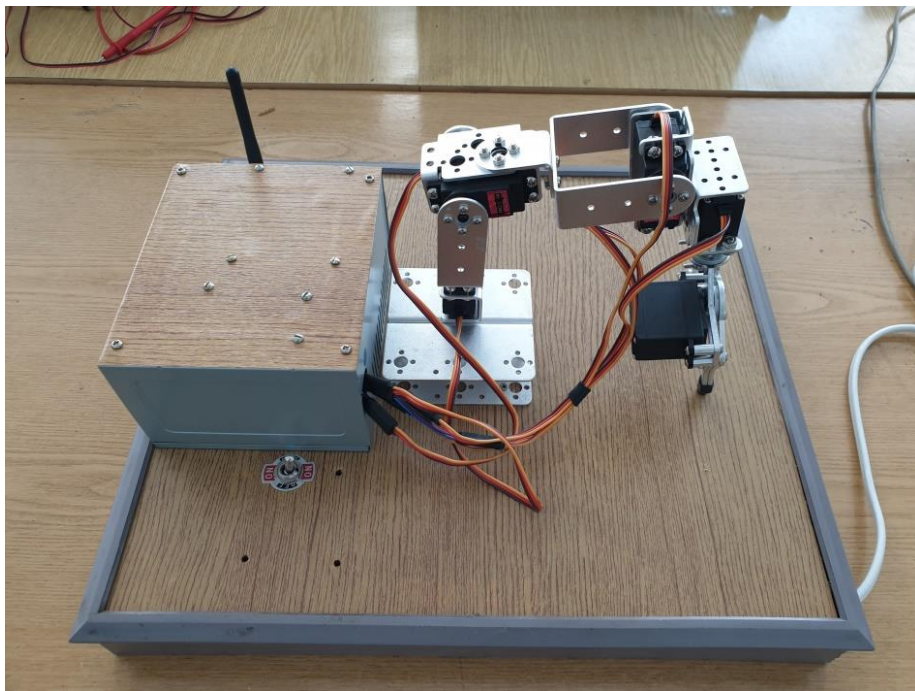


Рисунок 3.9 – Зображення повністю зібраного маніпулятора

3.3. Алгоритм роботи маніпулятора

Зробимо більш детальний огляд роботи сконструйованого нами робота руки маніпулятора. На рисунку 3.10 бачимо алгоритм роботи у вигляді блок-схеми.

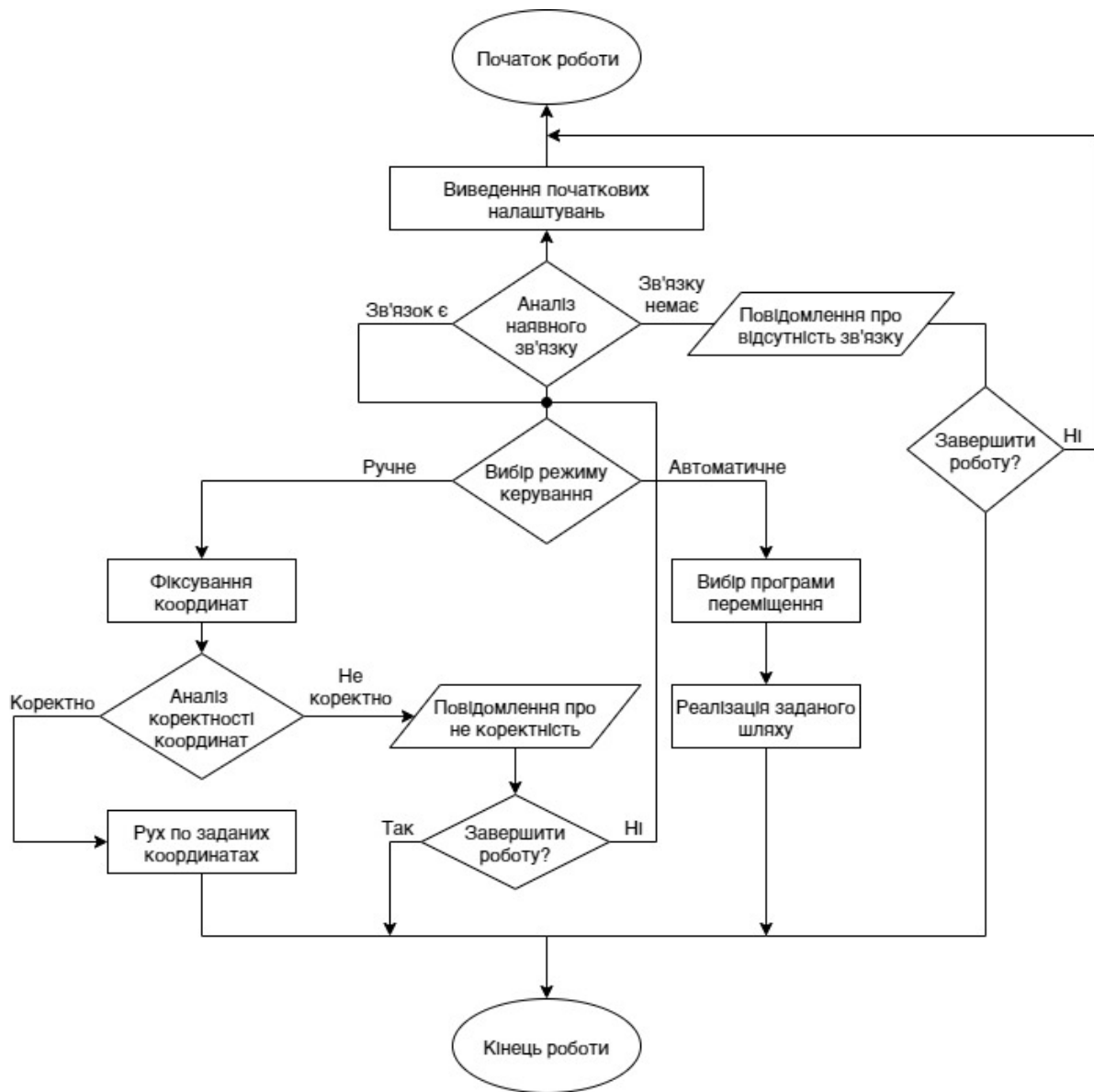


Рисунок 3.10 – Зображення блок схеми з алгоритмом роботи руки маніпулятора

Як бачимо на рисунку 3.10 початок роботи робота маніпулятора починається з включення, вводу початкових налаштувань і самотестування робота. Після завершення налаштувань проводиться аналіз наявного зв'язку. Якщо зв'язку немає, проводиться повторний аналіз або завершення роботи маніпулятора. При наявності зв'язку обирається режим керування: автоматичний або ручний. Автоматичний режим це рух частин робота по певній траєкторії, як попередньо запрограмована в прошивці програмним кодом. Вибираючи ручне керування, відбувається фіксація координат та їх

аналіз коректності. При коректно заданих координатах, маніпулятор виконує заданий рух, якщо координати не коректні, програма завершує роботу робота, або повертається до початкового етапу.

3.4. Передача даних по протоколу MQTT

Прийом і передача даних від модуля до брокера здійснюється за допомогою інтернет мережі WIFI по протоколу повідомлень MQTT. Функціональна схема підключення пристроїв до контролера показана на рисунку 3.11.

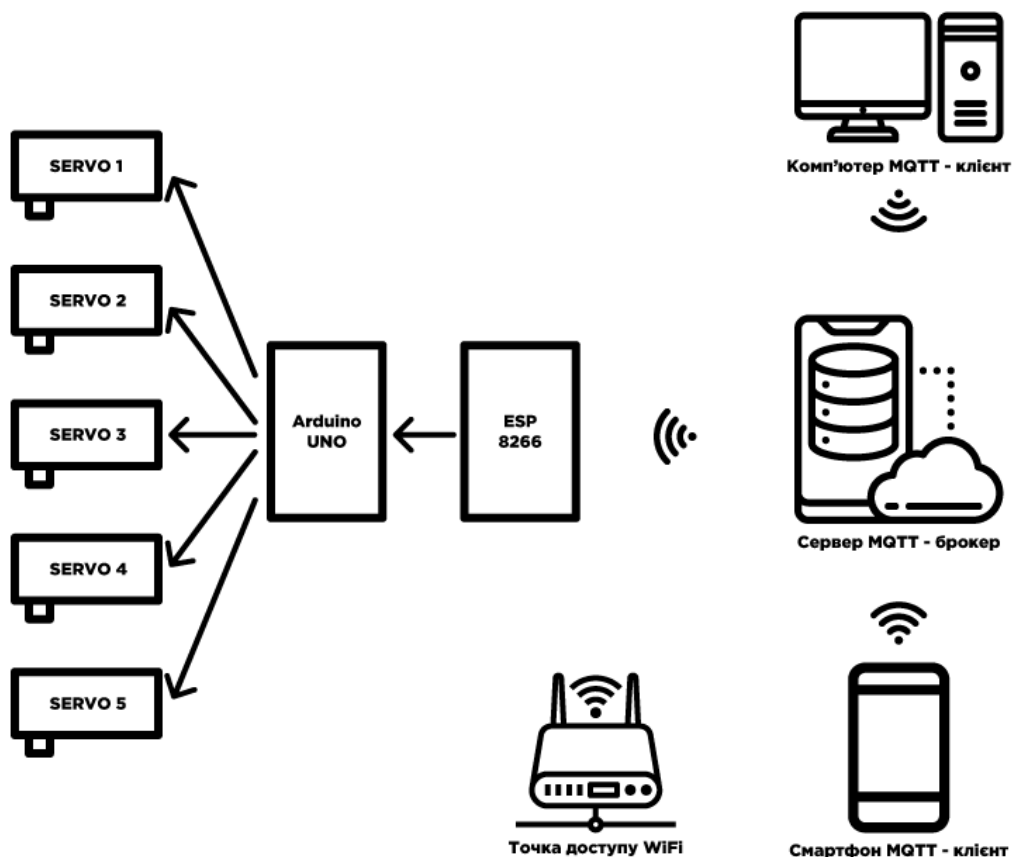


Рисунок 3.11 – Функціональна схема підключення пристроїв до контролера

Перелік основних функції, які виконує модуль:

- Ініціалізація пристрою;
- Прийом повідомлень від брокера;
- Подача сигналу на виходи модуля, який керує сервоприводами;
- Читання даних;

- Передача даних на брокер;
- Періодичне інформування брокера про стан функціонування пристроїв.

