

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроніки

Кафедра електронних пристроїв та систем

До захисту допущено:

Завідувач кафедри

_____ Євген ВЕРБИЦЬКИЙ

«___» червня 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Електронні компоненти і системи»
спеціальності 171 «Електроніка»**

на тему: «Система керування корпусом людиноподібного робота»

Виконав (-ла):

студент (-ка) IV курсу, групи ДС-91

Андрій КОРИЖ

Керівник:

Проф. каф. ЕПС, д.т.н., проф.

Володимир РОМАШКО

Консультант з Технічний розділ:

Старший викладач

Ганна САРИБОГА

Рецензент:

Доцент кафедри АМЕС, к.т.н, доцент

Наталія ФІЛІПОВА

Консультант

з нормоконтролю

доц. каф. ЕПС, к.т.н., доц.,

Лариса БАТРАК

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2023 року

[illegible]

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет електроніки
Кафедра електронних пристроїв та систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 171 «Електроніка»

Освітньо-професійна програма «Електронні компоненти і системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Євген ВЕРБИЦЬКИЙ

«17» травня 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту
Коржу Андрію Андрійовичу

1. Тема проєкту «Система керування корпусом людиноподібного робота»

керівник проєкту Ромашко Володимир Якович проф. каф. ЕПС, д.т.н., проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання,

затверджені наказом по університету від «30» травня 2023 р. №2063-с

2. Термін подання студентом проєкту 17.06.2023 _____

3. Вихідні дані до проєкту Нагріваючі панелі 10 Вт, автономність 9 годин, система керування роботом на мікроконтролері Raspberry Pi Pico H Cortex-M0133 MHz

4. Зміст пояснювальної записки 1. Аналіз існуючих рішень 2. Розробка та проєктування системи керування роботом. 3. Розробка алгоритму та програмного забезпечення системи керування 4. Апробація діючої моделі системи керування людиноподібним роботом _____

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Презентація, Схема електрична структурна, Схема електрична принципова, _____

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Технічний	Ганна САРИБОГА		

7. Дата видачі завдання 17 травня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
	Отримання завдання	17.04.2023	Виконано
	Аналіз існуючих рішень	24.04.2023	Виконано
	Вибір компонентів	01.05.2023	Виконано
	Вибір програмного середовища	10.05.2023	Виконано
	Проектування системи	24.05.2023	Виконано
	Створення програмної частини	31.05.2023	Виконано
	Створення прототипу	07.06.2023	Виконано
	Оформлення пояснювальної записки	17.07.2023	Виконано

Студент

Андрій КОРЖ

Керівник

Володимир РОМАШКО

АНОТАЦІЯ

У дипломному проєкті наведено огляд літератури на тему аналізу засобів керування об'єктами та розглянуто різні види систем керування роботами. Було розроблено схему електричну принципову, схему електричну структурну та на основі вищезазначених даних проведено аналіз та вибір комплектуючих. Було створено алгоритм роботи людиноподібного робота та описано його виконання. На основі отриманих даних було створено програмне забезпечення для системи керування людиноподібним роботом.

В результаті було створено та випробовано систему керування корпусом людиноподібного робота з наступними характеристиками:

- автономність на рівні 9 годин;
- варіативність сценаріїв роботи;
- робота у важких погодних умовах;
- стабільність роботи.

Ключові слова: система керування, автономна система керування, людиноподібний робот, мікроконтролери.

ANNOTATION

In the diploma project, a literature review was conducted on the topic of object control analysis and various types of robot control systems were considered. An electrical schematic diagram and a structural electrical diagram were developed, and based on the obtained data, an analysis and selection of components were carried out. An algorithm for the operation of a humanoid robot was created and its execution was described. Based on the collected data, software was developed for the control system of the humanoid robot.

As a result, a control system for the body of a humanoid robot was created and tested, with the following characteristics:

- Autonomy of up to 9 hours.
- Variability of work scenarios.
- Operation in harsh weather conditions.
- Stability of operation.

Keywords: control system, autonomous control system, humanoid robot, microcontrollers.

