

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Електронної та біомедичної інженерії \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Кафедра мікроелектроніки, електронних приладів та пристроїв \_\_\_\_\_  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_  
\_\_\_\_\_ Розробка системи керування екзоскелетом \_\_\_\_\_  
\_\_\_\_\_ на базі мікроконтролера \_\_\_\_\_  
(тема)

Виконав:

студент \_4\_ курсу, групи \_\_\_ЕЕПС–21–1\_\_\_  
\_\_\_\_\_ Бордунов Олександр Андрійович \_\_\_\_\_  
(прізвище, ініціали)

Спеціальність \_\_\_171 Електроніка\_\_\_  
\_\_\_\_\_  
(код і повна назва спеціальності)

Тип програми \_\_\_освітньо-професійна\_\_\_  
Освітня програма: «Електронні пристрої та \_\_\_  
системи» \_\_\_\_\_  
( повна назва освітньої програми)

Керівник ст. викл. каф. МЕЕПП \_\_\_\_\_  
\_\_\_\_\_ Васильєв Ю.С. \_\_\_\_\_  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

\_\_\_\_\_ Бондаренко І.М. \_\_\_\_\_  
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Електронної та біомедичної інженерії \_\_\_\_\_

Кафедра \_\_\_\_\_ Кафедра мікроелектроніки, електронних приладів та пристроїв \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Спеціальність \_\_\_\_\_

(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Освітня програма \_\_\_\_\_

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові: Бордунову Олександрю Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи керування екзоскелетом на базі мікроконтролера

затверджена наказом університету від 31.05 \_\_\_\_\_ 2025\_ р. № 726 Ст \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 2025\_ р.

3. Вихідні дані до роботи \_\_\_\_\_

STM32F103C8T6

Датчики м'язової активності MyoWare, AD8232, INA333, AD620

IRS2184

Абсолютні енкодери AS5600 , AS5047, AS5048, MT6701 , MA730

Датчики струму INA240, ACS712, ACS711, ACS758

Пневматичні розподільники 5/2

Датчик тиску USP-G41-1.2

4. Перелік питань, що потрібно опрацювати в роботі: 1.Аналіз існуючих рішень та типів

приводів. 2. Аналіз елементної бази, вибір та обґрунтування вибору компонентів

3. Розробка програмного забезпечення 4. Розробка апаратних рішень

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

слайд-презентація (14 слайдів), схема електрична принципова \_\_\_\_\_

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	05.03.2025	
2	Аналіз зібраної інформації	20.03.2025	
3	Оформлення пояснювальної записки	30.03.2025	
4	Розробка апаратної частини	10.04.2025	
5	Розробка програмного забезпечення	20.05.2025	
6	Оформлення презентації	25.05.2025	
7	Рецензування, нормоконтроль	01.06.2025	
8	Захист роботи	11.06.2025	

Дата видачі завдання \_\_\_\_\_ 2025 р.

Студент \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи бакалавра містить: 58 сторінок, 75 рисунків, 2 додатка, 20 джерел.

ЕКЗОСКЕЛЕТ, ЕЛЕКТРОМІОГРАФІЯ, СЕРВОМОТОР, ПНЕВМАТИКА,

Об'єкт дослідження – активний екзоскелет.

Мета роботи – розробити систему керування екзоскелетом на базі мікроконтролера

Метод – аналітичний, експериментальний.

Актуальність: сьогодні сучасні технології вдосконалюють всі сфери життя і діяльності людини. Окрім всього іншого, технології здатні вдосконалити поточні фізичні можливості або ж повернути втрачені. Один із способів – використання екзоскелетів – зовнішнього каркасу для тіла з приводами(активний екзоскелет) або без них(пасивний). Пасивні екзоскелети досить широко використовуються для полегшення роботи, що пов'язана з фізичними навантаженнями. Активні екзоскелети для такого використовуються значно рідше, проте частіше використовуються для реабілітації або відновлення втрачених фізичних можливостей, наприклад у людей з паралізованими кінцівками. Роботизовані(біонічні) екзопротези, на відмінну від екзоскелетів, є не каркасами, а заміниками відсутньої кінцівки (протезами), проте використовують схожі технології, принципи керування та приводи. Також екзоскелети мають перспективи у військовій справі. Тобто екзоскелети мають свою сферу застосування та перспективи, тому дана робота є актуальною.

## ABSTRACT

The explanatory note of qualification work includes: 58 pages, 75 figures, 2 appendixes, 20 sources.

### EXOSKELETON, ELECTROMYOGRAPHY, SERVOMOTOR, PNEUMATICS

The object of the study is an active exoskeleton.

The purpose of the work is to develop an exoskeleton control system based on a microcontroller

The method is analytical, experimental.

Relevance: today, modern technologies improve all areas of human life and activity. Among other things, technologies are able to improve current physical capabilities or return lost ones. One of the methods is the use of exoskeletons - an external frame for the body with drives (active exoskeleton) or without them (passive). Passive exoskeletons are widely used to facilitate work associated with physical exertion. Active exoskeletons are used much less often for this, but are more often used for rehabilitation or restoration of lost physical capabilities, for example, in people with paralyzed limbs. Robotic (bionic) exoprostheses, unlike exoskeletons, are not frames, but substitutes for a missing limb (prostheses), but use similar technologies, control principles and drives. Exoskeletons also have prospects in military affairs. That is, exoskeletons have their own scope and prospects, so this work is relevant.

## ЗМІСТ

ВСТУП	6
1 АНАЛІТИЧНИЙ ОГЛЯД	10
1.1 Огляд існуючих рішень	10
1.2 Огляд датчиків та приводів	13
2 Аналіз та вибір електронних компонентів	20
2.1 Мікроконтролер	20
2.2 Датчики для керування	21
2.2.1 Датчики м'язової активності	21
2.2.2 Інші способи керування	25
2.3 Електродвигуни та драйвери	25
2.4 Датчики положення	29
2.5 Датчики струму	33
2.6 Пневматичні розподільники	36
2.7 Датчик тиску	39
3 Програмна частина	40
3.1 Основна програма	40
3.1.1 Отримання та обробка сигналів з міодатчиків	40
3.1.2 Робота з колекторними двигунами	45
3.1.3 Робота з безколекторними двигунами	47
3.1.4 Робота з пневматичними приводами	49
3.1.5 Отримання даних від конфігуратора та налаштування	52
3.1.6 Додаткові функції	54
3.2 Конфігуратор	55
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60

ДОДАТОК А	61
ДОДАТОК Б	62
ДОДАТОК В	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І  
ТЕРМІНІВ

АЦП – Аналогово-цифровий перетворювач

ЕМГ – Електроміографія

ПІД – Пропорційно-інтегрально-диференціальний алгоритм

ШІМ – Широтно-імпульсна модуляція

ВОМ – Bill of materials

FOC – Field oriented control (полеорієнтоване керування)

I2C – Inter-integrated circuit

SPI – Serial peripheral interface

## ВСТУП

Конструкції подібні до сучасних пасивних екзоскелетів з'явилися наприкінці XIX– початку XX століття. Хоч вони і не використовували назву «екзоскелет», проте мали схожі конструкції та виконували ті самі функції – переносили або розподіляли навантаження, тобто зменшували його на окремих частинах тіла.

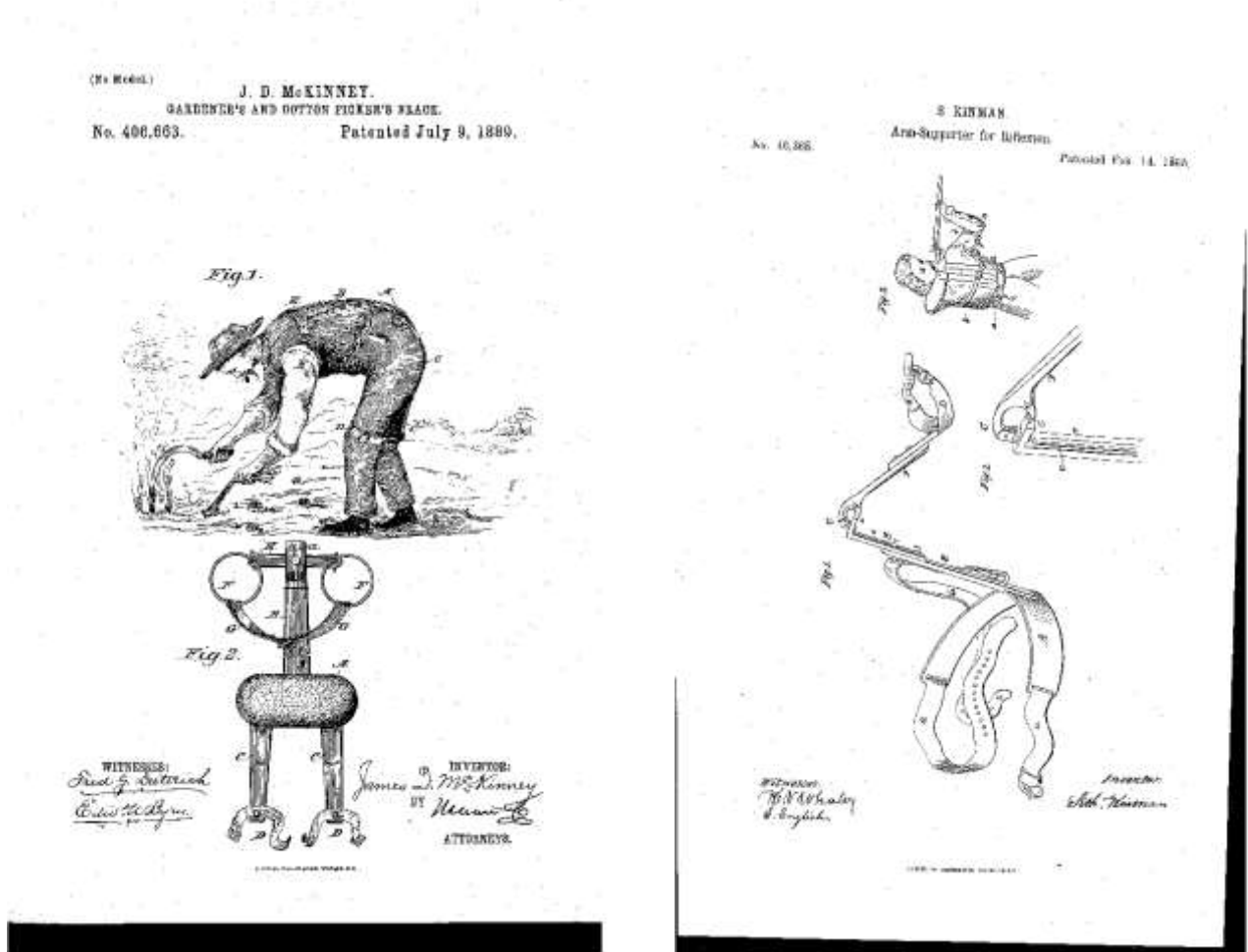


Рисунок – Конструкції для перенесення або розподілу навантаження XIX ст., подібні до сучасних пасивних екзоскелетів

Один з перших проектів машини подібної до активного екзоскелета з'явився на початку XX століття, хоча ця конструкція й не була реалізована на практиці.

Вона складалася з парової машини і тросів які з'єднують її з ногами. За задумкою це повинно було полегшувати ходьбу.

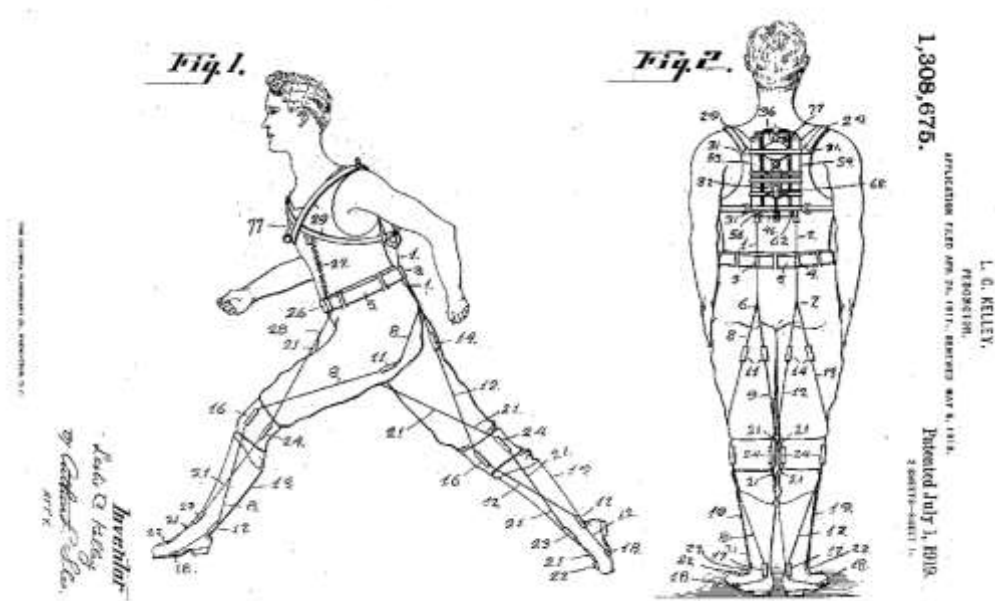


Рисунок – Парова машина, подібна до екзоскелета

Першим активним екзоскелетом створеним на практиці у 1965 - 1971 р.р. був Hardiman від General Electric. Проект виявився невдалим, бо конструкція була не стійкою та майже неконтрольованою, через що жодного разу не запускалася з людиною всередині. Окрім того конструкція була громіздкою - вона важила близько 680 кілограмів.