3.5. Алгоритм роботи mqtt протоколу

При подачі живлення відбувається ініціалізація пристрою. На даному етапі, також відбувається підключення модуля до доступної WiFi мережі, і запускається mqtt – клієнт, який здійснює відправку стану виходів (інтервал часу відправки по замовчуванню 2с), інтервал можна змінити в залежності від поставленого завдання.

Коли від брокера прийшла команда, відбувається її виконання, а пристрій стає в режим очікування нової команди. Якщо команда не прийшла, пристрій продовжує знаходитися в попередньому режимі. Алгоритм, у вигляді блок-схеми представлений на рисунку 3.12.

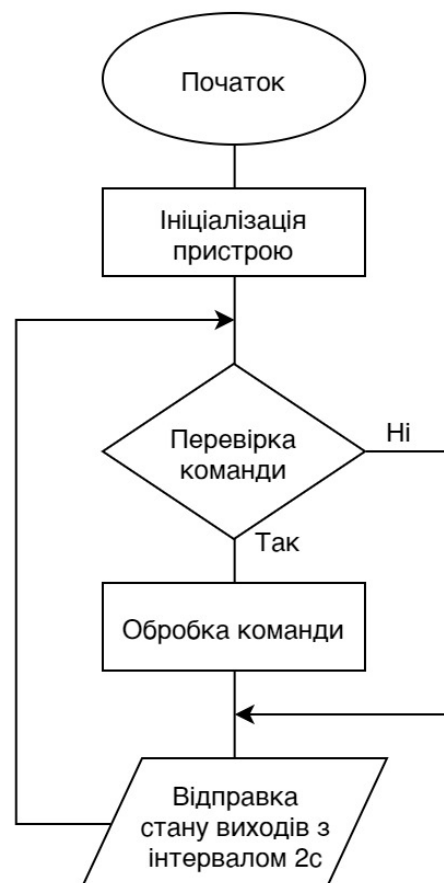
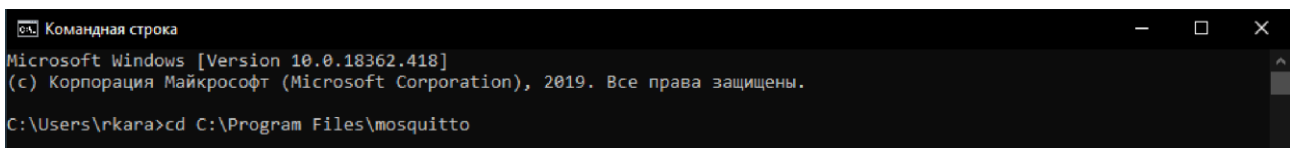


Рисунок 3.12 – Блок схема з алгоритмом роботи модуля

РОЗДІЛ 4. СПЕЦІАЛЬНА ЧАСТИНА

4.1. Запуск, налаштування і передача повідомлень через Mosquitto broker на операційній системі Windows

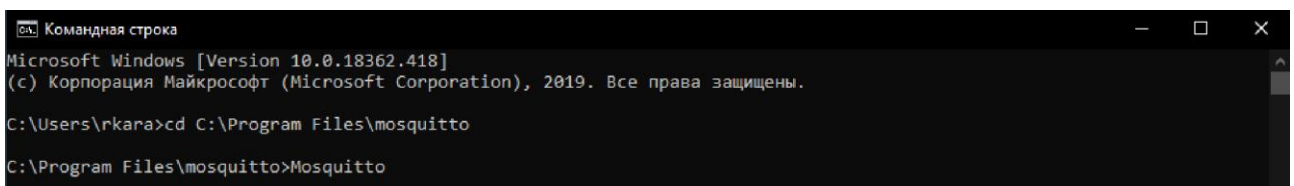
Щоб запустити брокер, переходимо в командний рядок та вводимо `cd` і директорію в якій встановлений `mosquitto`, в нашому випадку це `C:\Program Files\mosquitto` (рис. 4.1).



```
Командная строка
Microsoft Windows [Version 10.0.18362.418]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.
C:\Users\rkara>cd C:\Program Files\mosquitto
```

Рисунок 4.1

Далі пишемо `mosquitto` і нажимаємо `Enter` (рис. 4.2).

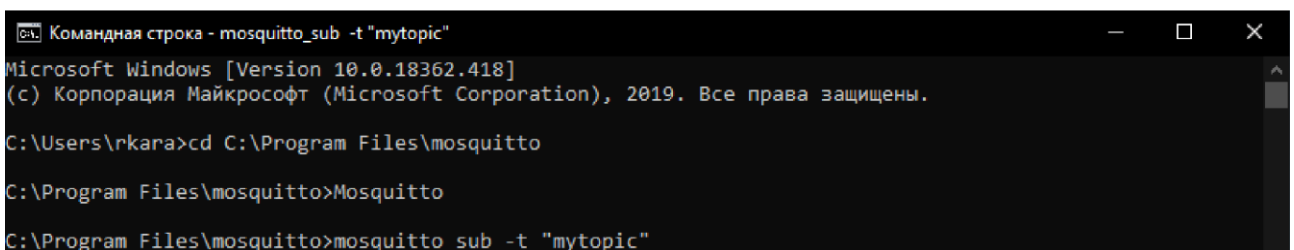


```
Командная строка
Microsoft Windows [Version 10.0.18362.418]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.
C:\Users\rkara>cd C:\Program Files\mosquitto
C:\Program Files\mosquitto>Mosquitto
```

Рисунок 4.2

Брокер Mosquitto MQTT готовий до роботи.

Проведемо тест чи працює передача і отримання повідомлень в `mosquitto`. Для цього відкриваємо два командних рядки. В першому командному рядку підпишемося на тему «`mytopic`» ввівши команду `mosquitto_sub -t mytopic` (рис. 4.3).



```
Командная строка - mosquitto_sub -t "mytopic"
Microsoft Windows [Version 10.0.18362.418]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.
C:\Users\rkara>cd C:\Program Files\mosquitto
C:\Program Files\mosquitto>Mosquitto
C:\Program Files\mosquitto>mosquitto_sub -t "mytopic"
```

Рисунок 4.3

В другому командному рядку опублікуємо тему «mytopic» та передамо значення «1». Для цього вводимо команду `mosquitto_pub -t mytopic -m 1` і нажимаємо Enter (рис. 4.4).

```
C:\Program Files\mosquitto>mosquitto_pub -t mytopic -m 1
```

Рисунок 4.4

Як бачимо значення «1» появилось в першому командному рядку, отже все працює добре (рис. 4.5).

```
C:\Program Files\mosquitto>mosquitto_sub -t "mytopic"
1
```

Рисунок 4.5

4.2. Опис клієнтського додатку на операційній системі Android

Ми використовуємо додаток MQTT dashboard що дозволяє підключатися до MQTT брокера Mosquitto і отримувати/відправляти дані на конкретні топіки. Особливість цього додатку полягає в тому що він легко масштабований і звичайний користувач може опублікувати чи отримати повідомлення не володіючи знаннями з програмування.

Створюємо новий проект. Вводимо назву проекту, адресу mqtt брокера і порт (рис 4.6).

Рисунок 4.6 – Вкладка створення нового проекту

На вкладці subscribe створюємо 5 об'єктів вказавши їм назву топіка (рис. 4.7).

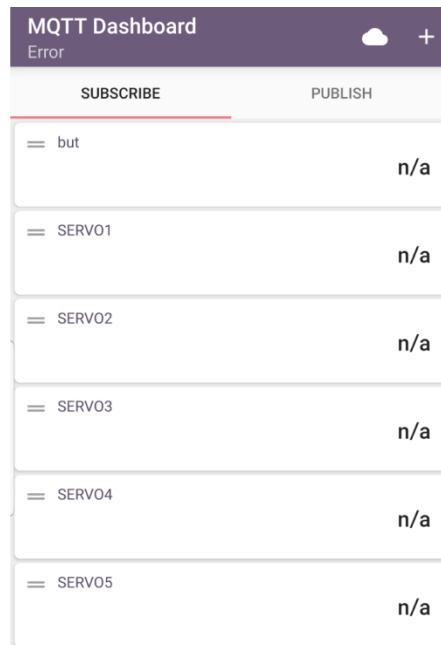


Рисунок 4.7 – Вкладка Subscribe

Таблиця 1 – Назви об'єктів і відповідні їм назви топіків які будуть прийматися

Friendly name (назва об'єкта)	Topic (назва топіка)
Servo1	inTopic1
Servo2	inTopic2
Servo3	inTopic3
Servo4	inTopic4
Servo5	inTopic5
But1	inTopic6

На вкладці Publish створюємо 5 об'єктів типу «повзунок» вказавши їм назву топіка а також мінімальне і максимальне значення яке вони будуть публікувати, а також переключатель з можливими значеннями «1» і «0» (рис. 4.8).