ЗМІСТ

ВСТУП.....	2
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	3
1.1 Системи керування.....	3
1.2. Аналіз засобів керування об'єктом	13
2. РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ РОБОТОМ	19
2.1 Розробка структурної схеми системи керування	19
2.2 Розробка принципової електричної схеми.....	21
2.3 Огляд та підбір комплектуючих	31
3. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ	43
3.1 Опис алгоритму роботи системи керування людиноподібним роботом	43
3.2 Розробка програмного забезпечення	48
4. АПРОБАЦІЯ ДІЮЧОЇ МОДЕЛІ СИСТЕМИ КЕРУВАННЯ ЛЮДИНОПОДІБНИМ РОБОТОМ	53
4.1 Розрахунок енергоефективності системи.....	53
4.2 Підбір акумуляторів для забезпечення автономності системи	55
ВИСНОВКИ	58
ДОДАТОК А.....	61
ДОДАТОК Б	64
ДОДАТОК В	65
ДОДАТОК Г	66
ДОДАТОК Д.....	72

ВСТУП

Сьогоднішній день характеризується популярністю та потужністю автоматичного та інтелектуального управління, що широко використовується у нашому повсякденному житті. У даній роботі надано огляд класичних технологій управління, що використовують реальні мікроконтролерні системи. Завжди відчувалася пропасті між теоріями управління та їх практичними реалізаціями, і цей розрив часто не був задоволеною багатьма наявними книгами з контролю, включаючи підручники.

У роботі надається інформація про методи найпопулярніших та практичних систем контролю, а також описуються методи контролю з практичними застосуваннями. Це дозволяє зрозуміти принципи та підходи до управління системами з реальними цілями.

Цей огляд технологій управління є важливим для розуміння сучасних досягнень та практичного використання в області автоматизації та контролю. Враховуючи швидкий технологічний розвиток, оновлення та застосування нових методів управління стають необхідними для досягнення ефективності та прогресу в різних галузях, включаючи промисловість, транспорт, медицину та багато інших.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						к
Змн.	Арк.	№ докум.	Підпи	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Системи керування

Система - це група пов'язаних речей, які працюють разом як одне ціле[1]. Ці речі можуть бути реальними чи уявними. Системи можуть бути створеними людиною, як-от автомобільний двигун, або природними, як-от зоряна система. Системи також можуть бути концепціями, створеними людьми для організації ідей. А система контролю керує, командує, направляє або регулює поведінку інших пристроїв або систем за допомогою контурів керування [2]. Він може варіюватися від одного контролера домашнього опалення за допомогою термостата, який керує домашнім котлом, до великих промислових систем керування, які використовуються для керування процесами чи машинами. Дві популярні системи управління відкритий і замкнений цикл системи керування[1]. Система керування повинна містити контролер для виконання різноманітних функцій керування системою для отримання бажаних результатів. Таким чином, повна система управління повинна складатися з входів, контролера, системи або процесу та виходів системи. Більшість систем керування в нашому світі є замкнутими системами керування, але деякі з них належать до розімкнених систем керування.

Система з відкритим контуром

У системі керування з розімкненим контуром керуюча дія від контролера не залежить від змінної процесу. Прикладом цього є котел центрального опалення, керований тільки таймером. Керуюча дія – увімкнення або вимкнення котла. Змінною процесу є температура будівлі. Цей контролер управляє системою опалення протягом постійного часу незалежно від температури в будівлі. Іншою системою керування з відкритим контуром є мікрохвильова піч, і порядок її роботи можна проілюструвати на рис. 1.1