У 1967 році був готовий перший прототип. Він приводився в дію гідромеханічними сервоприводами, однак вони працювали нестабільно і тому у 1968 р. були замінені на електрогідравлічні сервоприводи.

У 1969 році була налагоджена рука, в 1970 – ноги і пояс, проте конструкція виявилася нестабільною – вона дуже погано тримала рівновагу, отже була нездатна ходити. Після змін в конструкції і з допомогою підтримуючого тросу це все ж таки вдалося.



Рисунок – Hardiman, перший в світі активний екзоскелет

Після Hardiman екзоскелети почали розробляти активніше. З'явилося багато варіантів від різних розробників, наприклад проект екзоскелету для військових під назвою Pitman, хоч він так і не був реалізований через високу складність конструкції. Деякі проекти все ж були реалізовані і навіть використовувалися на практиці вже тоді, наприклад медичні екзоскелети, розроблені командою Міоміра Вукобратовича в 1970 роках.

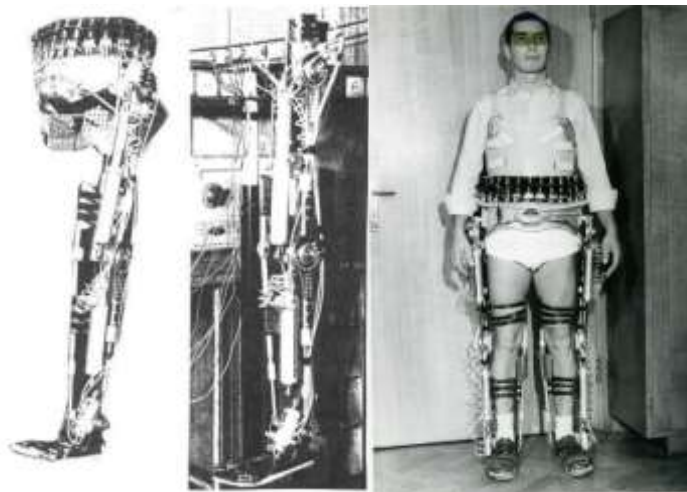


Рисунок – Екзоскелет створений командою Міоміра Вукобратовича

В 1986 Монті Рід отримав перелом спини через невдалий стрибок з парашутом і почав розробку власного екзоскелету. В 2001 році він створив перші прототипи екзоскелета під назвою Lifesuit та заснував організацію «They Shall Walk». На сьогоднішній день цей проєкт існує та розвивається.



Рисунок – Екзоскелет Lifesuit

В XXI столітті технології стали більш розвинутими та головне – доступними, тому з'явилося багато нових варіантів екзоскелетів, причому як від інженерів, так і від ентузіастів.



Рисунок – Сучасний екзоскелет ReWalk

# 1 АНАЛІТИЧНИЙ ОГЛЯД

## 1.1 Огляд існуючих рішень

На сьогоднішній день головною сферою використання активних (роботизованих) екзоскелетів є медицина (реабілітація, відновлення втрачених функцій кінцівок, також сюди можна віднести роботизовані протези). Тому більшість комерційних активних екзоскелетів є саме медичними.

Перший прототип екзоскелету HAL (Hybrid Assistive Limb) від японської компанії Cyberdyne з'явився у 1997 році. В 2008 році почалося практичне використання екзоскелетів для медичних задач в Японії. В 2013 році вони отримали сертифікати, що дозволяють використання для медичних цілей в Європі.



Рисунок 1.1 – Екзоскелети Cyberdyne HAL

Існує декілька видів екзоскелетів HAL, такі як екзоскелет для ніг(однієї або обох), для одного суглоба, для поясу. Ці конструкції має жорстку основу та м'які кріплення до кінцівок, приводяться в дію за допомогою сервоприводів, керуються за допомогою міодатчиків та контролеру. Привід має три режими керування:

- Cybernic Voluntary Control (CVC);
- Cybernic Autonomous Control (CAC)
- Cybernic Impedance Control (CIC)



Рисунок 1.2 – Контролери екзоскелетів Cyberdyne HAL

В режимі Cybernic Voluntary Control (CVC) міодатчики визначають положення і стан кінцівки або тіла та на основі цієї інформації приводи змінюють положення. Контролер додатково підтримує положення для ходьби, стояння чи сидіння.

В режимі Cybernic Autonomous Control (CAC) використовуються заздалегідь запрограмовані патерни ходьби, контролер реагує на зміну положення тіла.

Режим Cybernic Impedance Control (CIC) компенсує вплив тертя та інших сил.

Для використання екзоскелету Cyberdyne HAL на кінцівку мають приходити біоелектричні сигнали, тобто цей пристрій підходить для людей без проблем зі здоров'ям(для збільшення сили та витривалості), або для людей з частково функціонуючими кінцівками. Він не підходить для використання людьми з паралізованими кінцівками.

Екзоскелети ReWalk американської компанії ReWalk Robotics має інший принцип керування та призначений для використання людьми з травмами спини та паралічем нижніх кінцівок. Конструкція подібна до японського HAL, проте

керування виконано не за допомогою датчиків м'язової активності, що додає певні переваги та обмеження.



Рисунок 1.3 – Екзоскелет ReWalk 6.0

Керування розроблено спеціально для людей з вадами нижніх кінцівок і виконується за допомогою кнопок на зап'ясті і положення тулуба.



Рисунок 1.4 – Наручний годинник з кнопками для керування екзоскелетом  
ReWalk

Цей екзоскелет є недешевою, проте цікавою заміною інвалідного візка. Він дозволяє ходити людям з проблемами з нижніми кінцівками чи шиєю, пом'якшує вторинні ускладнення, пов'язані з травмами спинного мозку, дозволяє підійматися та спускатися сходами. Він має суто медичне призначення та, на відмінну від HAL,

непридатний для використання з метою покращення фізичних можливостей здорової людини.

Екзоскелет Ekso Indego від компанії Ekso Bionics конструктивно схожий на попередні, але налаштовується через мобільний додаток і керується виключно нахилами тулуба, що накладає деякі обмеження.



Рисунок 1.5 – Екзоскелет Ekso Indego

Цей екзоскелет також призначений для реабілітації або розширення можливостей людей з травмами спини та нижніх кінцівок, але через особливості керування цей екзоскелет поступається ReWalk, який має додаткові органи керування та дозволяє, наприклад, підійматися та спускатися сходами, чого не може Ekso Indego.

## 1.2 Огляд датчиків та приводів

Сучасні технології, елементна база та пристрої пропонують декілька різних варіантів для керування та приводів екзоскелетів.

Одним з кращих рішень для керування екзоскелетом – використання електрографічних датчиків. Людське тіло передає інформацію та керується за допомогою електричних імпульсів. За допомогою датчиків можна відслідковувати ці імпульси та отримувати корисну інформацію про м'язову, мозкову та деякі інші активності.

Процес реєстрації електричних сигналів м'язової активності має назву електроміографія (ЕМГ). Сигнали з м'язів це зручний метод керування екзоскелетом, для нього майже не потрібно виконувати зайвих дій чи рухів. Система відслідковує активність м'язів і на основі цієї інформації дає команди штучним м'язам екзоскелета.

Існує два основних способи з'єднання тіла з контроллером: голчата міографія та поверхнева.

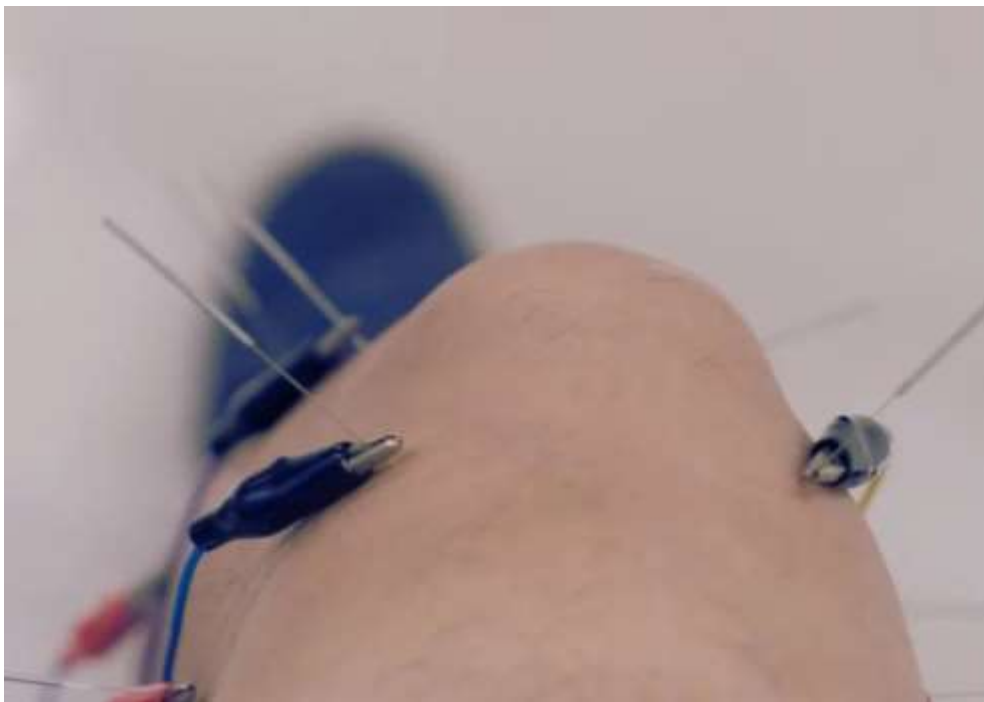


Рисунок 1.6 – Голчата електроміографія

При використанні голчатих електродів з'єднання виходить більш якісним та стабільним, якість сигналу теж покращується. Проте цей метод потребує введення сторонніх предметів до тіла, тому не підходить для тривалого або частого використання, проте підходить для діагностики захворювань м'язів.

Електричні сигнали можна отримувати з поверхні шкіри, без втручання в тіло. Для цього потрібні нашкірні електроди, бажано з дуже доброю електричною провідністю.

Електроди для таких задач бувають сухі або гелеві. Сухі електроди не потребують ніяких додаткових речовин для роботи, для гелевих потрібен твердий(в'язкий) або рідкий гель.



Рисунок 1.7 – Нашкірний електрод для електрографії  
(електрокардіографії, електроміографії тощо)

Сигнал, отриманий з електродів, має амплітуду близько 50 мкВ – 30 мВ і є двополярним. Для роботи з мікроконтролером сигнал потрібно підсилити в декілька сотень разів і привести до однополярного. Для цього використовують різні підсилювачі.

Переваги таких датчиків – зручність керування. Ніякі додаткові органи керування не потрібні, тільки рухи цільовими м'язами. Проте такий метод не завжди підходить для використання в медичних цілях, бо потребує наявності хоча б частково функціонуючих м'язів, тому деякі існуючі рішення(такі як ReWalk та Ekso Indego) не мають датчиків м'язової активності і керуються іншими методами.





Рисунок 1.9 – Модуль з датчиком положення та прискорення

Для приведення механізмів в дію теж існує декілька різних технологій. Частина з них широко використовується в комерційних продуктах, частина мають ситуативне призначення, деякі – експериментальні, але перспективні.

Сервомотори – електричні двигуни(частіше за все з редукторами) зі зворотнім зв'язком та автоматичним регулюванням одного або декількох параметрів (положення, швидкість, сила). Існують готові рішення, окрім того будь-який електричний двигун може бути модифікований до серводвигуна. Сучасні електродвигуни достатньо компактні, потужні, плавні і швидкі, тому найчастіше у існуючих рішеннях використовуються саме вони. Головний недолік – необхідність використання додаткової електроніки, в тому числі силової, та достатньо високий робочий струм що також створює додаткові вимоги до джерела живлення.

Незважаючи на всі недоліки, саме цей тип приводу є найпопулярнішим серед існуючих рішень, що активно використовуються на практиці.



Рисунок 1.10 – Серводвигун

Пневматичні та гідравлічні приводи мають іншу конструкцію та принцип дії, що відкриває нові можливості, але і накладає нові обмеження. Конструктивно вони є циліндрами з поршнями, положення яких регулюється шляхом подачі рідини або стисненого газу. Цей принцип руху більше схожий на роботу скелетних м'язів, ніж обертання двигуна. Окремо можна виділити приводи, що навіть конструктивно схожі на м'язи, що ще більше виділяється на тлі електродвигуна.

Обидва варіанти потребують компресор(принципи дії пневматичних та гідравлічних компресорів відрізняються), ресивер, систему трубок для підведення робочого тиску та керуючі механізми(клапани, розподільники тощо), тобто фінальний пристрій буде більш громіздким за конструкцію з електродвигунами. Проте такі приводи забезпечують кращу передачу механічної енергії, тобто здатні витримувати більші навантаження, ніж електродвигун. Тобто екзоскелет с пневматичними або гідравлічними м'язами буде набагато більшим ніж аналогічний з електроприводами, проте забезпечить більшою силою та вантажопідйомністю. Один з перших екзоскелетів, що використовувався на практиці і встановив рекорд зі швидкості ходьби – Lifesuit – використовував саме пневматику.



Рисунок 1.11 – Пневматичний циліндр



Рисунок 1.12 – Робот зі штучними (пневматичними) м'язами

Більш сучасні технології дозволяють створити штучні м'язи ще більш подібні до органічних, з ще більшою ефективністю та зручнішим керуванням. Одне з таких рішень – штучні м'язи на основі карбонових нанотрубок або нейлону. М'язи такої конструкції керуються зміною температури – скорочуються при нагріванні та розпрямляються при охолодженні. Конструкція зручніше і ефективніше за пневматику, проте на сьогоднішній день є експериментальною і не використовується на практиці.