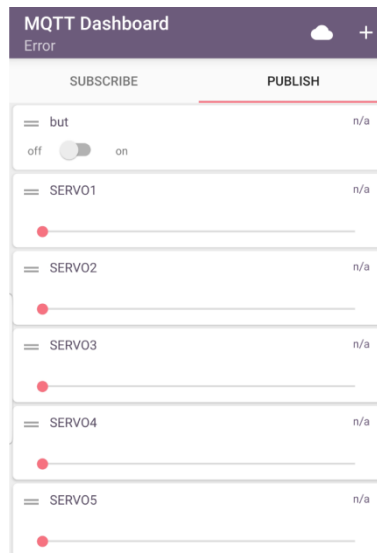


Рисунок 4.8 – Вкладка Publish

Таблиця 4 – Назви об'єктів і відповідні їм назви топіків з максимальними і мінімальними значеннями які будуть видаватися

Friendly name (назва об'єкта)	Topic (назва топіка)	Min value	Max value
Servo1	inTopic1	0	255
Servo2	inTopic2	0	255
Servo3	inTopic3	0	255
Servo4	inTopic4	0	255
Servo5	inTopic5	0	255
But1	inTopic6	0	1

4.3. Мікроконтроллер ESP8266

При проектуванні руки маніпулятора ми вирішили використовувати в якості контролера модуль ESP8266 Node MCU v3 (див. рис. 4.9), оскільки: він містить достатню нам кількість входів і виходів; містить вбудований стек протоколів TCP/IP; містить декілька послідовних портів UART, що будуть необхідні для відлагодження системи; забезпечує задовільною обчислювальною потужністю. Також великою перевагою плати є її мале

енергоспоживання. Часто їх використовують в схемах з автономним живленням.

Платформа має сучасне API для апаратного введення і виведення. Це дозволяє скоротити кількість дій під час роботи з обладнанням і при його налаштуванні. За допомогою прошивки NodeMCU можна задіяти весь робочий потенціал для швидкої розробки пристрою.



Рисунок 4.9 – Зовнішній вигляд контролера ESP8266 NodeMCU v3

4.3.1. Технічні характеристики модуля ESP8266 NodeMCU:

Основні технічні характеристики плати.

- Підтримка Wi-Fi протоколу 802.11 b / g / n;
- Підтримка режиму Wi-Fi – клієнт, точка доступу;
- Вхідна напруга 3,7 В - 20 В;
- Робоча напруга 3В-3,6В;
- Максимальний струм 220мА;
- Вбудований стек TCP / IP;
- Діапазон робочих температур від -40С до 125С;
- 80 МГц, 32-бітний процесор;
- Час пробудження і відправки пакетів 22МС;
- Вбудовані TR перемикач і PLL;
- Наявність підсилювачів потужності, регуляторів, систем керування живленням.

Існує кілька поколінь плат NodeMcu - V1 (версія 0.9), V2 (версія 1.0) і V3 (версія 1.0). Позначення V1, V2, V3 використовуються при продажу в

інтернет-магазинах. Нерідко відбувається плутанина в платах - наприклад, V3 зовні ідентична V2. Також всі плати працюють за принципом open-source, тому їх можуть виробляти будь-які фірми.

Живлення мікроконтролера ESP8266 NodeMcu v3

Подавати живлення на модуль можна декількома способами:

- Подавати 5-18 вольт через пін «Vin»;
- 5В через USB-роз'єм або контакт VUSB;
- 3,3 через вивід 3V.

4.3.2. Переваги і недоліки модуля NodeMcu v3

Переваги модуля:

- Наявність на платі інтерфейсу UART-USB з вбудованим роз'ємом мікро USB дозволяє користувачу без лишніх зусиль підключити плату до персонального комп'ютера.
- Наявність на платі флеш-пам'яті розміром 4 Мбайт.
- Можливість оновлювати прошивку мікроконтролера через USB кабель.
- Є можливість створювати скрипти на мові програмування LUA і зберігати їх в файлової системі.

Недоліки модуля:

Основним недоліком є можливість виконувати тільки LUA скрипти, розташовані в оперативній пам'яті. Цього типу пам'яті мало, обсяг складає всього 20 Кбайт, тому написання великих скриптів викликає ряд труднощів. В першу чергу, весь алгоритм доведеться розділяти на лінійні блоки. Ці блоки необхідно записати в окремі файли системи. Всі ці модулі виконуються за допомогою оператора dofile.

При написанні потрібно дотримуватися правила - при обміні даними між модулями потрібно користуватися глобальними змінними, а при обчисленні всередині модулів - локальними. Також важливо в кінці кожного написаного скрипта викликати функцію `collectgarbage` (збирач сміття).

Терморегулятори NodeMcu v3

Модуль V3 має 11 контактів введення-виведення загального призначення. Крім цього деякі з виходів володіють додатковими функціями: (рис. 4.10)

- D1-D10 - виходи з широтно імпульсною модуляцією;
- D1, D2- виходи для інтерфейсу I²C / TWI;
- D5-D8 - виходи и для інтерфейсу SPI;
- D9, D10 - UART;
- A0 - вхід з АЦП.

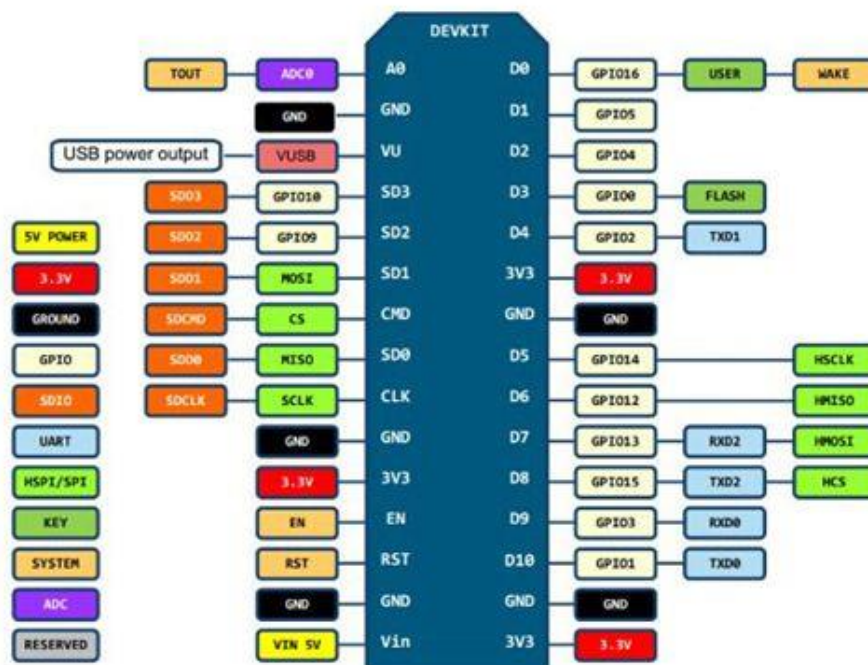


Рисунок 4.10 - Схема розташування виводів ESP8266 NodeMCU v3

«Мікроконтролер не має на кристалі енергонезалежної пам'яті для користувача. Виконання програми ведеться із зовнішнього SPI ПЗП шляхом

динамічного завантаження необхідних проміжків програми в КЕШ інструкцій. Завантаження виконується апаратно, прозоро для програміста. Підтримується до 16 МБ зовнішньої пам'яті програм. Можливий Standard, Dual або Quad SPI інтерфейс. Виробник не надає документації на внутрішню периферію контролеру. Замість цього він надає набір бібліотек, через API яких програміст отримує доступ до периферії.»[6]

Програмування мікроконтролера виконується в середовищі Arduino IDE мовою програмування C. Мікроконтролер підключається до комп'ютера через microUSB, який забезпечує його живленням а також через цей роз'єм здійснюється його програмування.

4.4. Програмування системи керування

4.4.1. Опис середовища програмування Arduino IDE

Для створення розробок на базі Arduino, необхідно програмне забезпечення для написання і завантаження програм в мікроконтролер. Дані функції виконуються за допомогою середовища програмування Arduino IDE. Середовище програмування Arduino складається з текстового редактора для написання програмного коду, панелі інструментів, вікна виведення тексту. Для завантаження програм необхідно підключити середовище розробки до апаратної частини мікроконтролера. Будь-яка програма, написана в Arduino IDE, має назву скетч. Скетч пишуть за допомогою текстового редактора, який містить інструменти, що дозволяють виробляти над ним будь-які дії, такі як вставка / видалення, заміна / пошук тексту. Під час збереження або компіляції скетчу в області повідомлень з'являються повідомлення про помилку, в разі неправильного написання програмного коду. Мова програмування Arduino стандартний C ++, які мають особливості, які полегшують написання програм:

- Файли програм перед компіляцією обробляються препроцесором Arduino;

- Программист зобов'язаний написати дві обов'язкові функції. Це `setup()`, викликається при старті і `loop()` повторюється в нескінченному циклі;

- В текст програми не обов'язково записувати заголовки при використанні стандартних бібліотек;

- Відсутність попередніх налаштувань компілятора.

Завантаження скетчу відбувається через вбудований завантажувач (Bootloader), є програмою, що дозволяє завантажувати код програми без використання будь-яких додаткових апаратних засобів. А також може працювати через інтерфейси RS-232, USB і Ethernet. Завантажувач активний протягом періоду в декілька секунд при завантаженні будь-якого скетчу в мікроконтролер. Про роботу завантажувача сигналізує світлодіод, вбудований в плату мікроконтролера.

Для знаходження рішення задачі управління рукою маніпулятором в Arduino IDE необхідно виконати наступні етапи налаштувань:

- Створення самого проекту;
- Конфігурація обладнання;
- Підключення пристроїв до мережі;
- Програмування мікроконтролера;
- Налаштування візуалізації;
- Завантаження даних конфігурації;
- Діагностика роботи різних функцій обладнання.