					ДП. ДС-91. 07. 000 ПЗ	Ар. №
Змн.	Арк.	№ докум.	Підпи	Дата		

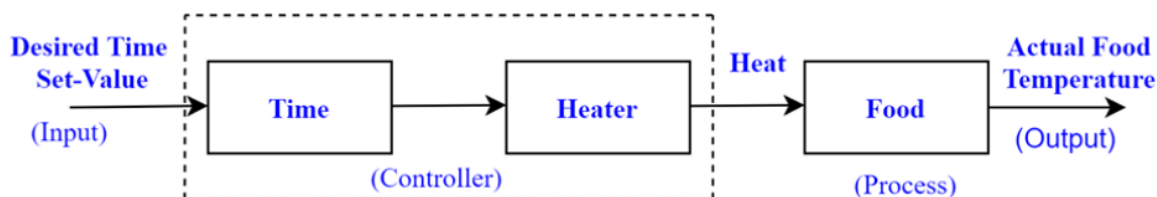


Рис.1.1. Система з відкритим контуром

Встановивши бажаний проміжок часу (встановлене значення), який можна вважати входом, таймер і нагрівач (контролер) можуть запускати та відкладати потрібний період часу, щоб нагрівач міг нагріти їжу (процес) до відповідна температура (вихід). Причина, по якій ми називаємо цей тип системи керування системою з відкритим контуром, полягає в тому, що вихід не впливає або не впливає на керуючу дію вхідного сигналу. Іншими словами, у цій системі керування з відкритим контуром вихідний сигнал не вимірюється та не отримує зворотного зв'язку для порівняння з входом. Таким чином, очікується, що система з відкритим контуром точно слідуватиме своїй вхідній команді або заданому значенню незалежно від кінцевого результату. Цю систему керування ще називають системою управління без зворотного зв'язку [1]. Крім того, система з відкритим контуром не має інформації про вихідні умови, такі як температура їжі, тому вона не може самостійно виправити будь-які помилки, які вона може зробити, коли задане значення дрейфує, навіть якщо це призводить до великих відхилень від заданого значення. Іншим недоліком системи з відкритим контуром є те, що вони не можуть впоратися з перешкодами або змінами в умовах, які можуть зменшити її здатність виконувати бажане завдання.

Система замкнутого циклу

Система керування із замкнутим контуром, також відома як система контролю зі зворотним зв'язком це система керування, яка використовує

					ДП. ДС-91. 07. 000 ПЗ	Ар
						А
Змн.	Арк.	№ докум.	Підпи	Дата		

концепцію системи з відкритим контуром, оскільки її прямий шлях має одну або більше петель зворотного зв'язку або шляхів між виходом і входом [1]. Посилання на зворотний зв'язок просто означає, що деяка частина вихідного сигналу повертається назад на вхід для формування частини збудження системи.

Системи із замкнутим контуром розроблені для автоматичного досягнення та підтримки бажаного результату шляхом порівняння його з фактичним станом входу. Він робить це шляхом порівняння частини всього зворотного зв'язку від виходу та бажаного вхідного сигналу для генерації сигналу помилки, який є різницею між виходом і опорним вхідним сигналом

$$\text{Помилка} = \text{Вхід} - \text{Вихід}.$$

Іншими словами, замкнута система є повністю автоматичною системою керування, в якій її керуюча дія певним чином залежить від виходу. Наша попередня мікрохвильова піч як приклад, і вона може стати замкнутою системою керування, додавши канал зворотного зв'язку з датчиком, який використовується для визначення фактичної температури їжі, і компаратором, який використовується для порівняння вихідного сигналу зворотного зв'язку з вхід (встановлене значення). Сигнал помилки може бути отриманий шляхом порівняння вихідного сигналу зворотного зв'язку та входу, і цей сигнал помилки може бути використаний як вхідний сигнал для контролера для автоматичного налаштування вихідного сигналу до бажаного заданого значення вхідного сигналу або зробити вихідний сигнал максимально рівним вхід, наскільки це можливо. Цей датчик відстежуватиме фактичну температуру їжі та порівнюватиме її з вхідним еталонним значенням або відніме від нього. Сигнал помилки потім посилюється контролером, і вихідний сигнал контролера вносить необхідну корекцію в систему опалення, щоб зменшити будьяку помилку. Наприклад, якщо їжа недостатньо гаряча, контролер може збільшити температуру або час нагрівання. Так само, якщо їжа дуже гаряча, контролер може знизити температуру або зупинити процес, щоб не перегріти або