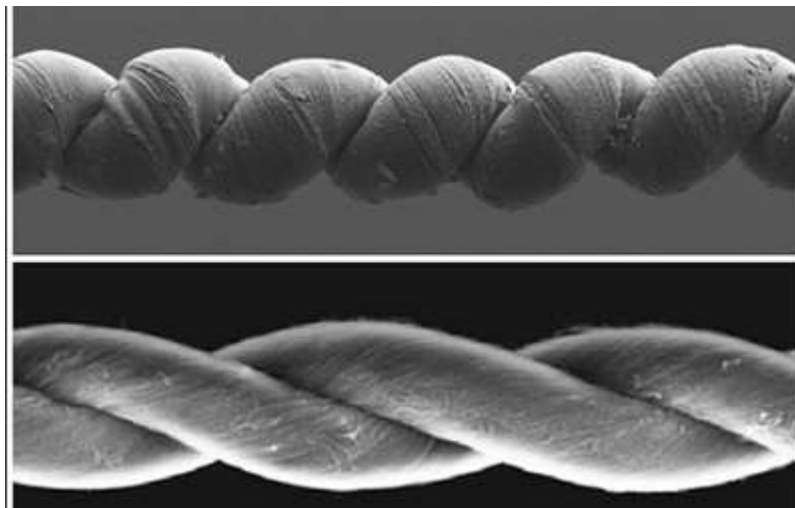


Рисунок 1.13 – Штучні м'язи на основі полімерних матеріалів під мікроскопом

## 2 АНАЛІЗ ТА ВИБІР ЕЛЕКТРОННИХ КОМПОНЕНТІВ

### 2.1 Мікроконтролер

На сьогоднішній день існує велике різноманіття мікроконтролерів від різних виробників. Одне з найпопулярніших сімейств – STM32 від STMicroelectronic.

Для проекту було обрано мікроконтролер STM32F103C8T6. Характеристики:

- тактова частота – до 72 МГц;
- Flash – 64 Кб;
- SRAM – 20 Кб;
- напруга живлення - 2...3.6В ;
- GPIO – 37 шт.;
- АЦП – 2 шт., 12 біт, 10 каналів;
- Інтерфейси - 2xSPI, 2xI2C, 3xUSART, USB, CAN;
- 7-канальний DMA контролер;
- 3 універсальних таймера і 1 advanced таймер;
- 16 каналів ШІМ;
- SWD і JTAG інтерфейс.

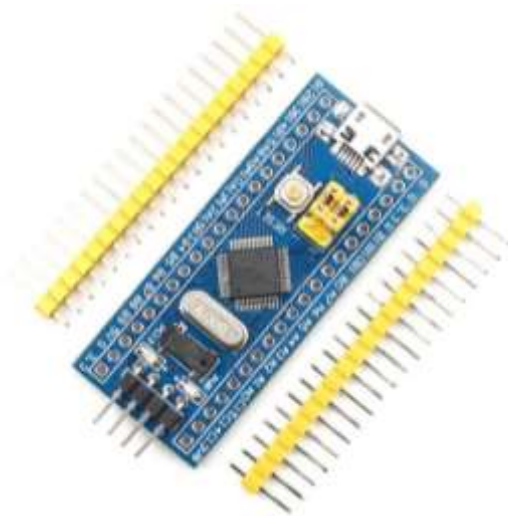


Рисунок 2.1 – Плата розробника Bluepill на основі мікроконтролера STM32F103C8T6

## 2.2 Датчики для керування

### 2.2.1 Датчики м'язової активності.

Основним способом керування екзоскелетом є отримання сигналів м'язової активності та подальша їх обробка. Для отримання сигналів використовують спеціальні електроди та підсилювачі сигнал.

Електроди можуть бути виготовлені з будь-якого провідникового матеріалу, але краще за все використовувати матеріали з високою електричною провідністю.

Електроди для ЕКГ мають покриття з хлориду срібла, що покращує характеристики сигналу, і підходять і для інших методів електрографії, в тому числі і електроміографії.



Рисунок 2.2 – Електроди покриті хлоридом срібла

Такі електроди використовуються з рідким або в'язким гелем. Якщо планується багаторазове використання, то краще використовувати в'язкий. Проте навіть в'язкий гель з часом висихає та змінює свої властивості, тому електроди краще використовувати взагалі без гелю.

Сигнал від м'язу має амплітуду близько від 50 мкВ до 30 мВ, і до того ж є двополярним, тому не може бути прочитаний мікроконтролером без попереднього підсилення та обробки. Для цього існує декілька популярних рішень.

Модулі MyoWare одне з найкращих, проте і найдорожчих рішень для підсилення сигналів з електродів.

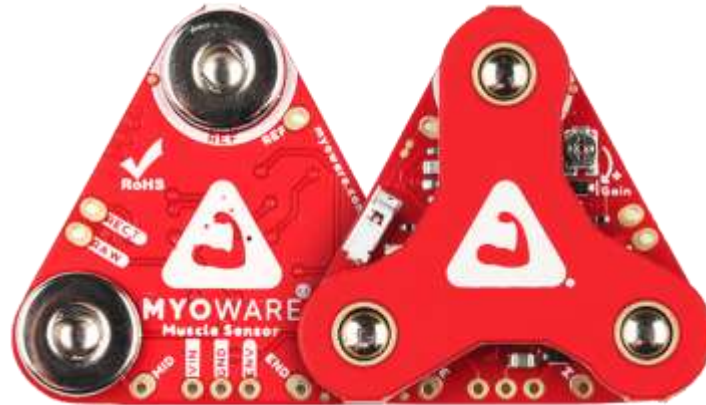


Рисунок 2.3 – Модуль MyoWare

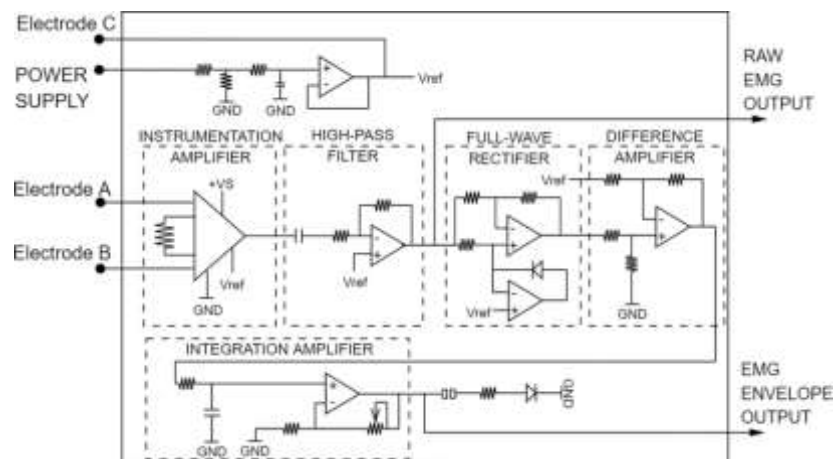


Рисунок 2.4 – Блок схема модулю MyoWare

В його основі – rail-to-rail операційний підсилювач AD8648. Цей модуль має регулювання рівня підсилення та додаткові функції, а саме фільтр та випрямлювач сигналу.

Модуль на основі rail-to-rail операційного підсилювача AD8232 дешевший, окрім функції підсилення має фільтри та детектор контакту електродів – два виводи що подають високий рівень коли один з електродів від'єднаний. Цей модуль не має регулювання рівня підсилення.

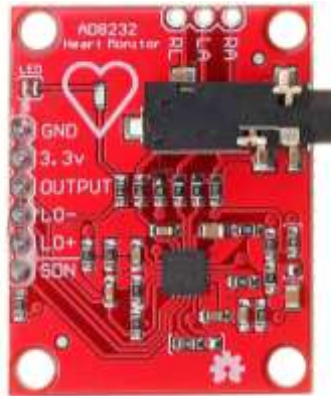


Рисунок 2.5 – Модуль AD8232

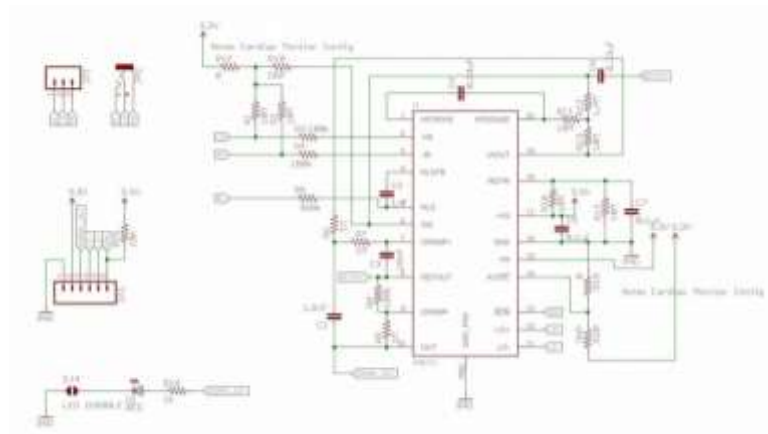


Рисунок 2.6 – Принципова схема модулю AD8232

Модуль на основі операційного підсилювача INA333 – одне з найдешевших, проте дієвих рішень. Він має регулювання рівня підсилення та не має додаткових функцій.

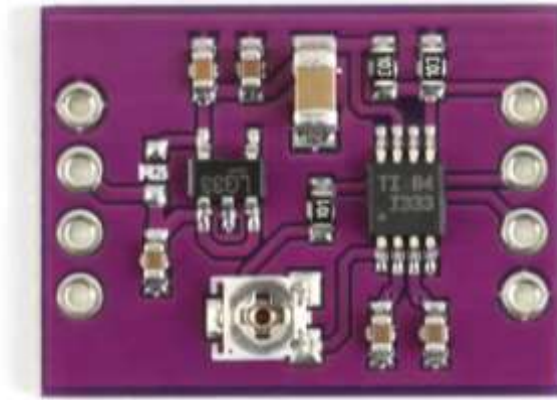


Рисунок 2.7 – Модуль INA333

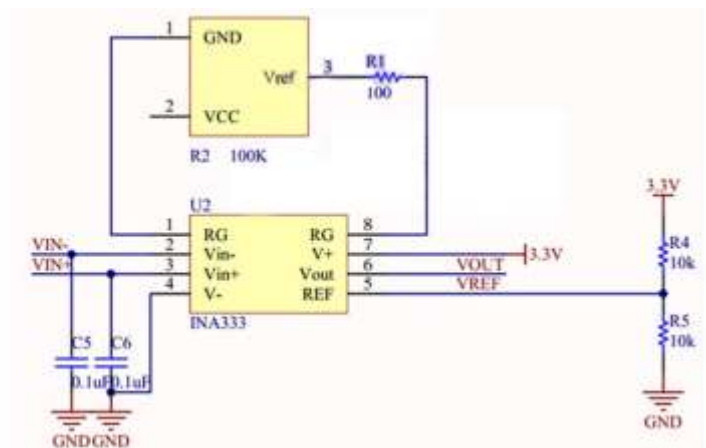


Рисунок 2.8 – Принципова схема модулю INA333

Модуль AD620 - ще одне бюджетне рішення. Він має регулювання ступеню підсилення, а також регулювання зміщення сигналу.



Рисунок 2.9 – Модуль AD620

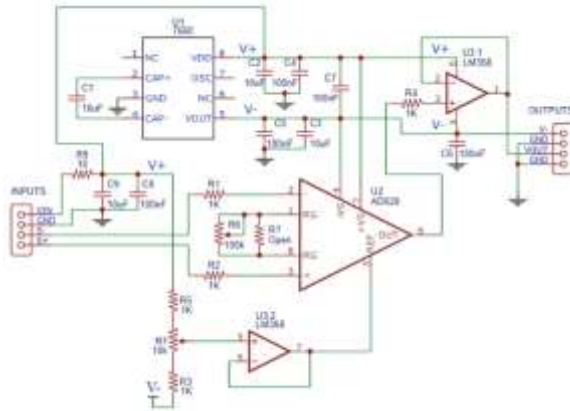


Рисунок 2.10 – Принципова схема модулю AD620

Всі ці модулі мають невеликі відмінності, проте всі мають аналоговий вихід і тому можуть бути використані в проекті з однаковим підключенням і налаштуванням в програмі. Для демонстрації буде використано модуль AD620.

### 2.2.2 Інші способи керування.

Деякі існуючі моделі екзоскелетів, наприклад ReWalk, мають альтернативні методи керування, призначені в основному для людей з обмеженими можливостями. Тому не зайвим буде додати цю функцію в цей проект.

Для керування може бути використаний будь-який змінний резистор, джойстик або абсолютний(магнітний) енкодер.

## 2.3 Електродвигуни та драйвери.

Як було сказано в аналітичному розділі, найпопулярніший тип приводу для екзоскелетів – серводвигуни. Проте знайти готові рішення відповідної потужності майже неможливо. Втім, будь-який електричний двигун може бути модифікований до сервопривода. В цьому розділі розглянуто вибір електродвигуна та драйверу до нього для цієї задачі.

Контролер підтримує двигуни двох типів – колекторні та безколекторні. Безколекторні двигуни тихіші, потужніші та ефективніші, проте для деяких задач достатньо і колекторного двигуна, наприклад для приводу пальців в екзопротезі.

Для роботи з цією системою керування підійде двигун будь-якого з зазначених типів, потужність вибирається відповідно до необхідного навантаження.