Інтерфейс програми забезпечує просту навігацію по задачам і даними проекту

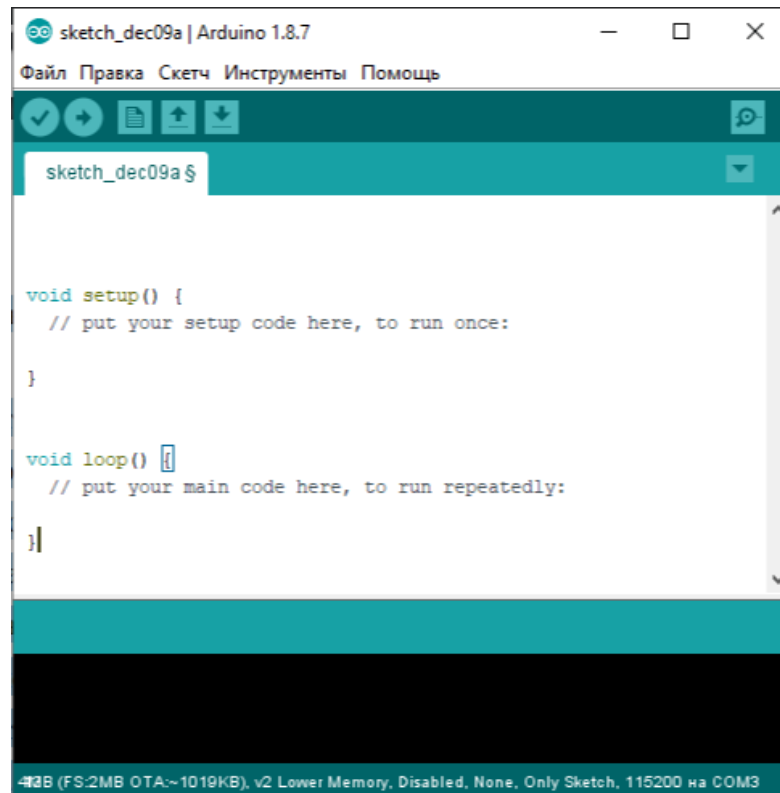


Рисунок 4.11-Зображення структури інтерфейсу створення нового скетчу

- 1) Кнопки панелей інструментів, що дозволяють перевірити, завантажити, створити програму, відкрити, закрити або зберегти скетч, а також відкрити моніторинг порту;
- 2) Текстовий редактор, призначений для написання скетчу;
- 3) Консоль;
- 4) Область повідомлень з інформацією про обраний в поточному мікроконтролері порту виводу.

4.4.2. Розробка програми керування рукою-маніпулятором

Далі будуть детально описані етапи створення проекту з керуванням сервоприводами. Дані та код програми, розроблені в процесі створення проекту, написані по порядку. Номери портів введення / виводу, записані далі, використовувалися при роботі з мікро контролером ESP-8266.

Перший крок полягає в підключенні необхідних бібліотек, ініціалізації Wi-Fi і MQTT клієнта а також вказуємо назву мережі, пароль якщо мережа не відкрита і адресу mqtt клієнта (рис. 4.12).

Директива «#include» пропонує компілятору включити інший вихідний файл, ім'я якого вказується після директиви.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char *ssid = "Android";
const char *password = "";
const char *mqtt_server = "192.168.43.205";

WiFiClient espClient;
PubSubClient client(espClient);
```

Рисунок 4.12 – Фрагмент написаного коду програми підключення бібліотек, ініціалізації Wi-Fi і MQTT клієнта

Наступний крок полягає в ініціалізації портів виведення мікроконтролера Arduino. Для більш зручного орієнтування в кодї програми за допомогою директиви «#define» кожному задіяному виходу будуть присвоєні імена, які більш явно виражають їх виконувані функції.

Таблиця 5 - Таблиця директиви «#define» для програми скетчу керування рукою маніпулятором.

Назва виводу	Номер виводу	Опис
SERVO1	16	ШІМ вивід, що подає сигнал керування з заданою шириною імпульсу
SERVO2	5	ШІМ вивід, що подає сигнал керування з заданою шириною імпульсу
SERVO3	4	ШІМ вивід, що подає сигнал керування з заданою шириною імпульсу
SERVO4	0	ШІМ вивід, що подає сигнал керування з заданою шириною імпульсу
SERVO5	2	ШІМ вивід, що подає сигнал керування з заданою шириною імпульсу
BUTTON1	14	Цифровий вивід, який подає сигнал на включення/виключення сервоприводів

```

#define SERVO1 16
#define SERVO2 5
#define SERVO3 4
#define SERVO4 0
#define SERVO5 2
#define BUTTON1 14

```

Рисунок 4.13 - Запис ініціалізації в програмі входів / виходів мікроконтролера директивою «#define»

Вивід 14 – це цифровий вивід, налаштований за допомогою функцій «pinMode()» і «digitalWrite()», працює як сигнал на включення/виключення сервоприводів руки-маніпулятора.

ШІМ виходи 16, 5, 4, 0, 2 подають на драйвер сигнал керування з відповідною шириною імпульсу для керуванням кута повороту серводвигунів за допомогою функції «analogWrite()».

Далі застосуємо функцію зворотного виклику для отримання топіків від брокера. Функція дуже зручна, оскільки написання загального коду, який не залежить від логіки в викликаній функції і може бути повторно використаний з різними зворотами.

Функцією «Serial.print(topic)» виводимо в серіал прінт назву топіка, виводимо значення отриманих даних, отримуємо назву топіка (рис. 4.14)

```

void callback(char* topic, byte* payload, unsigned int length)
{
  Serial.println();
  Serial.print(topic);
  Serial.print(" => ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }

  String strTopic = String(topic);

```

Рисунок 4.14 – Функція отримання топіків від брокера

Для ініціалізації сервоприводів використовується блок на (рис. 4.15). Даний блок перевіряє чи з потрібного топика прийшли дані. Якщо так, то зчитуються з нього значення які прийшли, конвертуються в ШІМ, конвертуються значення 0-255 в значення 0-1023 і встановлюється рівень ШІМ сигналу. Цей код програми дублюємо 5 разів, оскільки для реалізації руки-маніпулятора використовується 5 сервоприводів. В кожному наступному блоці міняємо назву топика на відповідну йому назву сервопривода

```

if (strTopic == "inTopic1")
{
  payload[length] = 0;
  String strPayload = String((char*)payload);
  int val = strPayload.toInt();
  int stled = map(val, 0, 255, 0, 1023);
  analogWrite(SERV01, stled);
  Serial.print(" => ");
  Serial.print(stled);
}

```

Рисунок 4.15 – Фрагмент коду програми для керуванням сервоприводів

В ході розробки проекту виникла необхідність створення переключателя, який міг би при зміні значень 1 і 0 включити або виключити всі сервоприводи. Реалізація такого коду зображена на (рис. 4.16).

```

if (strcmp(topic, "inTopic6") == 0) {
  if ((char)payload[0] == '1') {
    digitalWrite(BUTTON1, HIGH);
  }else {
    digitalWrite(BUTTON1, LOW);
  }
}
}

```

Рисунок 4.16 – Фрагмент коду програми для реалізації переключателя

Функція «setup ()», що викликається при старті скетчу. У середині фігурних дужок знаходиться код, який ініціалізує змінні, одразу як програма почне роботу, визначає режим роботи виводів контролера, запускає

використовувані бібліотеки. Запускається лише раз після подачі живлення на мікроконтролер.

Завантаживши програму, Arduino дає нашому коду можливість взяти участь в ініціалізації системи. Для цього ми повинні вказати мікроконтроллеру команди, які він виконає в момент завантаження і потім забуде про них. І саме з цією метою в нашій з вами програмі ми повинні виділити блок, в якому будуть зберігатися ці команди. `void setup ()`, а вірніше простір всередині фігурних дужок цієї функції, і є таким місцем всередині Arduino скетчу.

Функція «`pinMode`» використовується для ініціалізації режиму роботи виходів плати мікроконтролера. Запис `pin` ініціалізує вхід мікроконтролера, який буде використовуватися в роботі. `Mode` встановлює режим роботи входи або вихід.

На (рис. 4.17) бачимо ініціалізацію змінних які відповідають за сервоприводи і кнопку-переключатель. Режим роботи ставимо на вихід.

```
void setup() {
  pinMode(BUTTON1, OUTPUT);
  pinMode(SERVO1, OUTPUT);
  pinMode(SERVO2, OUTPUT);
  pinMode(SERVO3, OUTPUT);
  pinMode(SERVO4, OUTPUT);
  pinMode(SERVO5, OUTPUT);
  Serial.begin(115200);
}
```

Рисунок 4.17 – Фрагмент коду програми функції «`setup ()`» для ініціалізації змінних

Функція «`loop ()`» повторюється в нескінченному циклі. Вона ініціалізує початкові значення змінних, виробляє обчислення в програмі і реагує на них. Почавши виконання з першої команди, мікроконтролер дійде до кінця і відразу ж перестрибне в початок, щоб повторити ту ж

послідовність. І так безліч разів (до тих пір, поки на плату буде подаватись струм).

В середині блока функції «loop ()» пишемо код програми який буде відповідати за підключення контроллера до Wi-Fi мережі і MQTT брокера. Підписуємось на топіки за допомогою функції «client.subscribe ("");» і вказуємо id клієнта (повинен бути унікальний в мережі) за допомогою функції «client.connect ("");» Останній фрагмент програми відповідає за перевірку підключення і запуск циклу отримання топіків (рис. 4.18).

```
void loop() {  
  if (WiFi.status() != WL_CONNECTED)  
  {  
    WiFi.begin(ssid, password);  
    if (WiFi.waitForConnectResult() != WL_CONNECTED)  
      return;  
  }  
  if (WiFi.status() == WL_CONNECTED)  
  {  
    if (!client.connected())  
    {  
      client.setServer(mqtt_server, 1883);  
      client.setCallback(callback);  
      client.connect("ESP03led");  
      client.subscribe("inTopic1");  
      client.subscribe("inTopic2");  
      client.subscribe("inTopic3");  
      client.subscribe("inTopic4");  
      client.subscribe("inTopic5");  
      client.subscribe("inTopic6");  
    }  
    if (client.connected()) {  
      client.loop();  
    }  
  }  
}
```

Рисунок 4.18 – Фрагмент коду програми функції «loop ()»

РОЗДІЛ 5. ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА

5.1. Вдосконалення організації наукових досліджень

Основна мета пошукових прикладних науково-дослідних робіт – знаходження нових шляхів дослідження та створення нової техніки та технології виробництва, практичне застосування досліджуваних явищ та фактів, вони направлені на створення нових технологічних процесів, механізмів, машин, виробів, організаційних та економічних структур, методичних рекомендацій. На стадії виконання пошукових робіт визначається науково – технічний ефект. Вияснюються можливі області застосування результатів в народному господарстві і на основі комплексного якісного аналізу дається характеристика очікуваної технічної та економічної ваги результатів пошукових робіт. При цьому визначається сукупність показників, які змінюються під впливом впровадження результатів робіт, а також можливий діапазон цих змін.

Основні етапи виконання проекту управління роботом маніпулятором:

- постановка задачі;
- аналіз досліджень управління маніпулятором;
- розробка методики та проведення експериментальних досліджень;
- уточнення моделей та оптимізація параметрів та режимів роботи маніпулятора;
- розробка систем керування за допомогою смартфона по протоколу передачі повідомлень mqtt.

Основними напрямками вдосконалення організації наукових досліджень є:

- математичне планування експерименту та скорочення обсягів дослідних робіт;

- розширення області пошуку за рахунок використання нових технічних засобів;
- використання комп'ютерно-інтегрованих технологій, автоматизація процесу вимірювань та скорочення трудоємності і підвищення точності вимірювань;
- розробка подібно-функціональних математичних моделей, реалізація на ПК обчислювального експерименту, що дозволяє уникнути проведення попередніх експериментальних досліджень.

З цією метою для реалізації поставлених задач розроблена програма для реалізації експерименту за ортогональним планом другого порядку, ядром якого є план повно факторного експерименту типу ПФЕ 2^k , де $k = 2$ - кількість факторів. Функція відгуку при цьому знаходиться у вигляді полінома другого порядку.

Крім цього методика проведення експериментальних досліджень процесу керування роботом маніпулятором передбачала проведення експериментальних досліджень в навчальній аудиторії, використанням спеціального керуючого та вимірною обладнання аудиторії 503 ТНТУ.

Для реалізації руки маніпулятора була використана модель 6DOF маніпулятора і сервоприводи MG996R. При цьому для живлення сервоприводів використовували блок живлення Gametax GM-400, який забезпечує стабільну роботу двигуна у будь-якому режимі.

5.2. Планування та розрахунок перед виробничих затрат та капіталовкладень на проведення дипломного проекту

При планування затрат на виконання проекту розрізняють перед виробничі затрати $z_{НДР}$ та капіталовкладення $K_{НДР}$.

Перед виробничі затрати складаються із затрат на виконання таких робіт: постановка задач НДР та розробка технічного завдання; теоретичні дослідження та огляд літератури; лабораторні та заводські дослідження; проектування та конструювання виробів, обладнання, оснастки техпроцесів, цехів і т.д., що є об'єктами НДР; виготовлення, випробування та підналадка зразків. Всі розглянуті затрати є поточними затратами для виконання НДР. Проте при визначені капіталовкладень та госпрозрахункового економічного ефекту від впровадження результатів НДР, виробничі затрати повинні впроваджуватись разом з поточними затратами виробництва нових видів продукції, обладнання і т.д.

Капіталовкладення, які необхідні для виконання НДР, складаються із вкладів в лабораторне обладнання, апарати, прилади з врахуванням затрат на їх проектування та монтаж; в будови та споруди лабораторій, необхідність в яких обумовлена виконанням даної НДР.

Капітальні вклади в НДР складають окремими складовими (додатками) в загальну суму, разом з прямими вкладками в підприємство, що виготовляє продукцію, а також (спряженими і супутніми) вкладками і інші галузі. Їх величина приймається в частині, що відповідає зайнятості обладнання лабораторії, будов та інших засобів на протязі року виконання розглядуваної НДР.

Для визначення перед виробничих (поточних) затрат на виконання НДР складається кошторис затрат, вихідними даними для якого є:

- план проведення НДР;
- розрахунок вартості обладнання для проведення НДР;

– план потреби в основних та допоміжних матеріалах та готових покупних виробках;

– план по праці та заробітній праці.

Для планування праці та заробітної плати на виконання НДР необхідно визначити:

1. Етапи впровадження НДР.
2. Трудомісткість етапів в людино–годинах, людино–днях.
3. Кількість учасників, що виконують роботи по окремих етапах.
4. Тривалість окремих етапів НДР в днях.

Трудомісткість НДР та її окремих етапів визначається за даними НДЧ ТДТУ. На основі трудомісткості встановлюється чисельність робочих, фонд зарплати. Оплата праці науково – технічного персоналу основних лабораторій, проводиться у відповідності із схемою посадових окладів, затверджених вище згаданою організацією для даного вузу чи НДІ. Посадовий оклад повинен мати вилку.

Після розрахунку кількості робочих та фонду зарплати слід визначити продуктивність праці (відношення кошторисної вартості робочих до кількості працюючих) та середню зарплату.

Розрахунок вартості обладнання для проведення дипломної роботи

Вартість обладнання:

$$K_{\text{НДР}} = \sum_{i=1}^n K_{\text{НДР}}^i \cdot N_i \cdot \eta_i, \text{ де}$$

n – кількість найменувань обладнання та інших засобів, які застосовуються для виконання НДР;

$K_{\text{НДР}}^i$ – вартість одиниці засобів i -того виду, $\frac{\text{грн}}{\text{од}}$;

N_i – кількість екземплярів i -того засобу, необхідних для

виконання дослідження, *од*;

η_i – коефіцієнт зайнятості засобу i -того виду на протязі року для виконання даного i -того дослідження.

Якщо обладнання та прилади будуть використовуватись і після завершення НДР, то необхідно визначити величину амортизаційних відрахувань, що припадають на дану НДР. Амортизація за час (період) використання обладнання складає долю затрат, які припадають на дослідницьку роботу, і визначаються так:

$$A = \frac{C_B \cdot N_A \cdot T_{\text{ФАК}}}{T_{\text{ГОД}}}, \text{ де}$$

C_B – балансова вартість обладнання, *грн*;

N_A – норма амортизаційних відрахувань в рік, %;

$T_{\text{ГОД}}$ – річний робочий фонд часу, *год*;

$T_{\text{ФАК}}$ – фактичний час роботи обладнання по дослідній темі, *год*.

Оскільки для проведення досліджень використовувалось установа, що виготовлялась для вказаних досліджень то її вартість визначаємо за витратами на виготовлення.

Оцінка вартості витрат установки робота маніпулятора.

Характеристика установки:

Таблиця 6 – Параметри установки

№	Назва	Параметри
1	Висота, мм	480
2	Довжина, мм	420
3	Ширина, мм	100
4	Потужність приводу, квт	0.1
5	Маса, кг	10

Оцінка вартості:

Таблиця 7 – Вартість комплектуючих

Вартість комплектуючих (за фактичними витратами)				Вартість, грн
№	Назва	Розміри, мм	Кількість	
1	Рука маніпулятора		1 шт	730
2	Сервоприводи		5 шт	130
3	Arduino UNO		1 шт	95
4	ESP 8266		1 шт	120
5	Набір проводів	100	10 шт	1
6	Вилка електрична		1 шт	15
7	Виключачель (тумблер)		1 шт	14
8	Блок живлення		1 шт	450
9	Кріпильні деталі		-	12
10	Допоміжні матеріали		-	50
Сумарна вартість комплектуючих				2146
Вартість складання та наладки*				134
Вартість установки				2280

Приймаємо, що установка використовувалась тільки для наших досліджень

Розрахунок амортизації дослідного обладнання

Для обладнання аудиторії, що використовувалось в експериментах
 $C_B = 2250$ грн; $N_A = 20\%$; $T_{ГОД} = 2016$ год.

Фактичний час роботи обладнання під час проведення експериментів -
 $T_{ФАК} = 90$ год.

Відповідно амортизація за використання обладнання аудиторії складає

$$A = \frac{C_B \cdot N_A \cdot T_{\text{ФАК}}}{T_{\text{ГОД}}} = \frac{2250 \cdot 0,2 \cdot 52}{2016} = 116,07 \text{ грн}$$

Експериментально-виробничі витрати визначаються як витрати на машинний час, який є потрібним для виконання необхідного об'єму робіт виходячи з його вартості за одиницю часу. Вартість роботи на ПЕОМ і користування мережею Інтернет (A_{II}) встановлюємо виходячи з реальних даних. Приймаємо середній тариф роботи на ПЕОМ 2 грн. / год.

$$A_{II} = 360 \text{ грн}$$

Експериментально- виробничі витрати

$$C = 965 + 116,07 + 360 = 1441,07 \text{ грн}$$

Розрахунок витрат на електроенергію

Час роботи установки складав безпосередньо 40% ($\psi = 0,6$) від часу проведення експеримент. $T_{\text{ФАК}}$. Середній коефіцієнт навантаження становив $\eta = 0,7$.

Відповідно витрати на електроенергію становлять $E_N = \psi \eta p_E N T_{\text{ФАК}}$,