Для підключення обох типів двигунів до мікроконтролера необхідний драйвер на два або три напівмости відповідно. Цей драйвер повинен бути розрахований на робочу напругу та струм двигуна. Готових рішень для відповідної задачі майже не існує: всі вони розраховані на недостатньо велику потужність.

Більшість двигунів, розрахованих на автономну роботу(від акумуляторів) працюють при напрузі 12-100 В та струмі в кілька десятків ампер.

Типові схеми підключення двигунів показані на рисунках 2.11-2.12

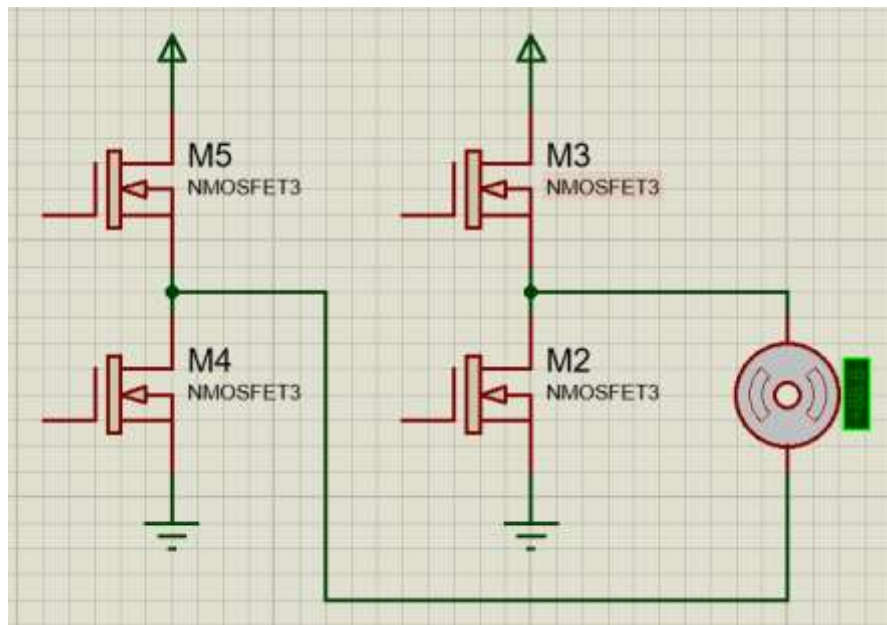


Рисунок 2.11 – Типова схема підключення колекторного(DC) двигуна

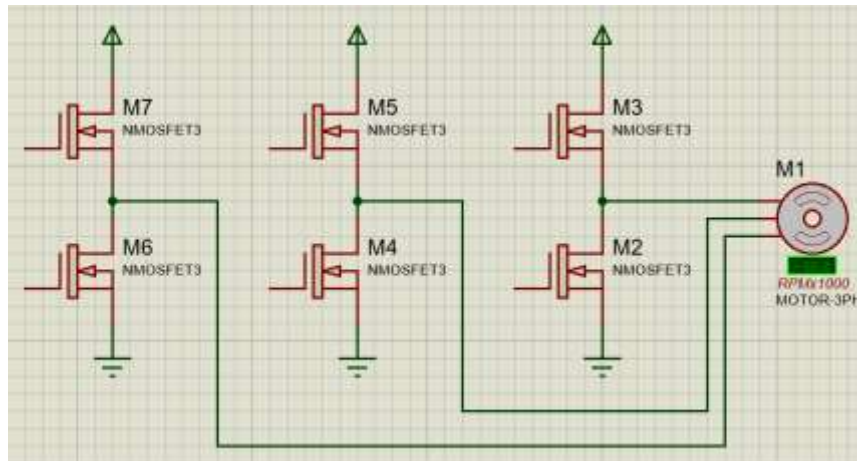


Рисунок 2.12 – Типова схема підключення  
безколекторного(BLDC) двигуна

Для керування транзисторами, що показані на рисунку, необхідна спеціальна мікросхема-драйвер. Для керування кожним з напівмостів можна використовувати два ШІМ канали мікроконтролера або один. Дана система керування використовує другий варіант задля економії каналів, хоча варіант з роздільним керуванням верхнім і нижнім плечем має більшу ефективність.

Також ця мікросхема має забезпечувати достатній струм на виході для відкриття MOSFET або IGBT транзисторів. Хоча вони керуються напругою, для достатньо швидкого їх відкриття необхідний деякий струм. Швидке відкриття транзисторів необхідне для зменшення втрат при роботі на високій частоті. Висока робоча частота(вища за чутний діапазон) необхідна для беззвучної роботи.

Зазначеним вище вимогам відповідає мікросхема IRS2184. IRS2184 – це драйвер напівмоста з апаратним deadtime, робочою напругою до 20В та вихідним струмом до 2А. При необхідності може бути замінена слабкішими рішеннями з цієї лінійки.

Принципова схема та плата драйверу для колекторних та безколекторних двигунів на базі IRS2184 показана на рисунках.

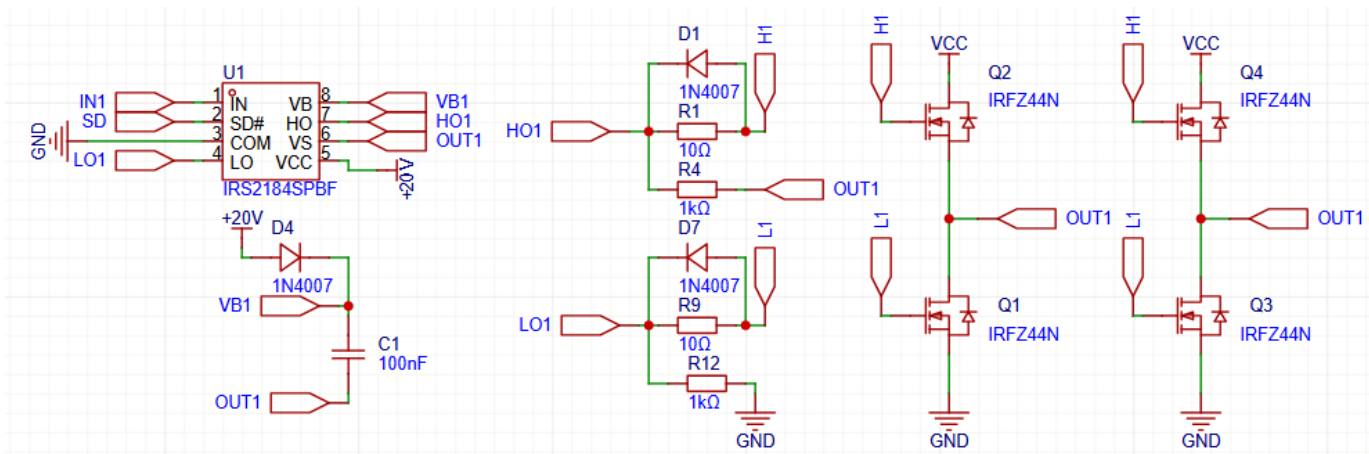


Рисунок 2.13 – Принципова схема драйверу напівмоста на базі IRS2184

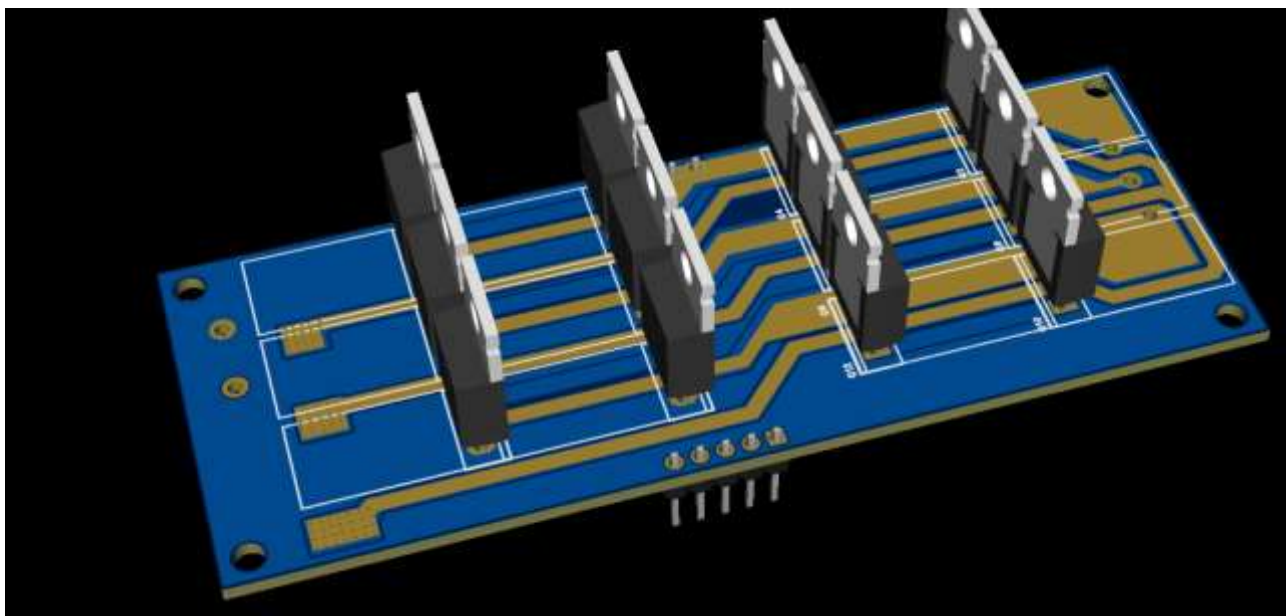
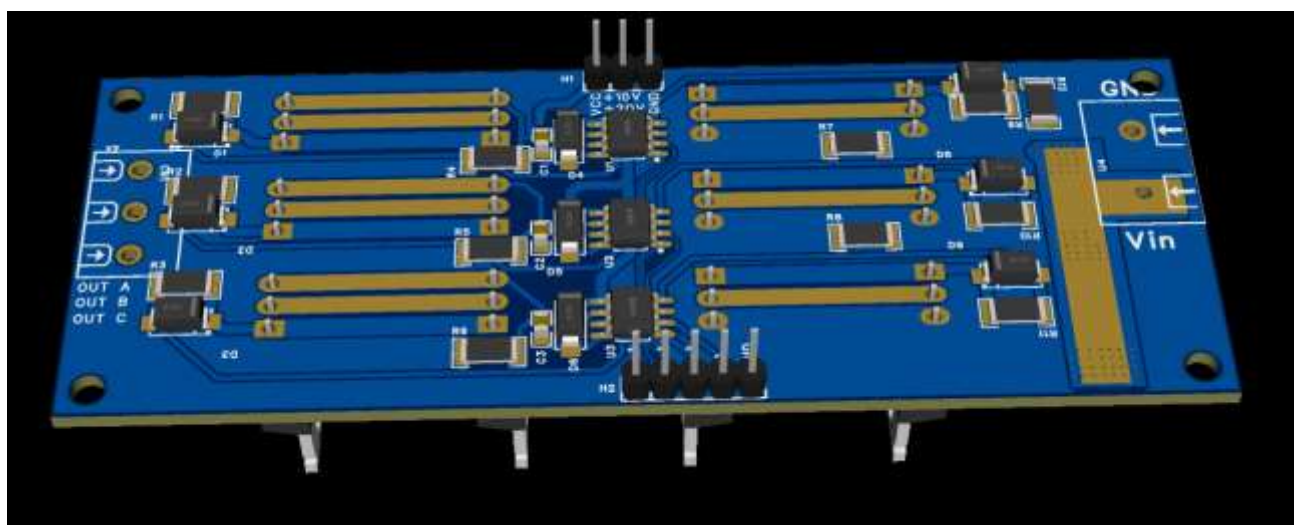


Рисунок 2.14-2.15 – Модель плати драйверу електродвигуна

No.	Quantity	Value	Name	Footprint	Primary Category
1		1	Screw Terminal Block	CONN-TH_2P-P7.62_6	Connectors
2		1	Screw Terminal Block	CONN-TH_3P-P5.08_D	Connectors
3		1	Pin Headers	HDR-TH_5P-P2.54-V-M	Connectors
4		1	Pin Headers	HDR-TH_3P-P2.54-V-M	Connectors
5		3	IRS2184SPBF	SOIC-8_L4.9-W3.9-P1.2	Motor Driver ICs
6		3 100nF	100nF	C0805	Capacitors
7		6 1k $\Omega$	1k $\Omega$	R2010	Resistors
8		6 10 $\Omega$	10 $\Omega$	R2010	Resistors
9		9	1N4007	SMA_L4.4-W2.6-LS5.2-	Diodes
10		12	IRFZ44N	TO-220-3_L9.9-W4.5-P-	Transistors/Thyristors

Рисунок 2.16 – BOM для драйвера електродвигуна

Для прикладу було взято MOSFET транзистори IRFZ44N з максимальною робочою напругою до 55В та максимальним робочим струмом до 49А, тобто кожне плече з двома такими транзисторами в паралелі видає до 100 ампер струму. За необхідності транзистори можуть бути замінені на будь-які інші MOSFET або IGBT транзистори в корпусі TO220.

## 2.4 Датчики положення

Для створення сервоприводу, тобто для точного регулювання положення електродвигуна, необхідно щоб система керування знала це положення. Для цього існують різні види датчиків.

Датчики Холла часто використовуються для керування безколекторними двигунами, проте вони мають невелику точність визначення положення. Цього достатньо для керування обертанням двигуна, але для контролю положення краще використати більш точні рішення.



Рисунок 2.17 – Датчики Холла

Механічні або оптичні енкодери мають більшу точність і цілком підходять для визначення положення електродвигуна. Проте вони, так само як і датчики Холла, відслідковують не положення, а його зміну. Це додає складностей в роботі з датчиком, окрім того при ввімкненні контролер не знатиме положення двигуна, тому це рішення також не є найкращим.