де p_E - ціна електроенергії (за квт год) $p_E = 0,9$ грн.

$$E_N = \psi \eta p_E N T_{\text{ФАК}} = 0,6 \cdot 0,7 \cdot 0,9 \cdot 1,5 \cdot 90 = 51,03 \text{ грн}$$

Витрати електроенергії на освітлення та комп'ютер $E_0 = 0,9 \cdot 75 = 67,5$ грн.

Сумарні витрати на електроенергію

$$E_{\Sigma} = E_N + E_0 = 5,11 + 65,5 = 70,61 \text{ грн.}$$

Розрахунок затрат на заробітну плату

Для визначення загальної тривалості проведення наукових досліджень доцільно дані витрат часу на виконання окремих стадій (етапів) звести у таблицю 3.1

Таблиця 8 – Середній час виконання розробки

Номер і назва етапу	Середній час виконання етапу, год.	
	інженер	керівник
Постановка задачі	25	24
Розробка теоретичних моделей	120	12
Розробка методики та проведення експериментальних досліджень	80	10
Уточнення моделей та оптимізація параметрів та режимів роботи	40	12
Розробка систем керування та інших технічних рішень	65	12
Разом	~330	~70

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25 %-35 % від прямої з/п. При розрахунку з/п кількість робочих днів в місяці приймаємо рівною 25,4 дні / міс, що відповідає 203,2 год. / міс. Розмір місячних окладів керівника приймаємо 8000 грн. та інженерів — 5000 грн.

Пряма з/п визначається наступним чином:

$$ЗП = (O_i - T_i) / 203,2$$

де O_i — розмір місячних окладів 1-х категорій працівників;

T_i — трудомісткість робіт виконаних працівниками /*-х категорій.

Для інженера:

$$ЗП = (5000 \cdot 330) / 203,2 = 8120,08 \text{ (грн.)};$$

Для керівника:

$$ЗП = (8000 \cdot 70) / 203,2 = 2755,90 \text{ (грн.)}.$$

Величина доплат визначається наступним чином:

$$ЗП_1 = ЗП \cdot K_1$$

де K_1 — коефіцієнт доплат (0,25-0,35).

Приймаємо коефіцієнт доплат рівним 0,3:

Для інженера:

$$ЗП_1 = 8120,08 \cdot 0,3 = 2436 \text{ (грн.)};$$

для керівника:

$$ЗП_1 = 2755,90 \cdot 0,3 = 826,77 \text{ (грн.)};$$

Основна з/п визначається наступним чином:

$$ЗП_0 = ЗП + ЗП_1$$

Для інженера:

$$ЗП_0 = 8120,08 + 2436 = 10556,08 \text{ (грн.)};$$

для керівника:

$$ЗП_0 = 2755,90 + 826,77 = 3582,67 \text{ (грн.)}.$$

Величина додаткової з/п визначається наступним чином:

$$ЗП_0 = ЗП_0 \cdot K_0$$

Де K_{δ} — коефіцієнт додаткової з/п (0,05-0,1).

Приймаємо коефіцієнт додаткової з/п рівним 0,1, тоді:

для інженера:

$$ЗП_{\delta} = 10556,08 \cdot 0,1 = 1055,61 \text{ (грн.)};$$

для керівника:

$$ЗП_{\delta} = 3582,67 \cdot 0,1 = 358,27 \text{ (грн.)}.$$

Витрати, на проведення розробки програмного продукту крім річного фонду заробітної плати включають ще й соціальні нарахування. Нормативи нарахувань на заробітну плату наступні:

- фонд страхування від безробіття – 1,3%;
- пенсійний фонд – 31,8%;
- фонд соціального страхування – 2,9%;
- фонд соціального страхування від нещасних випадків і професійних захворювань 1%.

Всього норматив нарахувань на заробітну плату інженера становить 37%:

$$11611,69 \cdot 0,37 = 4296,32 \text{ (грн.)},$$

а для керівника 37 % :

$$3914,97 \cdot 0,37 = 1458,16 \text{ (грн.)}.$$

Таким чином, результати розрахунку заробітної плати та нарахувань на неї зведемо в таблицю 9.

Таблиця 9 – Зведена відомість витрат на заробітну плату, грн.

	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата	Нарахування на заробітну плату	Всього витрат на заробітну
		Пряма заробітна плата	Доплати	Всього			
1	Інженер	8120,08	2436	10556,08	1055,61	11611,69	23223,38
2	Керівник	2755,90	826,77	3582,67	358,27	3914,97	7855,91
Всього		10875,98	3262,77	14138,75	1413,88	15526,66	31079,29

Загальновиробничі витрати при укрупнених розрахунках приймаємо на рівні 80 % від суми основної і додаткової з/п інженера, яка була нарахована за роботу по проведенні досліджень. Аналогічно визначаються адміністративні витрати, які доцільно прийняти на рівні 50 % від суми основної і додаткової з/п інженера. Поза виробничі витрати приймаємо на рівні 5 % від виробничої собівартості.

Розрахунок поточних витрат на проведення розробки програмного продукту зведемо в таблицю 10.

Таблиця 10 – Калькуляція собівартості проведення дипломного проекту (розробки програмного продукту)

Статті витрат	Витрати, грн.	В % до загальної суми, %
1. Основна заробітна плата	14138,75	37,97
2. Додаткова заробітна плата	1413,88	3,7

Продовження таблиці 10

3. Нарахування на заробітну плату	15526,66	41,7
4. Консультації	50,00	0,13
5. Матеріали	2280	6,12
6. Електроенергія	70,61	0,19
7. Експериментально-виробничі витрати	1441,07	3,87
Загальновиробничі витрати	1242,92	3,34
Разом виробнича собівартість	36163,9	97,12
7. Адміністративні витрати	776	2,08
8. Поза виробничі витрати	296,1	0,8
Повна собівартість	37236	100%

Розрахунок ціни розробки і економічна ефективність від використання програмного продукту

Ціну розробки науково-дослідного продукту можна визначити

$$C = (C_{np} / N_z + C_{кон}) + П$$

де C_{np} — собівартість розробки програмного продукту), грн.;

N_z — кількість замовлень, од.;

$C_{кон}$ — собівартість копіювання (ксерокопії, дискети, компакт-диски, поштові витрати, відрядження спеціалістів для запуску та наладки програмного забезпечення тощо), грн.;

$П$ — нормативна величина прибутку (15 % - 30 % від собівартості $C_{пр}$).

Нехай собівартість копіювання програмного продукту становитиме 100 грн.

Прийmemo нормативну величину прибутку рівною 20 % від собівартості. Таким чином ціна розробки програмного продукту складатиме;

$$Ц = (37236 / 1 + 100) + 1402 = 38738$$

Отже ціна розробки програмного продукту дорівнює 46485,6 грн.

Оцінка економічної ефективності розробки

При визначенні економічної ефективності необхідно порівняти використовуваний (базовий) програмний продукт і пропонуванний. З допомогою відповідних розрахунків (в разі значної складності використання експертних оцінок) визначається скорочення трудовитрат і (або) машинного часу, і як наслідок — економія коштів при використанні нового програмного продукту.

Критерії оцінки економічної ефективності дипломного проекту

Річну економію, досягнуту при реалізації НДР у виробництві визначають так:

$$E_{УМОВ.РІЧ} = (S_1 - S_2) \cdot Q_2,$$

де S_1 та S_2 - собівартість транспортування одиниці маси вантажу на 1 метр до і після впровадження НДР;

Q_2 - обсяги транспортованого вантажу за рік.

При зменшенні енергоємності транспортування в середньому на 50% економія складе 0,5 витрат на електроенергію для транспортування вантажу

$$E_{УМОВ.РІЧ} = 0,5C_{ТРАНС} = 0,5p \cdot K \cdot N_{\Sigma} \cdot T,$$

де K - коефіцієнт, що враховує режим використання потужностей N_{Σ} ;

T - час роботи конвеєрів.

Впровадження НДР забезпечує одержання річного економічного ефекту, величина якого визначається по формулі:

$$E_{РІЧ} = [(S_1 + E_H \cdot K_1) - (S_2 + E_H \cdot K_2)] \cdot Q_2, \text{ де}$$

$S_1 + E_H \cdot K_1$ – приведені затрати до впровадження НДР, грн;

$S_2 + E_H \cdot K_2$ – приведені затрати після впровадження НДР, грн;

Q_2 – об'єм транспортних операцій після впровадження НДР.

Враховуючи дані по економії

$$E_{РІЧ} = E_{УМОВ.РІЧ} - E_H \Delta K,$$

де $\Delta K = C + K_{\Delta} = 8513,56 + 1000 = 9513,56$ грн.

Відповідно для того, щоб окупити затрати на дипломний проект за 1 рік необхідно впровадити нововведення на транспортних операціях, де використовуються ГК, де витрати на електроенергію складають $C_{ТРАНС} = 2\Delta K = 19027$ грн.

Для визначення ефективності розробки при менших обсягах транспортування потрібно розрахувати чисту приведену цінність NPV і термін окупності $T_{ок}$:

$$NPV = \sum \left((D_t - B_t) / (1+i)^t \right),$$

де D_t — повний дохід за рік (при використанні нововведень

B_t — повні витрати за рік при використанні нововведень;

t — відповідний рік проекту;

i — дисконтна ставка (0,2).

РОЗДІЛ 6. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1. Правила безпеки при роботі з маніпулятором

При виконанні дипломного проекту дуже важливо дотримуватись правил безпеки при роботі з комп'ютером та маніпулятором.

Персональний комп'ютер від інших електроприладів відрізняється тим, що для нього передбачена можливість постійної експлуатації без відключення від електричної мережі. Окрім звичайного режиму роботи комп'ютер може знаходитися в режимі роботи зі зниженим електроспоживанням або в режимі сну. Блок живлення маніпулятора працює від стандартної електромережі 220В. У зв'язку з можливістю тривалої роботи комп'ютера і маніпулятора без відключення від електромережі слід приділити особливу увагу якості організації електроживлення.

Неприпустимо використання неякісних і зношених компонентів в системі електропостачання, а також їх сурогатних замінників: розеток, подовжувачів, перехідників, трійників. Неприпустимо самостійно модифікувати розетки для підключення вилок, що відповідають іншим стандартам. Електричні контакти розеток не повинні випробовувати механічних навантажень, пов'язаних з підключенням масивних компонентів (адаптерів, трійників і т. п.). Усі живлячі кабелі і дроти повинні розташовуватися із заднього боку комп'ютера і периферійних пристроїв. Їх розміщення в робочій зоні користувача неприпустимо.

Неприпустимою є робота з якими-небудь операціями, пов'язаною з підключенням, відключенням або переміщенням компонентів комп'ютерної системи без попереднього відключення живлення.

Також комп'ютер і маніпулятор не слід встановлювати поблизу електронагрівних приладів і систем опалювання.

Неприпустимо розміщувати на системному блоці, блоці живлення маніпулятора, моніторі і периферійних пристроях сторонні предмети: книги, листи паперу, серветки, чохли для захисту від пилу. Це призводить до постійного або тимчасового перекриття вентиляційних отворів.