Рисунок 2.18 – Оптичний енкодер

Магнітні абсолютні енкодери є найкращим рішенням для цієї задачі. Принцип їх дії заснований також на ефекті Холла, проте вони мають декілька таких датчиків та додаткову схему для обробки та перетворення сигналів з них. Для їх

роботи необхідний магніт зі спеціально розташованими полюсами. Енкодери цього типу не мають недоліків як у попередніх. Вони забезпечуть високу точність вимірювань, здатні визначити поточне положення двигуна при ввімкненні контролера, мають різні інтерфейси для зв'язку з ним. Розглянемо деякі з доступних рішень цього типу.

AS5600 має 12-бітну роздільну здатність, інтерфейси: аналоговий, ШІМ, I2C(тільки одна незмінна адреса).



Рисунок 2.19 – Модуль на основі мікросхеми AS5600 та комплектний магніт

AS5047 має роздільну здатність 12 біт, інтерфейси: аналоговий, ШІМ, інкрементальний, SPI. AS5048 має також I2C з можливістю зміни адреси. Наявність цих інтерфейсів дозволяє використовувати декілька енкодерів(для керування декількома сервоприводами) без використання аналогових портів, кількість яких невелика і які необхідні для роботи з мідатчиками.

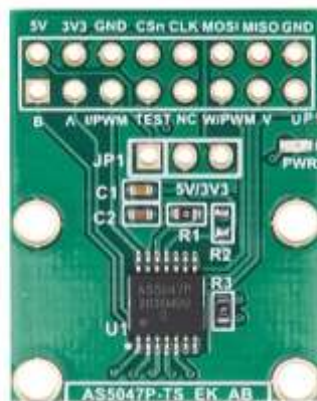


Рисунок 2.20 – Модуль на основі мікросхеми AS5047

MT6701 має роздільну здатність 12 біт, інтерфейси: аналоговий, ШІМ, інкрементальний, I2C, SSI.



Рисунок 2.21 – Модуль на основі мікросхеми MT6701 та комплектний магніт

MA730 має роздільну здатність 14 біт, інтерфейси: аналоговий, ШІМ, інкрементальний, SPI.

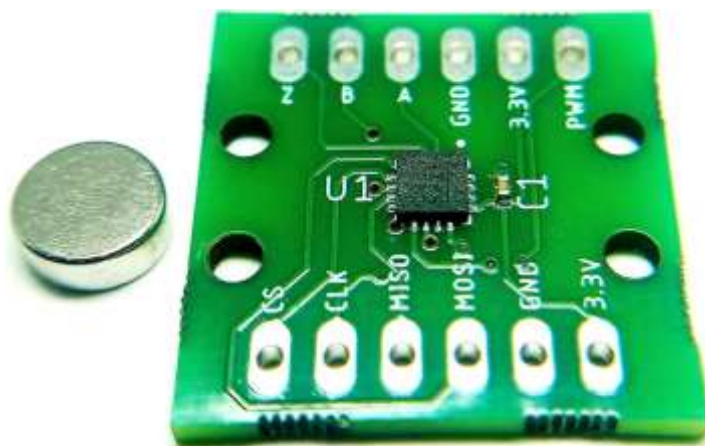


Рисунок 2.22 – Модуль на основі мікросхеми MA730 та комплектний магніт

Як бачимо, всі ці енкодери мають схожі характеристики та інтерфейси. AS5048 має найбільшу кількість інтерфейсів, особливо слід звернути увагу на наявність I2C з можливістю зміни адреси, тобто з можливістю використовувати

декілька таких датчиків за цим інтерфейсом. AS5600, на відмінну від інших, не має інтерфейсу SPI, декілька таких датчиків можливо підключити тільки за аналоговим або ШІМ інтерфейсом.

Ця система керування здатна працювати з усіма вказаними датчиками за інтерфейсами I2C, SPI та аналоговим.

## 2.5 Датчики струму

Для забезпечення захисту від перенавантаження та виходу з ладу двигуна необхідно забезпечити вимірювання та контроль струму. Для цього існує декілька видів датчиків.

Шунт для вимірювання струму – простий, проте ефективний метод вимірювання. Шунт це деякий провідник з відомим опором. Падіння напруги на шунті пропорційне струму, що протікає через нього.



Рисунок 2.23 – Шунт

Напруга на шунті має невелике значення, окрім того для роботи з електродвигуном необхідно вимірювати струм, що протікає в обох напрямках. Для вирішення цих проблем необхідно використовувати монітори шунтів, такі як INA240.

INA240 використовується для підсилення та зміщення сигналу. Коефіцієнт підсилення 20, 50, 100, 200(фіксований, необхідно вибрати модифікацію з відповідним коефіцієнтом)



Рисунок 2.24 – Модуль на основі мікросхеми INA240

Датчики струму трансформаторного типу не вносять додаткового опору в коло та мають гальванічну розв'язку між силовою та логічною частинами, на відмінну від шунта, проте здатні вимірювати тільки змінний струм, а система в проєкті керує моторами постійного струму. Тому ці датчики не підходять для нашої задачі



Рисунок 2.25 – Датчик струму трансформаторного типу

Датчики струму на ефекті Холла поєднують в собі переваги двох попередніх типів датчиків. Вони здатні працювати як з постійним, так і зі змінним струмом як

і шунти та не вносять додатковий опір та мають гальванічну розв'язку як трансформаторні датчики. Розглянемо деякі з них.

Один з найпопулярніших датчиків цього типу - ACS712 мають діапазон вимірювання 5, 20 або 30А в залежності від модифікації. Напруга ізоляції – 180В. Проте діапазон напруги живлення цього датчика від 4,5В до 5В, що впливає на вихідний сигнал та ускладнює роботу з мікроконтролерами з 3,3 вольтовою логікою, а саме такий було вибрано для проекту.



Рисунок 2.26 – Модуль з датчиком ACS712

Датчик ACS711 має всі переваги попереднього, проте напруга його ізоляції 100В, що в теорії може бути недостатньо для роботи з деякими двигунами. На практиці ж краще відмовитись від двигунів з напругою навіть такої величини з метою підвищення електробезпеки. Діапазон живлення від 3,3В до 5,5 В, що підходить для роботи з 3.3 вольтовими мікроконтролерами. Діапазон вимірювання струму – 12,5; 25; 15,5 або 31А в залежності від модифікації.

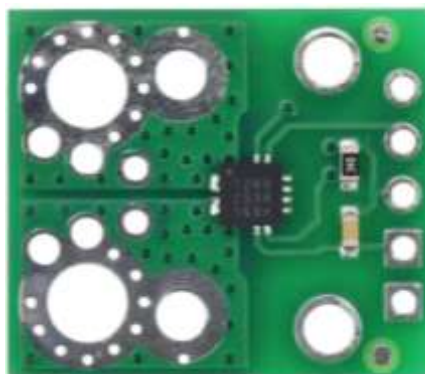


Рисунок 2.27 – Модуль з датчиком ACS711

Якщо діапазону вимірювання недостатньо то можна скористатися ACS758. Його робоча напруга 3.0-5.5 В, напруга ізоляції – до 4800В, діапазон вимірювань – 50, 100, 50, 200А в залежності від модифікації.



Рисунок 2.28 – Модуль з датчиком ACS758

Всі сумісні датчики мають схожий сигнал на виході і принципи взаємодії з мікроконтролером, а отже можуть бути використані в проекті. Серводвигуни можуть працювати і без контролю струму, проте цей спосіб рекомендується використовувати тільки при запуску з мінімально можливим навантаженням, в рамках цього проекту – тільки для тестування.

## 2.6 Пневматичні розподільники

Дана система керування може працювати з декількома різними типами приводів. Один з них – пневматичні циліндри або пневматичні штучні м'язи. Вони приводяться в дію повітрям або іншим газом під тиском. Для керування ними можна використовувати пневматичні розподільники з електромагнітним регулюванням.

Основні типи пневматичних розподільників - 5/2 і 3/2. Розподільник 3/2 здатний керувати тільки одним циліндром або м'язом, 5/2 може керувати циліндром з обох боків або двома парними м'язами. Основний недолік такого типу керування – відсутність плавності або проміжних значень. Клапан розподільника

або відкритий, або закритий. Проте можливо змінювати положення почергово відкриваючи та закриваючи клапан.

Електромагнітні(соленоїдні) розподільники можуть мати в своїй конструкції один або два соленоїди, що переміщують золотник. Якщо електромагніт тільки один, то розподільник повертається в зворотне положення за допомогою пружини.

Для керування розподільником необхідно подати робочу напругу на електромагніт. Робоча напруга електромагніту зазвичай 12В або 24В (модифікації на 220В не розглядаємо, бо екзоскелет має працювати від автономного джерела живлення). Для його керування не потрібні складні драйвери як для двигунів, достатньо підключити до мікроконтролера через транзистор з відповідними робочою напругою та струмом.

Дана система керування розрахована для роботи з пневматичними розподільниками 5/2 або 3/2 NC(normally closed, нормально замкнуті) з одним соленоїдом.



Рисунок 2.29 – Пневматичний розподільник 5/2

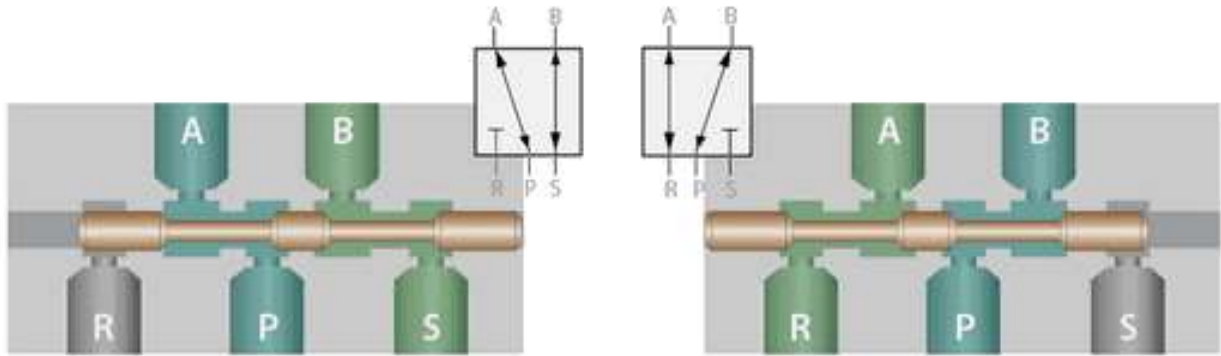


Рисунок 2.30 – Схема пневматичного розподільника 5/2



Рисунок 2.31 – Пневматичний розподільник 3/2

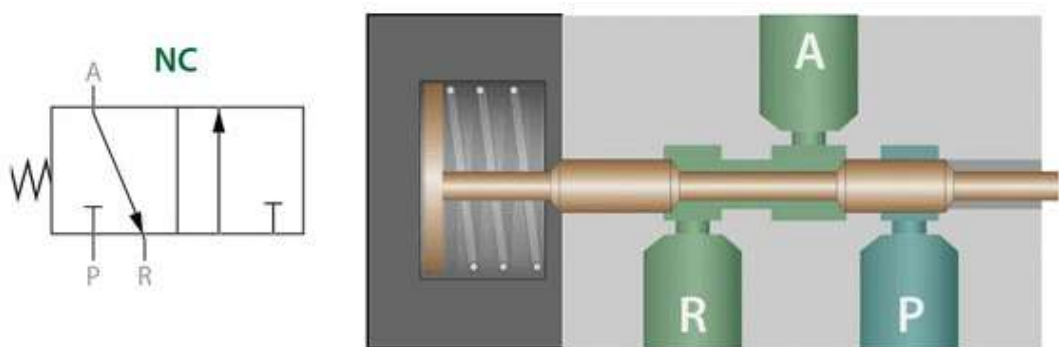


Рисунок 2.32 – Схема пневматичного розподільника 3/2 NC

## 2.7 Датчик тиску

Для роботи з пневматичними приводами екзоскелет повинен мати джерело стисненого газу, наприклад компресор з ресивером. Як нестача, так і надлишок тиску мають свої наслідки. При недостатньому рівні тиску в системі приводи не працюватимуть або працюватимуть некоректно, надлишковий тиск може призвести до пошкодження ресивера або шлангів, а також до травмування. Саме тому важливо відслідковувати та керувати рівнем тиску.

Для вимірювання тиску в необхідному діапазоні підходить датчик USP-G41-1.2. Робочий діапазон цього датчику - 0-1,2 МПа, напруга живлення – 5В, що додає складностей при роботі з 3.3 В логікою.

Керування компресором можна виконувати за допомогою транзистора або електромагнітного реле.



Рисунок 2.33 – Датчик USP-G41-1.2

## 3 ПРОГРАМНА ЧАСТИНА

### 3.1 Основна програма

Основна програма розроблена в середовищі STM32CubeIDE з використанням допоміжних програм STM32CubeMX та STMStudio мовами C та C++ для мікроконтролера STM32F103C8T6.

#### 3.1.1 Отримання та обробка сигналів з міодатчиків.

Перед початком роботи за необхідністю треба налаштувати підсилювач. Деякі з них мають налаштування рівня підсилення. Для демонстрації використано модуль AD620, який додатково має регулювання зміщення сигналу.



Рисунок 3.1 – Налагоджувальні резистори на модулі AD620

Сигнал з міодатчика по суті є звичайним аналоговим сигналом. Він зчитується за допомогою АЦП, який в свою чергу працює з DMA в циклічному режимі. Цей сигнал має вигляд, показаний на рисунку.