Особливості електроживлення монітора. Монітор має елементи, здатні зберігати високу напругу впродовж тривалого часу після відключення від мережі. Розкриття монітора користувачем неприпустимо ні за яких умов. Це не лише небезпечно для життя, але і технічно марно, оскільки усередині монітора немає ніяких органів, регулюванням або налаштуванням яких користувач міг би поліпшити його роботу. Розтин і обслуговування моніторів може робитися тільки в спеціальних майстернях.

Усі компоненти системного блоку отримують електроенергію від блоку живлення. Блок живлення ПК – це автономний вузол, що знаходиться у верхній частині системного блоку. Правила техніки безпеки не забороняють розкривати системний блок, наприклад при установці додаткових внутрішніх пристроїв або їх модернізації, але це не відноситься до блоку живлення. Блок живлення комп'ютера – джерело підвищеної пожежної небезпеки, тому розкриттю і ремонту він підлягає тільки в спеціалізованих майстернях. Блок живлення має вбудований вентилятор і вентиляційні отвори.

У зв'язку з цим в ньому неминуче накопичується пил, який може викликати коротке замикання. Рекомендується періодично (один – двічі в рік) за допомогою пилососа видаляти пил з блоку живлення через вентиляційні отвори без розкриття системного блоку. Особливо важливо

робити цю операцію перед кожним транспортуванням або нахилом системного блоку.

6.2. Вимоги до робочого місця

У вимоги до робочого місця входять вимоги до робочого столу, посадочного місця (стілця, крісла), підставок для рук і ніг. Незважаючи на уявну простоту, забезпечити правильне розміщення елементів комп'ютерної системи і правильну посадку користувача надзвичайно важко. Повне рішення проблеми вимагає додаткових витрат, порівнянних за величиною з вартістю окремих вузлів комп'ютерної системи, тому і в побуті і на виробництві цими вимогами часто нехтують.

Монітор має бути встановлений прямо перед користувачем і не вимагати повороту голови або корпусу тіла. Маніпулятор має бути встановлений збоку від користувача.

Робочий стіл і посадочне місце повинні мати таку висоту, щоб рівень очей користувача знаходився трохи вище за центр монітора. На екран монітора слід дивитися зверху вниз, а не навпаки. Навіть короткочасна робота з монітором, встановленим занадто високо, призводить до стомлення шийних відділів хребта.

Клавіатура має бути розташована на такій висоті, щоб пальці рук розташовувалися на ній вільно, без напруги, а кут між плечем і передпліччям складав 100° – 110° . При тривалій роботі з клавіатурою можливе стомлення сухожилів кистьового суглоба. Відоме важке професійне захворювання – кистьовий тунельний синдром, пов'язане з неправильним положенням рук на клавіатурі. Щоб уникнути надмірних навантажень на кисть бажано надати робоче крісло з підлокітниками, рівень висоти яких, заміряний від підлоги, співпадає з рівнем висоти розташування клавіатури.

При роботі з мишею рука не повинна знаходитися на вазі. Лікоть руки або хоч би зап'ясток повинні мати тверду опору. Якщо передбачити необхідне розташування робочого столу і крісла скрутно, рекомендується застосувати килимок для миші, що має спеціальний опорний валик. Тривалість безперервної роботи з персональним комп'ютером без регламентованих перерв не повинна перевищувати 2 години.

Перед початком роботи потрібно переконатися в наявності захисного заземлення й підключення екранного провідника до корпусу процесора; перевірити правильність установки столу, стільця, кута нахилу екрана, положення клавіатури, відрегулювати частоту, яскравість, контрастність, рівень звуку.

Забороняється приступати до роботи при виявленні несправності устаткування. Для зменшення впливу шкідливих чинників рекомендується підготувати робоче місце так, щоб виключити незручні пози й тривалі напруги, дисплей установити так, щоб низ екрану був на 20 см нижче рівня очей а маніпулятор щоб при виконанні рухів не зачепив яких небуть предметів.

Під час роботи потрібно дотримуватись такої послідовності вимикання обчислювальної техніки: закрити всі активні задачі, програми керування маніпулятором; виключити живлення всіх периферійних пристроїв; виключити живлення системного блоку (процесора); відключити блок живлення маніпулятора. Після роботи на персональному комп'ютері потрібно оглянути і упорядкувати робоче місце. Поставити руку маніпулятора в початкове робоче положення.[8]

Перед включенням маніпулятора необхідно візуально перевірити електрошнур на наявність механічних порушень. Всі електроприлади повинні бути надійно заземлені згідно з правилами улаштування приладу.

Забороняється працювати з електроприладами вологими руками. Не залишати маніпулятора і персональний комп'ютер (який виконує функції сервера) без нагляду на довгий час, після закінчення роботи перевірити, чи всі прилади вимкнені. Категорично заборонено виконувати будь-які ремонтні роботи самостійно.

Рятування життя людини, ураженої струмом, у багатьох випадках залежить від швидкості і правильності дій осіб, що здійснюють допомогу. Передусім потрібно якнайшвидше звільнити потерпілого від дії електричного струму. Якщо неможливо відключити електричне обладнання від мережі, потрібно відразу приступити до звільнення потерпілого від струмопровідних частин, не доторкаючись при цьому до потерпілого.[9]

6.3. Загальні вимоги безпеки при роботі з маніпулятором

Джерелом небезпеки для життя при роботі з комп'ютерами є напруга живлення 220 — 380 В. До роботи з комп'ютером і маніпулятором допускаються особи, які:

- а) ознайомилися з інструкцією щодо роботи з ЕОМ і отримали інструктаж на робочому місці;
- б) засвоїли відповідний практичний курс, необхідний для роботи за комп'ютерами.

Вимоги безпеки при роботі за пультами ЕОМ:

- не вмикати і не вимикати роз'єми кабелів при поданій напрузі живлення;
 - не залишати комп'ютери під живленням без нагляду;
- Перед вмиканням ЕОМ у мережу необхідно переконатись:
- у наявності заземлення приладів;

— у справності шнура живлення, шнура зв'язку клавіатури з блоком живлення;

— увімкнути живлення.

При ураженні струмом треба діяти негайно (див.: Техніка безпеки при користуванні електроприладами).

У разі виникнення пожежі необхідно:

а) використати всі наявні засоби пожежогасіння, крім води і вогнегасника ОХП-10;

б) при необхідності викликати пожежну команду телефоном 01.

Категорично забороняється:

а) включати ЕОМ у розетку при несправному шнурі живлення;

б) під'єднувати і роз'єднувати роз'єми кабелів;

в) проводити будь-який ремонт під час включення ЕОМ.

РОЗДІЛ 7. ЕКОЛОГІЯ

7.1. Вимоги до приміщень для експлуатації моніторів і ПЕОМ. Шляхи дотримання цих вимог.

Широке поширення мікроелектроніки, комп'ютерів індивідуального користування, потужних засобів автоматизованої обробки тексту і графічної інформації, високо ефективних пристроїв її зберігання і пошуку, сучасних засобів зв'язку та мереж електронно-обчислювальних машин дозволяють деяким фахівцям ставити питання про перспективи створення електронних офісів майбутнього. Робота операторів, програмістів, інженерів і просто користувачів безпосередньо пов'язана з комп'ютерами, а відповідно з додатковими шкідливими впливами цілої групи факторів, що істотно знижує продуктивність їх праці.

Конструкція монітора повинна забезпечувати можливість фронтального спостереження екрана шляхом повороту корпусу в горизонтальній площині навколо вертикальної осі в межах 30° і у вертикальній площині навколо горизонтальної осі в межах 30° з фіксацією в заданому положенні. Дизайн повинен передбачати фарбування в спокійні м'які тони з дифузним розсіюванням світла. Корпус монітора і ПЕОМ, клавіатура повинні мати матову поверхню одного кольору з коефіцієнтом відображення 0,4 - 0,6 і не мати блискучих деталей, здатних створювати відблиски. Конструкція ВДТ повинна передбачати наявність ручок регулювання яскравості і контрасту, що забезпечують можливість регулювання яскравості і контрасту, що забезпечують можливість регулювання цих параметрів від мінімальних до максимальних значень. ВДТ і ПЕОМ повинні забезпечувати потужність експозиційної дози рентгеновського випромінювання в будь-якій точці на відстані 0,05 м. від екрану і корпусу монітора при будь-яких положеннях регулювальних

пристроїв не повинна перевищувати $7,74 \times 10$ А / кг, що відповідає еквівалентній дозі, рівної 0,1 мбер / год (100 мкР / год).

При виконанні основної роботи на моніторах і ПЕОМ (диспетчерські, операторські, розрахункові, кабіни і посади управління, зали обчислювальної техніки та ін.) Де працюють інженерно - технічні працівники, що здійснюють лабораторний, аналітичний чи вимірювальний контроль, рівень шуму не повинен перевищувати 60 дБА. У приміщеннях операторів ЕОМ (без дисплеїв) рівень шуму не повинен перевищувати 65 дБА. На робочих місцях у приміщеннях для розміщення гучних агрегатів обчислювальних машин рівень шуму не повинен перевищувати 75 дБА. Шумляче обладнання, рівні шуму якого перевищують нормовані, повинне знаходитися поза приміщенням з монітором і ПЕОМ. Знизити рівень шуму в приміщеннях з моніторами і ПЕОМ можна використанням звукопоглинальних матеріалів з максимальними коефіцієнтами звукопоглинання в області частот 63 - 8000 Гц для обробки приміщень, підтверджених спеціальними акустичними розрахунками. Додатковим звукопоглинанням служать однотонні фіранки з щільною тканини, гармонійні з забарвленням муру і підвішені в складку на відстані 15 - 20 см від огорожі. Ширина фіранки повинна бути в 2 рази більше ширини вікна.[10]