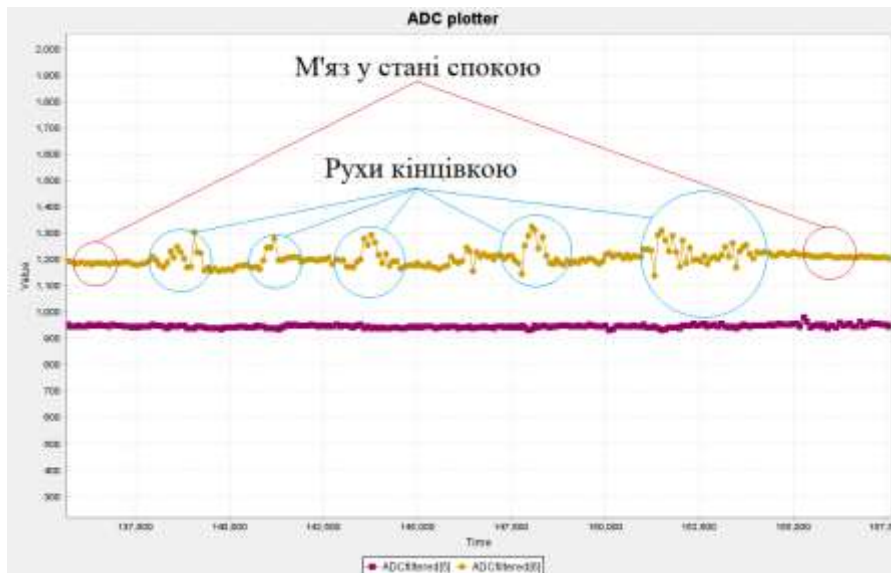


Рисунок 3.2 – Графік сигналу з міодатчика

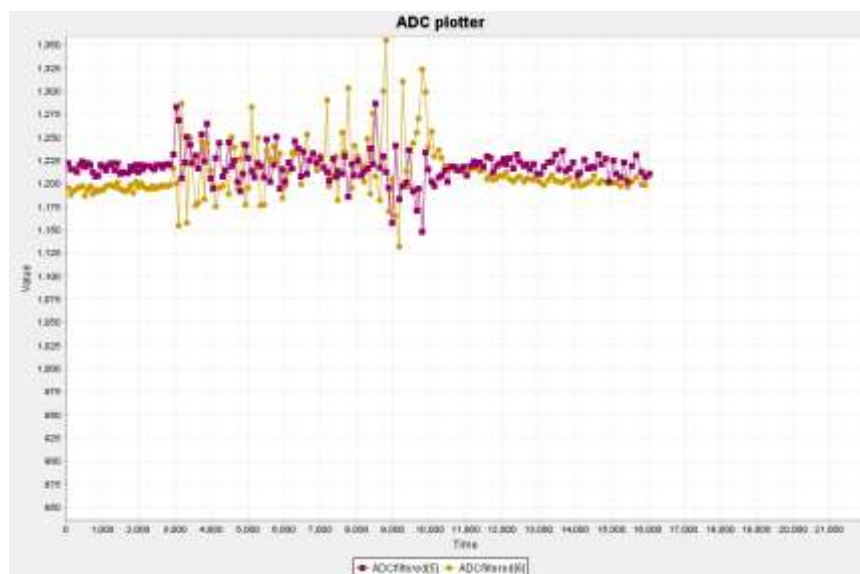


Рисунок 3.3 – Графік сигналу з міодатчика під час напруження двох м'язів(біцепс і трицепс)

Як видно з графіку, сигнал не є постійним і змінюється тільки в момент скорочення або сильного напруження м'язу. Під час утримання м'язу в певному положенні сигнал близький до сигналу в стані спокою. Тому треба придумати спосіб обробки сигналу для використання його в якості керуючого.

Для початку визначаємо амплітуду сигналу на деякому невеликому проміжку часу.

```
if (dt < amplitude_dt){
    for (uint8_t i = 1; i < 11; i++) {
        if(ADCfiltered[i] > miosignal_max[i - 1]){           //find min and max value
            miosignal_max[i - 1] = ADCfiltered[i];
        }
        else if(ADCfiltered[i] < miosignal_min[i - 1]){
            miosignal_min[i - 1] = ADCfiltered[i];
        }
    }
}
else{
    for (uint8_t i = 0; i < 10; i++) {
        amplitude[i + 1] = miosignal_max[i] - miosignal_min[i];           //get maximum amplitude
        miosignal_min[i] = miosignal_max[i];                               //and reset
        miosignal_max[i] = 0;
    }
}
```

Рисунок 3.4 – Визначення амплітуди

Програма безперервно записує найбільше та найменше значення сигналу. Після деякого проміжку часу визначається максимальна амплітуда. Один зі способів керування - порівнювати це значення з пороговим і при його досягненні вмикати керування на фіксовану величину. Метод примітивний і не дуже зручний в використанні, проте цілком дієвий, а в деяких ситуаціях навіть незамінний.

Цей метод можна покращити і зробити керування більш плавним і подібним до звичайних рухів тілом. Алгоритм аналізує значення амплітуди сигналу, отримане вище і визначає зростає амплітуда чи падає. Якщо амплітуда перестала рости, то її значення обробляється далі. В залежності від налаштувань керування може виконуватись одним м'язом, або ж парою м'язів(біцепс-трицепс, триголовий-чотириголовий м'язи ніг тощо). Керуюче значення змінюється на деякий крок вгору(розмір кроку налаштовується окремо) при досягненні деякого порогового значення(теж налаштовується). Зниження керуючого значення на цей самий крок виконується при досягненні іншого(меншого) порогового значення при роботі з одним м'язом або ж при досягненні порогового значення протидіючим м'язом, тобто значення плавно змінюється, наприклад, при русі кінцівкою в один та інший бік. Значення кроку регулювання можна зробити динамічним, таким чином можна

задавати не тільки положення, але і швидкість руху заданої кінцівки або іншої частини тіла. Для працездатності такого алгоритму екзоскелет повинен чинити деякий опір цільовим м'язам задля того, щоб м'яз при русі напружився з силою, достатньою для отримання чіткого та стабільного сигналу. Також система може працювати нестабільно коли цільовий м'яз виконує якусь додаткову роботу, окрім керуванням екзоскелетом. Тобто для ідеальної роботи цього алгоритму екзоскелет повинен брати на себе абсолютно все навантаження, максимально розвантаживши цільові м'язи.

```
n = Settings.mio[i];
m = Settings.mio2[i];

if(n > 0 && m > 0){ //if pair of muscles used
if(max_amplitude[n] > Settings.up_threshold[i]){
if(max_amplitude[n] > Settings.max_up_threshold[i]){
muscle_pos[i] += (Settings.muscle_step[i] * 3);
}
else if(max_amplitude[n] > (Settings.max_up_threshold[i] + Settings.up_threshold[i]) / 2){
muscle_pos[i] += (Settings.muscle_step[i] * 2);
}
else{
muscle_pos[i] += Settings.muscle_step[i];
}
if(muscle_pos[i] > Settings.muscle_lim[i]) muscle_pos[i] = Settings.muscle_lim[i];
}
else if(max_amplitude[m] > Settings.down_threshold[i] && max_amplitude[m] < Settings.power_treshold[i]){
if(max_amplitude[m] > Settings.max_down_threshold[i]){
muscle_pos[i] -= (Settings.muscle_step[i] * 3);
}
else if(max_amplitude[m] > (Settings.max_down_threshold[i] + Settings.down_threshold[i]) / 2){
muscle_pos[i] -= (Settings.muscle_step[i] * 2);
}
else{
muscle_pos[i] -= Settings.muscle_step[i];
}
if(muscle_pos[i] <= 0) muscle_pos[i] = 0;
}
max_amplitude[n] = 0;
max_amplitude[m] = 0;
}
```

Рисунок 3.5 – Алгоритм точного керування положенням та швидкістю за допомогою двох парних м'язів

```

else if(n > 0 && m == 0){ //if only 1
    if (max_amplitude[n] > Settings.up_threshold[i]){
        muscle_pos[i] += Settings.muscle_step[i];
        if(muscle_pos[i] > Settings.muscle_lim[i]) muscle_pos[i] = Settings.muscle_lim[i];
    }
    else if(max_amplitude[n] > Settings.down_threshold[i] && max_amplitude[n] < Settings.up_threshold[i]){
        muscle_pos[i] -= Settings.muscle_step[i];
        if(muscle_pos[i] <= 0) muscle_pos[i] = 0;
    }
    max_amplitude[n] = 0;
}
}
else if(Settings.signal_mode[i] == 0){
    muscle_pos[i] = 360;
}
}

```

Рисунок 3.6 – Алгоритм керування положенням за допомогою всього одного м'яза

Ще один цікавий і дуже корисний метод керування – використання макросів, тобто деяких комбінацій сигналів. Цей метод дозволяє не просто дублювати рухи тіла, а й розширити його можливості. По суті це кнопка, яку можна натиснути без рук, майже силою думки. До цієї кнопки можна прив'язати як фіксовані, запрограмовані заздалегідь рухи, так і будь-які інші функції, що не пов'язані з приводами екзоскелета. В даній системі керування за допомогою макросів можна змінити керуюче значення будь-якого з приводів або керувати сигналом одного з вільних цифрових пінів вводу-виводу.

```

void macros(){
    for (uint8_t i = 0; i < 10; i++){
        macros_dt[i] = (GetMicros() - macros_tar[i]) * 1e-6f;
        if(macros_dt[i] >= 1){
            uint8_t full_ticks = current_ticks[i];
            current_ticks[i] = 0;
            macros_tar[i] = GetMicros();

            if(full_ticks == Settings.macros_ticks[i]){ //macros 3 stuff
                if(macros_flag[i] == 1){
                    macros_flag[i] = 0;
                }
                else if(macros_flag[i] == 0){
                    macros_flag[i] = 1;
                }

                if(Settings.macros_action[i] > 0 && Settings.macros_action[i] < 15){ //set value
                    uint8_t cnct = Settings.macros_action[i];
                    if(Settings.macros_value[i] > 360){
                        macros_connector[cnct] = muscle_position[i]; //freeze position
                        macros_control[cnct] = macros_flag[i]; //on/off stage
                    }
                    else{
                        macros_connector[cnct] = Settings.macros_value[i]; //send value to the muscle drive
                        macros_control[cnct] = macros_flag[i]; //on/off stage
                    }
                }
                else if(Settings.macros_action[i] > 15 && Settings.macros_action[i] <= 15){ //on/off GPIO
                    GPIOWrites(Settings.macros_action[i]);
                }
            }
        }
    }
}

```

Рисунок 3.7 – Реалізація макросів

### 3.1.2 Робота з колекторними двигунами.

Колекторні двигуни по багатьом характеристикам програють безколекторним, проте вони простіші в керуванні та дешевші, що може стати основним фактором при виборі приводу під конкретну задачу. Тому цей тип приводу має право на існування.

В основі керування електродвигуном є ПІД контролер. ПІД(пропорційно-інтегрально-диференціальний) алгоритм керування це просте, проте достатньо точне рішення для регулювання будь-якого параметру.

Пропорційна частина визначає похибку регулювання, тобто різницю між заданим та поточним значенням параметру в даний момент часу. Проте сама по собі пропорційна частина не здатна точно регулювати параметр через наявність статичної похибки. При приближенні заданого значення до поточного похибка наближатиметься до нуля, проте регулювання перестане працювати при значенні, трохи більшому за нуль. Це означає що при використанні тільки пропорційної частини система ніколи не досягне заданого значення. Високі значення пропорційного коефіцієнту( $K_1$ ) зменшують вплив статичною помилки, проте не покривають її повністю і до того ж здатні перенасичити систему.

Інтегральна частина – це інтеграл похибки регулювання за часом. Ця частина призначена для компенсації статичною помилки. При досягненні значення пропорційної частини, яке недостатнє для регулювання, воно компенсується наростаючою інтегральною частиною.

Диференційна частина визначає темп зміни значення та прогнозує відхилення, що можуть виникнути у майбутньому, тобто прискорює або уповільнює регулювання.

$$u = K_1 \varepsilon + K_2 \int \varepsilon dt + K_3 \frac{d\varepsilon}{dt}$$

Рисунок 3.7 – Формула пропорційно-інтегрально-диференціального закону регулювання

Для більш точного та стабільного контролю положення необхідно керувати одразу кількома параметрами за допомогою пов'язаних між собою ПІД контролерів(залежних один від одного). Такий тип керування називається комплексним.

Необхідні параметри – безпосередньо положення, швидкість його зміни та струм. Останній параметр найменше за інші впливає на стабільність системи, проте необхідний для захисту від перевантажень.