7.2. Добування електроенергії за рахунок спалювання мінерального палива. Забруднення довкілля при цьому та шляхи його зменшення.

Розвиток людської цивілізації базується на енергетиці. Від стану паливно-енергетичного комплексу залежать темпи науково-технічного прогресу й виробництва, а отже, життєвий рівень людей. Як уже зазначалося, темпи зростання виробництва енергії у світі сьогодні є вищими за темпи приросту населення, що зумовлюється індустріалізацією, збільшенням енергозатрат на одиницю продукції в сільському господарстві, в гірничорудній промисловості й т. д.

Джерела енергії, які використовує людство, поділяються на відновлювані — енергія сонця, вітру, морських припливів, гідроенергія річок, внутрішнього тепла землі — й невідновлювані — викопне мінеральне паливо та ядерна енергія. Перші не порушують теплового балансу Землі, оскільки під час їх використання відбувається лише перетворення одних видів енергії на інші (скажімо, енергія Сонця перетворюється спочатку на електроенергію й тільки потім переходить у тепло). Зате використання других спричинює додаткове нагрівання атмосфери й гідросфери. Це небезпечно, бо може призвести до зміни рівня води у Світовому океані, що, своєю чергою, змінить співвідношення площі суші й водного дзеркала, вплине на клімат Землі, на тваринний і рослинний світ.

Отже, є теплова межа, яку людство не повинне переступати, інакше це матиме для нього катастрофічні наслідки. За розрахунками вчених, небезпечної межі буде досягнуто в разі використання невідновлюваних джерел енергії в кількості, яка перевищить 0,1 % потужності потоку сонячної енергії, що надходить на Землю, тобто більш як 100 млрд. кВт. Сьогодні на базі невідновлюваних джерел виробляється енергії в 10 разів менше за гранично допустиму кількість. Якщо темпи збільшення виробництва енергії запишаються такими самими, то теплової межі буде досягнуто приблизно в середині XXI ст. А людство ще й нарощує темпи, і нині 70 % усієї енергії воно отримує за рахунок спалювання вугілля, нафти й газу плюс 7 % — за рахунок роботи атомних електростанцій.

Якщо мінеральне паливо й далі спалюватиметься сьогоднішніми темпами, то, за розрахунками, всі його запаси будуть вичерпані через 130 років. Необхідно наголосити, що спалювання мінеральної сировини — вкрай нерациональний спосіб використання природних ресурсів. Нафта, наприклад, — дуже цінна сировина для хімічного синтезу (сьогодні з неї отримують безліч потрібних матеріалів — синтетичні тканини й каучук, пластмаси, добрива, фарби й тисячі інших). Ще видатний російський хімік Д. І.

Менделєєв з обуренням говорив: «Нафта — не паливо, топити можна й асигнаціями!» Крім вуглеводневого палива й урану, в природі є ще одне невідновлюване джерело енергії. Це дейтерій, або важкий водень, — потенційне паливо для термоядерних електростанцій майбутнього. Запаси його у Світовому океані оцінюються в 19000.

Виробництво електроенергії на ТЕС супроводжується виділенням великої кількості теплоти, тому такі станції, як правило, будуються поблизу міст і промислових центрів для використання (утилізації) цієї теплоти. Зважаючи на обмеженість світових запасів мінерального палива, вчені й технологи продовжують працювати над поліпшенням параметрів енергоблоків, підвищенням їхніх коефіцієнтів корисної дії (ККД), що забезпечує ощадливіше витрачання палива. Так, істотну економію палива дає збільшення одиничної потужності енергоблоків. Сьогодні на ТЕС установлюються енергоблоки потужністю 1000—1200 МВт. Сучасна технологія дає змогу підвищити цю потужність до 3000 МВт, що заощадить кілька процентів палива. Подальше зростання потужності блоків (до 5000 МВт) можливе в разі запровадження так званих криогенних генераторів, які охолоджуються зрідженим гелієм. Знизити питому витрату палива вдається також підвищенням ККД генераторів ТЕС. Нині максимальне значення ККД становить близько 40 %, але в принципі його можна збільшити до 60 % за рахунок упровадження перспективних магнітогідродинамічних (МГД) генераторів, дослідні зразки яких сьогодні випробовуються в ряді країн.

Спалювання мінерального палива супроводжується сильними забрудненнями довкілля. Розглянемо головні з них. Забруднення атмосфери газовими й пиловими викидами. Під час спалювання вуглеводневого палива в топках ТЕС, а також у двигунах внутрішнього згоряння виділяється вуглекислий газ, концентрація якого в атмосфері збільшується приблизно на 0,25 % за рік. Це спричинює розігрівання атмосфери за рахунок парникового ефекту. З труб ТЕС і вихлопних труб автомобілів у атмосферу викидаються

також оксиди сірки й азоту, внаслідок чого випадають кислотні дощі. Атмосфера забруднюється й дрібними твердими частинками золи, шлаку, не повністю згорілого палива (сажі).

Для зменшення шкоди від цих забруднень вдаються до таких технологічних заходів:

- Вугілля перед його спалюванням у топках ТЕС очищають від сполук сірки;
- Вловлюють із диму ТЕС оксиди сірки й азоту, пропускаючи його крізь спеціальні поглиначі;
- Частинки золи й сажі вловлюють за допомогою установок типу «Циклон» та іншими способами;
- Для зменшення токсичності вихлопних газів автомобілів застосовують регулювання двигунів, переходять на «екологічно чисті» марки палива, встановлюють на автомобілях спеціальні каталізатори, що допалюють чадний газ до вуглекислого, і т. д.

Після спалювання в топках ТЕС вугілля залишається багато твердих відходів (шлаку, золи). Вони забирають великі площі землі, забруднюють підземні й поверхневі води шкідливими речовинами. Ще більші ділянки землі порушуються величезними вугільними кар'єрами. Так, шлакові відвали й терикони пустих порід лише в Донбасі займають площу понад 50 тис. га, і вона дедалі збільшується. Зменшення шкоди від такого забруднення досягається утилізацією (корисним використанням) шлаків і пустих порід, з яких виготовляють будівельні матеріали, засипають ними яри, болота й кар'єри під час рекультивації. Ефективними є й економічні санкції, зокрема введення високої платні за порушення земель, особливо родючих. Завдяки цим обмеженням у більшості західних країн відмовилися від кар'єрного способу видобування корисних копалин у сільськогосподарських районах, оскільки платня за землю виявляється вищою, ніж та вигода, яку може дати

відкритий спосіб розробки родовища порівняно з шахтним. В Україні питання про відведення сільськогосподарських земель під будівництво великого кар'єру або ТЕС вирішує найвищий законодавчий орган країни — Верховна Рада.[11]

ЗАГАЛЬНІ ВИСНОВКИ ДО ДИПЛОМНОЇ РОБОТИ

В ході роботи магістерської роботи були вирішені всі поставлені задачі. А саме була сконструйована модель робота руки-маніпулятора, передача даних і керування якою була реалізована за допомогою протоколу повідомлень MQTT. Також для моделі робота був написаний програмний код, за допомогою якого здійснюється керування маніпулятора. ПЗ протестоване з використанням локального брокера, реалізованого на локальним ПК і смартфоні (MQTT – брокер Mosquitto), MQTT – клієнта, а саме додаток на ОС Android «mqtt dashboard» встановлений на особистий смартфон.

Була проведена порівняльна характеристика найпопулярніших моделей роботів маніпуляторів, а також зроблений підбір сервоприводів. Враховуючи те що використання роботів маніпуляторів це шлях до автоматизації процесів на виробництві, а також їх контролю, вважаю цю тему досить актуальною. Окрім вище згаданого даний макет маніпулятора налагоджено для подальшої роботи і для ознайомлення у навчальних цілях

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ардуіно [Електронний ресурс]//Сайт<https://uk.wikipedia.org> - Режим доступу : <https://uk.wikipedia.org/wiki/Arduino>
2. Сервоприводи [Електронний ресурс]//Сайт [princeton.edu](http://www.princeton.edu) - Режим доступу : <http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>
3. Робототехника:сервоприводы [Електронний ресурс]//Сайт <https://http://wiki.amperka.ru> - Режим доступу : <http://wiki.amperka.ru/>
Робототехника:сервоприводы
4. Wikipedia [Електронний ресурс]//Сайт<https://uk.wikipedia.org> - Режим доступу : <https://uk.wikipedia.org/wiki/ESP8266>
5. Протокол обмеженого застосування [Електронний ресурс]//Сайт [https:// en.wikipedia.org](https://en.wikipedia.org) - Режим доступу :
https://en.wikipedia.org/wiki/Constrained_Application_Protocol
6. Ардуіно [Електронний ресурс]//Сайт<https://uk.wikipedia.org> - Режим доступу : <https://uk.wikipedia.org/wiki/Arduino>
7. Что такое MQTT и для чего он нужен в IoT? Описание протокола MQTT [Електронний ресурс]//Сайт <https://ipc2u.ru/> - Режим доступу :
<https://ipc2u.ru/articles/prostye-resheniya/chto-takoe-mqtt/>
8. Техніка безпеки при роботі на ПК [Електронний ресурс]//Сайт <https://studopedia.info> - Режим доступу : <https://studopedia.info/3-51426.html>
9. Техніка безпеки при користуванні електроприладами [Електронний ресурс]//Сайт <https://buklib.net> - Режим доступу :
<https://buklib.net/books/30658/>
10. Комп'ютер в офісі і його екологічна безпека [Електронний ресурс]//Сайт <http://ur.co.ua> - Режим доступу : <http://ur.co.ua/112/219-1-komp-yuter-v-ofise-i-ego-ekologicheskaya-bezopasnost.html>

11. Энергетика [Электронный ресурс]//Сайт <http://www.info-library.com.ua> - Режим доступа : <http://www.info-library.com.ua/books-text-2694.html>