Реалізація триконтурного комплексного контролера показана на рисунку.

```
float ComplexPID::complexLoop(uint8_t loops){ //call
    if(target_velo == abs(velo_lim)){ //overflow protection
        loop_angle.I = 0;
    }
    target_velo = _constrain(loop_angle(target_angle - getAngle()), -velo_lim, velo_lim);
    check_rotation();
    if(loops == 1){
        return target_velo;
    }
    if(target_current == abs(current_lim)){ //overflow protection
        loop_velocity.I = 0;
    }
    target_current = _constrain(loop_velocity(target_velo - velocity), -current_lim, current_lim);
    if(loops == 2){
        return target_current;
    }
    if(output == abs(pwm_lim)){ //overflow protection
        loop_torque.I = 0;
    }
    output = _constrain(loop_torque(target_current - getCurrent()), -pwm_lim, pwm_lim);
    if(loops == 3){
        return output;
    }
    return 0;
}
```

Рисунок 3.8 – Комплексний регулятор

```
float ComplexPID::getAngle(){
    static float dt = 0;
    float real_angle;

    dt = (GetMicros() - prev_Ts) * 1e-6f;

    if (dt < min_dt){
        return prevAngle;
    }

    angle = ( (float) (input_angle) / (float)cpr) * max_angle;
    // angle = max_angle / 4000.f * (float)input_angle;
    dangle = angle - prevAngle;
    max_d_angle = dangle;
    if(abs(dangle) > (input_dangle) ) full_rotations += ( dangle > 0 ) ? -1 : 1;

    real_angle = full_rotations * 360 + angle;
    velocity = (full_rotations) * 360 + (angle - prevAngle) / dt;
    prevAngle = angle;
    prev_Ts = GetMicros();
    return real_angle;
}
```

Рисунок 3.9 – Визначення кута та швидкості

Для керування двигуном необхідно два канали ШІМ. Отримане значення з комплексного регулятора треба правильно розподілити між ними.

Регулятор налаштовано таким чином, щоб вихідне значення могло мати як звичайне, так і від'ємне значення, що означає обертання в один чи інший бік. Тому якщо значення позитивне, то воно має бути передано на один канал, якщо негативне – то на другий. Занадто малі значення регулятора обмежуються через те, що вони недостатні для того, щоб вплинути на роботу двигуна, але при цьому використовують електричну енергію. Енергія використовується тільки на нагрів обмоток двигуна. І хоч нагрів незначний через те, що регулятор обмежує струм, проте такі витрати енергії впливають на автономність системи.

```
for(uint8_t i = 0; i < 5; i++){ //DC motors' outs to PWM
    if(Settings.motor_out[i] && Settings.motor_out[i + 6]){
        if(motorPWM[i] > DC_min_PID[i]){
            PWMarray[Settings.motor_out[i]] = abs(motorPWM[i]);
            PWMarray[Settings.motor_out[i + 6]] = 0;
        }
        else if(motorPWM[i] < -DC_min_PID[i]){
            PWMarray[Settings.motor_out[i]] = 0;
            PWMarray[Settings.motor_out[i + 6]] = abs(motorPWM[i]);
        }
        else if (motorPWM[i] == 0){
            PWMarray[Settings.motor_out[i]] = 0;
            PWMarray[Settings.motor_out[i + 6]] = 0;
        }
    }
}
```

Рисунок 3.10 – Передача значень з регулятора до каналів ШІМ

### 3.1.3 Робота з безколекторними двигунами.

Безколекторні двигуни хоч і працюють за тими самими фізичними принципами, проте мають інший принцип передачі електричної енергії та керування.

Значення положення та швидкості регулюються так само за допомогою декількох ПІД контролерів, проте регулювання струму та розподілення його між трьома фазами виконується абсолютно іншим способом.

В основі керування безколекторного двигуна лежить алгоритм полеорієнтованого керування(FOC, field-oriented control), який розподілює струм між трьома фазами і, окрім того, підтримує певний кут між магнітним полем і ротором для забезпечення максимально можливого моменту сили.

Для керування безколекторними двигунами існує і простіший метод, проте він має зовсім іншу ефективність та інше призначення. Метод шести кроків показує себе краще при роботі без зворотного зв'язку(openloop control) і широко використовуються, наприклад, у БПЛА, де важливий кожний грам та міліметр вільного простору, а отже може не бути місця для датчика положення будь-якого типу, або ж в електротранспорті, де простота та зручність налаштування важливіша за максимально можливу ефективність. Алгоритм FOC при роботі без зворотного зв'язку поступається за ефективністю і до того ж сам по собі складніший в налаштуванні.

Проте в даній системі мова йде про сервоприводи – двигуни з точним регулюванням деяких параметрів, для якого необхідно мати значення цих параметрів, а отже і зворотній зв'язок, і бажано якомога точніший. При наявності зворотного зв'язку, до того ж точного – за допомогою абсолютного енкодера, алгоритм FOC показує максимальну ефективність, тому саме він і обраний для роботи.

В останньому контурі комплексного регулятора для безколекторного двигуна використовується не простий ПІД контролер, а алгоритм синусоїдальної модуляції, який має у своїй основі два рівняння перетворення – зворотне перетворення Парка та зворотне перетворення Кларка.

```
// Inverse park transform
Ualpha = _ca * Ud - _sa * Uq; // -sin(angle) * Uq;
Ubeta =  _sa * Ud + _ca * Uq; //  cos(angle) * Uq;

// Clarke transform
Ua = Ualpha;
Ub = -0.5f * Ualpha + _SQRT3_2 * Ubeta;
Uc = -0.5f * Ualpha - _SQRT3_2 * Ubeta;
```

Рисунок 3.11 – Перетворення для полеорієнтованого керування

Ця функція приймає вихідне значення з ПІД контролера для струму та значення для його компенсації(опціонально), на виході видає 3 значення ШІМ, що оброблені синусоїдальною модуляцією. Фактично на виході три синусоподібних сигнали.

Далі ці сигнали передаються до трьох або шести ШІМ каналів. Використання шести ШІМ каналів(роздільне керування верхніми та нижніми плечами драйвера) дає більший контроль над двигуном, а саме можливість перемикає одну з фаз(кожну по черзі) у третій, повністю відключений стан. Це дозволяє зменшити втрати потужності, а отже збільшити ефективність двигуна, проте цей спосіб потребує вдвічі більшу кількість ШІМ каналів, а також контроль deadtime засобами мікроконтролера. Обраний мікроконтролер має 16 ШІМ каналів, а також всього один advanced таймер(таймер з розширеним функціоналом, який здатний в тому числі апаратно керувати двома плечами з використанням тільки одного каналу, без використання процесорного часу). Тому в цій системі доцільніше використовувати керування трьома ШІМ сигналами.

Принцип передачі сигналів до каналів ШІМ схожий на принцип для колекторного двигуна. Різниця в тому, що вихідний сигнал не потрібно ділити на дві частини, а також потрібно передавати на три канали замість двох.

```
for(uint8_t a = 0; a < 3; a++){ //BLDC motors' outs to pwm
    if((Settings.BLDC_out[a] && (Settings.BLDC_out[a + 3]) && (Settings.BLDC_out[a + 6]))){
        PWMarray[Settings.BLDC_out[a] + 1] = BLDC_PWM_a[a];
        PWMarray[Settings.BLDC_out[a + 3] + 1] = BLDC_PWM_b[a];
        PWMarray[Settings.BLDC_out[a + 6] + 1] = BLDC_PWM_c[a];
    }
}
```

Рисунок 3.12 – Передача значень з регулятора до каналів ШІМ для BLDC двигуна

### 3.1.4 Робота з пневматичними приводами

Пневматичні приводи(циліндри або м'язи) мають зовсім інший принцип керування. На відмінну від двигунів вони керуються за допомогою розподільників,

що накладає деякі обмеження пов'язані з плавністю руху. Проте вони простіші в керуванні, можуть видавати більше потужності, принаймні в момент старту, а за формою і рухами подібні до людських м'язів.

Найпростіший спосіб керування, що показаний на початку розділу з міодатчиками, примітивний, проте підходить для роботи з пневматичними приводами. При досягненні деякої амплітуди сигналу пневматичний м'яз скорочується і тим самим збільшує силу. В цьому режимі роботи зворотній зв'язок не використовується, тобто м'яз скорочується до максимально можливого ступеня.

Максимальна робоча частота у таких розподільників всього 3-5 Гц, дуже рідко доходить до 10 Гц. Цю інформацію треба враховувати задля стабільної роботи системи та попередження виходу з ладу розподільника. В даному режимі роботи частота обмежується програмно(задається затримка між перемиканнями).

```
if(enable_pulse[i]){ //0.35 seconds delay
if(mio_input <= Settings.mioMIN[i]){
    MIOpwm[Settings.out[i]] = 0;
}

else if(mio_input >= Settings.mioMAX[i]){
    MIOpwm[Settings.out[i]] = arr2;
}
enable_pulse[i] = 0;
}
```

Рисунок 3.13 – Реалізація керування пневматичними приводами

Для забезпечення плавності роботи також можна використовувати ПІД контролер. І хоч, як сказано вище, максимальна частота перемикачів обмежена, проте ПІД контролер здатний працювати і з набагато повільнішими системами. Максимальна робоча частота розподільника 3 Гц, тому мінімальний період - 0,33. Отже для плавного регулювання період ШІМ сигналу має бути не меншим за 1 секунду. Для роботи ПІД контролера необхідний зворотній зв'язок, наприклад, за допомогою абсолютних енкодерів, які підтримуються системою керування. Нажаль забезпечити контроль за всіма можливими параметрами не вийде через

низьку та неоднорідну швидкість та неможливість виміряти момент сили(у двигунах він пропорційний струму)

При використанні всього одного м'язу, як у попередньому методі керування, привід здатен працювати тільки в один бік, переміщення в інший відбувається силою м'язів або за допомогою іншого приводу, який не пов'язаний з попереднім. Для роботи приводу в обидва боки можна використати два приводи за допомогою одного розподільника типу 5/2.

Для роботи з одним розподільником типу 5/2 необхідно змістити сигнал(нижнє крайнє положення – рух в один бік, верхнє – в інший). Також сигнал треба обмежити в крайніх положеннях згідно з мінімальним періодом для пневматичного розподільника.

```
else if(Settings.mio_out[i] > 0){  
  
    MIOpwm[i] += 550; //value must be 0-1100  
  
    if(psc2 > 60000){ //if 1Hz  
        if(MIOpwm[i] < 370){ //min period  
            MIOpwm[i] = 0;  
        }  
        else if(MIOpwm[i] < 730){  
            MIOpwm[i] = 1100;  
        }  
    }  
    else if(psc2 > 30000){ //if 0.5Hz  
        if(MIOpwm[i] < 200){ //min period  
            MIOpwm[i] = 0;  
        }  
        else if(MIOpwm[i] < 900){  
            MIOpwm[i] = 1100;  
        }  
    }  
}  
  
PWMArray[Settings.mio_out[i]] = MIOpwm[i];  
}
```

Рисунок 3.14 – Передача сигналу до одного розподільника

Робота з двома розподільниками 3/2 NC або 5/2 схожа на роботу з колекторним двигуном, відрізняються лише обмеження сигналу.

```

if(Settings.mio_out[i] > 0 && Settings.mio_out[i + 10] > 0){
    MIOpwm[i] *= 2; //value must be -1100-1100

    if(psc2 > 60000){ //if 1Hz
        if(MIOpwm[i] < 370){ //min period
            MIOpwm[i] = 0;
        }
        else if(MIOpwm[i] < 730){
            MIOpwm[i] = 1100;
        }
    }
    else if(psc2 > 30000){ //if 0.5Hz
        if(MIOpwm[i] < 200){ //min period
            MIOpwm[i] = 0;
        }
        else if(MIOpwm[i] < 300){
            MIOpwm[i] = 1100;
        }
    }
} //limit min and max value

if(MIOpwm[i] > 0){
    PwMarray[Settings.motor_out[i]] = abs(MIOpwm[i]);
    PwMarray[Settings.motor_out[i + 10]] = 0;
}
else if (motorPwm[i] < 0){
    PwMarray[Settings.motor_out[i]] = 0;
    PwMarray[Settings.motor_out[i + 10]] = abs(MIOpwm[i]);
}
else if (motorPwm[i] == 0){
    PwMarray[Settings.motor_out[i]] = 0;
    PwMarray[Settings.motor_out[i + 10]] = 0;
}
}

```

Рисунок 3.15 – Передача сигналу до двох розподільників

### 3.1.5 Отримання даних від конфігуратора та налаштування.

Для налаштування параметрів системи керування використовується додаткова програма – конфігуратор. Опис цієї програми показано у окремому розділі.

Ця програма зв'язується з програмою системи керування за допомогою протоколу UART. Програма системи керування отримує дані в перериванні по одному байту, записує їх в буфер, розділяє його(виконує парсинг) та записує в список налаштувань.

```

/* USER CODE BEGIN 4 */
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
    serial_data[serial_counter] = serial_buffer;
    if(serial_buffer == 254){ //if stop www
        max_data = serial_counter;
        stop_float = 254;
        parser();
        serial_counter = 0;
    }
    else if(serial_counter >= sizeof(serial_data)){
        serial_counter = 0;
    }
    else{
        serial_counter++;
    }

    HAL_UART_Receive_IT(&huart3, &serial_buffer, 1);
}

```

Рисунок 3.16 – Запис значень з UART у буфер та виклик парсера у перериванні

```

setcode = serial_data[0];
for (uint8_t i = 1; i < max_data; i++){
    switch (setcode){
        case 0:
            break;
        case 1:
            if(adc_match(serial_data[i]) == 0){
                Settings.mio[i - 1] = serial_data[i];
            }
            break;
        case 2:
            if(pwm_match(serial_data[i]) == 0){
                Settings.out[i - 1] = serial_data[i];
            }
            break;
        case 3:
            if(pwm_match(serial_data[i]) == 0){
                Settings.motor_out[i - 1] = serial_data[i];
            }
            break;
    }
}

```

Рисунок 3.17 – Фрагмент функції парсингу буфера

Використання цих налаштувань було показане в усіх попередніх прикладах коду програми. Налаштування дозволяють вибрати один з трьох типів приводів(колекторний двигун, безколекторний, пневматика), призначити виходи керування на будь-які з доступних ШІМ каналів, призначити для керування будь-який з доступних аналогових входів, налаштування коефіцієнтів ПІД контролерів, налаштування порогів спрацювання датчиків тощо. Налаштування записуються в енергонезалежну пам'ять за командою з конфігуратора

```

typedef struct {
uint8_t mio[10]; //muscle sensor signal 1
uint8_t out[10]; //output type(pneumatic, BLDC, DC motor) 2
uint8_t motor_out[15]; //DC motor PWM outs 3
uint8_t BLDC_out[15]; //BLDC motor PWM outs 4
float motor_PID[60]; //BLDC motor PID coefficients 5
float BLDC_PID[30]; //DC motor PID coefficients 6
uint8_t current[13]; //current sensors inputs 7
uint8_t angle[20]; //angle sensors inputs 8
uint8_t other[10]; //pressure sensor adc0 min1 max2, voltage sensor adc3 div4,5 min6-7
uint8_t type_m[10]; //muscle control type 10
uint8_t mioMIN[10]; //muscle min value 11
uint8_t mioMAX[10]; //muscle max value 12
float mioPID[40]; //muscle PID coefficients 13
uint8_t mio2[10]; //muscle sensor signal 2 14
uint8_t i2cadr[10]; //change i2c address (for AS5048 only) 15
uint8_t i2ctype[10]; //i2c sensor AS5600 or AS5048 or MT6701 16
uint8_t spitype[10]; //SPI sensor AS5147 or MA730 17
uint8_t spipin[10]; //SPI NSS pin 18
uint8_t max_angle[10]; //max angle of screw
uint16_t muscle_step[10];
uint16_t muscle_lim[10];
uint16_t up_threshold[10];
uint16_t max_up_threshold[10];
uint16_t down_threshold[10];
uint16_t max_down_threshold[10];
uint16_t macros_threshold[30];
uint8_t macros_ticks[30];
uint8_t macros_action[30]; //0 - no macros, 1-10 - fixed position, 11-15 - GPIO
uint16_t macros_value[30];
uint8_t mio_out[20];
} settings_S;

```

Рисунок 3.18 – Структура з параметрами налаштувань

### 3.1.6 Додаткові функції.

Також система має додаткові функції для керування частинами, що не пов'язані з сигналами та приводами. Це функції моніторингу напруги та тиску.

Функція контролю напруги вимірює напругу живлення(до неї необхідно під'єднати будь-який аналоговий вхід на вибір) та сигналізує світлодіодом при досягненні деякого порогу. Звичайно хороша акумуляторна збірка повинна мати BMS(систему керування батареєю), проте краще щоб екзоскелет не вимикався у непідходящий момент. До того ж BMS(частина, яка відповідає за мінімальну напругу) вимикає акумулятор при достатньо низькій напрузі, бо її призначення – останній ступінь захисту, якщо не працюють або відсутні всі інші.

```
void checkBattery(){
    if (Settings.other[3] > 0){
        float batteryVoltage;
        batteryVoltage = ADCfiltered[Settings.other[3]] / 4096 * 3.3 * (float)(Settings.other[4] + Settings.other[5]/100);

        if ((uint8_t)batteryVoltage < Settings.other[6]){
            HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 1);
        }
        else HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 0);
    }
}
```

Рисунок 3.19 – Функція моніторингу напруги та сигналізації

При використанні пневматичних приводів необхідно слідкувати за рівнем тиску в системі. Функція контролю тиску призначена для керування компресором.

```
void pressure_control(){
    static float pressure;

    pressure = ((ADCfiltered[Settings.other[0]] / 4096) * 2 - 0.1) / 0.75 / 10; // voltage to pressure, bar

    if(pressure < Settings.other[1]){
        setPin(Settings.other[9]);
    }
    else if(pressure > Settings.other[2]){
        resetPin(Settings.other[9]);
    }
}
```

Рисунок 3.20 – Функція керування компресором

Компресор керується за допомогою транзистора або електромеханічного реле так як частота перемикачів відносно невисока.

### 3.2 Конфігуратор

Для налаштування системи використовується додаткова програма конфігуратор. Вона необхідна для вибору приводів та датчиків, а також для деяких параметрів, які потребують індивідуальних налаштувань.

Програма написана мовою Python з використанням бібліотеки PySide та додаткової програми Qt Designer.

Розглянемо основні її елементи.

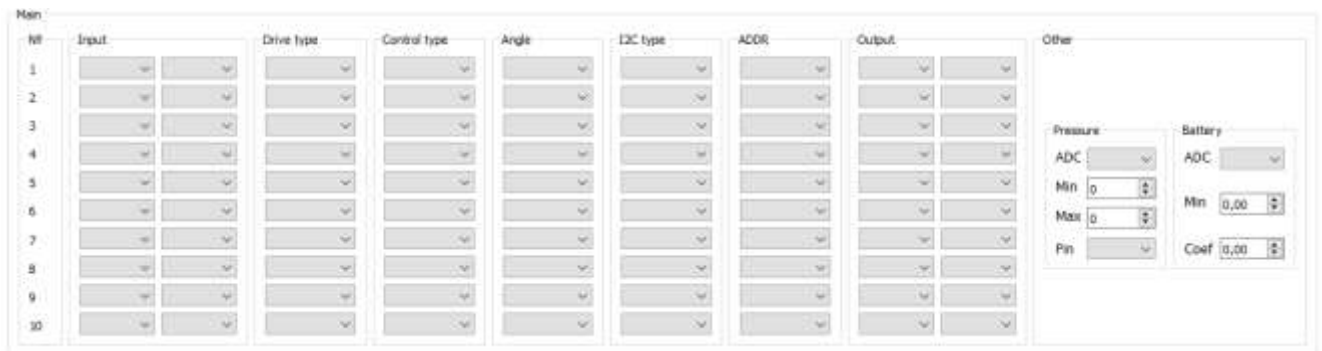


Рисунок 3.21 – Основні налаштування

У блоці основних налаштувань можна налаштувати роботу до 10 приводів (обмежено кількістю аналогових входів), а саме – цільовий м'яз(один або два для роботи в парі), тип приводу – колекторний двигун, безколекторний або пневматика, тип керування – між двома положеннями або плавний, налаштування датчику положення(для пневматичних приводів, опціонально), вибір виходу – до 10 пневматичних приводів, до 5 колекторних або до 3 безколекторних двигунів, налаштування функцій моніторингу тиску та напруги.

Value										
Nº	Down threshold		Up threshold		Step	Min	Max	P	I	D
1	0	0	0	0	0	0	0	0,00	0,00	0,00
2	0	0	0	0	0	0	0	0,00	0,00	0,00
3	0	0	0	0	0	0	0	0,00	0,00	0,00
4	0	0	0	0	0	0	0	0,00	0,00	0,00
5	0	0	0	0	0	0	0	0,00	0,00	0,00
6	0	0	0	0	0	0	0	0,00	0,00	0,00
7	0	0	0	0	0	0	0	0,00	0,00	0,00
8	0	0	0	0	0	0	0	0,00	0,00	0,00
9	0	0	0	0	0	0	0	0,00	0,00	0,00
10	0	0	0	0	0	0	0	0,00	0,00	0,00

Рисунок 3.22 – Налаштування значень основних параметрів

У блоці налаштування значень можна змінити порогові значення(вони можуть відрізнятися для різних м'язів, різних електродів та підсилювачів і навіть різних положень електродів), мінімальний крок зміни значення для роботи в плавному режимі, мінімальне та максимальне цільові значення для пневматичних приводів(тільки для роботи в режимі двох положень та при наявності зворотного зв'язку), ПІД коефіцієнти для роботи з пневматичними приводами.

Macros															
Nº	Macro 1				Macro 2				Macro 3						
	Threshold	Ticks	Action	Value	Threshold	Ticks	Action	Value	Threshold	Ticks	Action	Value			
1	0	0	▼	0	0	0	▼	0	0	0	▼	0			
2	0	0	▼	0	0	0	▼	0	0	0	▼	0			
3	0	0	▼	0	0	0	▼	0	0	0	▼	0			
4	0	0	▼	0	0	0	▼	0	0	0	▼	0			
5	0	0	▼	0	0	0	▼	0	0	0	▼	0			
6	0	0	▼	0	0	0	▼	0	0	0	▼	0			
7	0	0	▼	0	0	0	▼	0	0	0	▼	0			
8	0	0	▼	0	0	0	▼	0	0	0	▼	0			
9	0	0	▼	0	0	0	▼	0	0	0	▼	0			
10	0	0	▼	0	0	0	▼	0	0	0	▼	0			

Рисунок 3.23 – Налаштування макросів

Наступний блок відповідає за макроси та їх налаштування. Для кожного м'язу можна налаштувати до трьох макросів. Налаштування макросів – порогове значення(воно може відрізнятись від значення для звичайного керування, також звичайне керування може бути відсутнім), кількість пульсацій(рухів) м'язом, вибір дії – перехоплення керування приводом(поточним або будь-яким іншим, макрос має пріоритет над іншими способами керування) або керування зовнішніми пристроями за допомогою вільних цифрових виходів(вмикати або вимикати цифровий вихід командою макросу), значення для вибраного приводу(в градусах; якщо вказане значення більше за 360, то команда фіксує привід у поточному положенні). Повторне використання макросу вимикає команду.

Блок налаштування сервоприводів призначений для налаштування виводів, датчиків(струму та положення, датчики положення можуть бути підключені за допомогою різних інтерфейсів – аналоговий, I2C, SPI, датчики струму тільки аналогові), ПІД коефіцієнтів та додаткових параметрів. Безколекторні двигуни мають іншу будову, а отже потребують додаткові налаштування – кількість пар полюсів, коефіцієнт редукції(електричні двигуни зазвичай використовують в парі з редуктором, у безколекторних двигунів датчик положення ставиться до редуктора, отже значення кута датчика і валу редуктора відрізняються, проте вони пропорційні і пов'язані коефіцієнтом передачі редуктора), робоча напруга(для роботи без датчику струму).



Рисунок 3.24 – Налаштування електродвигунів(сервоприводів)

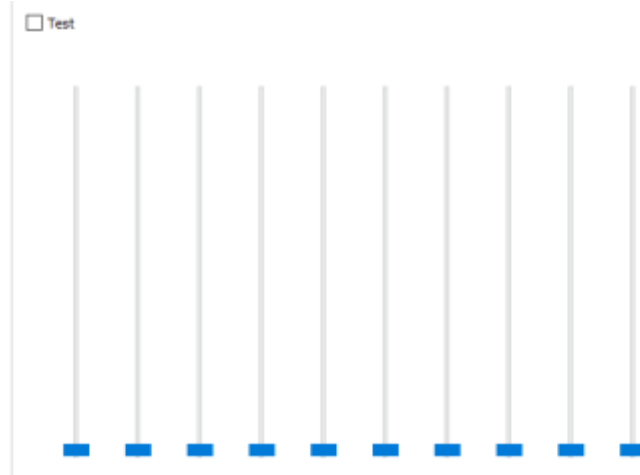


Рисунок 3.25 – Керування приводами

Блок керування приводами дозволяє давати команди приводам з графічного інтерфейсу. Це потрібно для перевірки справності та правильності налаштування приводу.

## ВИСНОВКИ

Під час вивчення теоретичного матеріалу було проаналізовано декілька моделей сучасних екзоскелетів, будову, типи приводів, датчиків та принципів керування екзоскелетом. Проведено порівняння різних датчиків та приводів, їх переваг, недоліків, особливостей використання, особливостей підключення та налаштування.

Було розроблено плату драйвера для двигунів, що оптимальна для використання в цьому проекті. Вона забезпечує керування двигунами з високою потужністю, високим робочим струмом та апаратне керування затримкою перемикачів між плечами напівмоста (deadtime).

З використанням мов програмування C та C++ було розроблено програму для керування екзоскелетом, яка включає в себе опитування датчиків м'язової активності, перетворення їх сигналів в сигнали керування з точними налаштуваннями процесу та приведення в дію екзоскелета за допомогою колекторних і безколекторних двигунів, а також пневматичних приводів (пневматичних циліндрів або штучних м'язів на вибір). Також програм має додаткові функції для обслуговування систем екзоскелета.

З використанням мови програмування Python було розроблено програму конфігуратор з графічним інтерфейсом, яка дозволяє налаштовувати екзоскелет для роботи з різними типами приводів, відкалібрувати датчики для правильної їхньої роботи, налаштувати самі приводи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. "Research and development prototype for machine Augmentation of human strength and endurance. Hardiman I project " General electric company, 1971
2. Miomir K. Vukobratovic, "When were active exoskeletons actually born? " Mihajlo Pupin Institute, 2007
3. Nigg B. M., Herzog W., "Biomechanics of the Musculo-Skeletal system." Wiley, 1999
4. Fouad Sabry, "Servomotor: Precision Control in Robotics and Automated Systems" One Billion Knowledgeable, 2025
5. Fouad Sabry, "Pneumatics: Driving Precision and Power in Robotics Science" One Billion Knowledgeable, 2025
6. Mohsen Shahinpoor, "Artificial Muscles: Applications of Advanced Polymeric Nanocomposites", CRC Press, 2021
7. A. Skuric, HS. Bank, R. Unger, O. Williams, D. González-Reyes , "SimpleFOC: A Field Oriented Control (FOC) Library for Controlling Brushless Direct Current (BLDC) and Stepper Motors", Journal of Open Source Software, 2022
8. Datasheet AD8232
9. Datasheet INA333
10. Datasheet AD620
11. Datasheet IRS2184
12. Datasheet AS5600
13. Datasheet AS5047
14. Datasheet AS5048
15. Datasheet MT6701
16. Datasheet MA730
17. Datasheet INA240
18. Datasheet ACS712
19. Datasheet ACS711
20. Datasheet ACS758