					ДП. ДС-91. 07. 000 ПЗ	Ар.
						№
Змн.	Арк.	№ докум.	Підпи	Дата		

підгоріти їжу. Конфігурація замкнутого циклу визначається сигналом зворотного зв'язку, отриманим від датчика або термометра в нашій системі нагрівання їжі. Величина та полярність сигналу помилки будуть безпосередньо пов'язані з різницею між необхідною температурою нагрівання та фактичною температурою їжі. Термін керування замкнутим контуром завжди означає використання керування зворотним зв'язком для зменшення будь-яких помилок між виходом і входом. Саме завдяки цьому зворотному зв'язку він розрізняє основні відмінності між системою з відкритим і замкнутим контуром. Таким чином, точність вихідного сигналу залежить від шляху зворотного зв'язку, який загалом можна зробити дуже точним у електронних системах керування та схемах. З цього простого прикладу можна побачити, що система із замкнутим контуром має багато переваг перед системами з відкритим контуром. Основною перевагою системи керування із зворотним зв'язком із замкнутим контуром є її здатність зменшувати чутливість системи до зовнішніх збурень і може автоматично виправляти або змінювати помилку, щоб зробити вихідний сигнал максимально рівним входу. Ще один момент, який слід відзначити для цієї системи керування зворотним зв'язком, полягає в тому, що шлях зворотного зв'язку або датчик повинні мати можливість забезпечувати дві такі функції:

1. Направте частину всього виходу назад на вхід, щоб дозволити компаратору порівняти вхід і вихід, щоб отримати різницю або сигнал помилки як оновлений вхід для контролера.

2. Перетворіть тип виведення на тип введення, щоб зробити це порівняння можливим.

Як показано в нашій замкнутій системі керування розігріванням їжі на рис. 1.2, датчик не лише забезпечує зворотний зв'язок від виходу до входу, але також потребує перетворення типу виходу (температура їжі) у тип входу (значення таймера). Регулюючи налаштування значення таймера, температуру страви можна змінити та змінити на бажану температуру. Інакше порівняння

					ДП. ДС-91. 07. 000 ПЗ	Ар. №
Змн.	Арк.	№ докум.	Підпи	Дата		

між введенням і виходом не може бути виконано через різні типи введення та виводу.

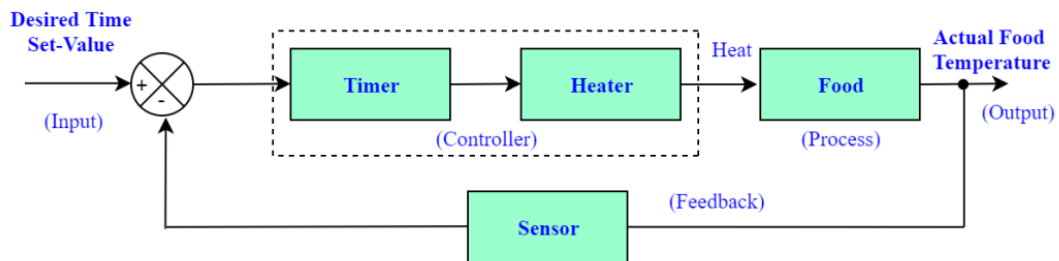


Рис.1.2 Система замкнутого циклу

Порівняно з розімкнутою системою керування, замкнуті системи керування мають багато переваг перед системами з розімкненим контуром. Однією з переваг є те, що використання зворотного зв'язку робить реакцію системи відносно нечутливою до зовнішніх збурень і внутрішніх коливань параметрів системи, таких як температура або коливання елементів. Таким чином, можна використовувати відносно неточні та дешевші компоненти для отримання точного керування певним процесом або установкою.

Проектування системи

Система управління складається з підсистем і процеси (або заводи), зібрані з метою отримання бажаного результату з бажаною продуктивністю, враховуючи визначений вихід. На рисунку 1.3 показана система керування в її найпростішій формі, де вхід представляє бажаний вихід.

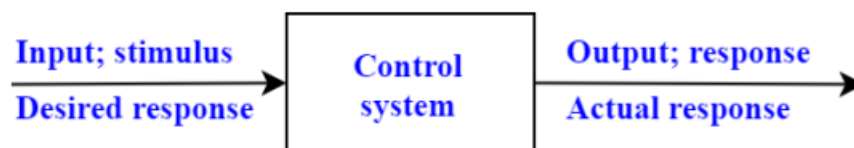


Рис.1.3 Система контролю

Наприклад, розглянемо ліфт. Натискання кнопки четвертого поверху є вхідним сигналом, який представляє наш бажаний результат, показаний як ступінчаста функція на рис. 1.4. Ефективність ліфта можна побачити з кривої відгуку ліфта на рисунку.

Очевидними є два основних показники ефективності:

1. перехідна характеристика;
2. помилка стаціонарного стану.

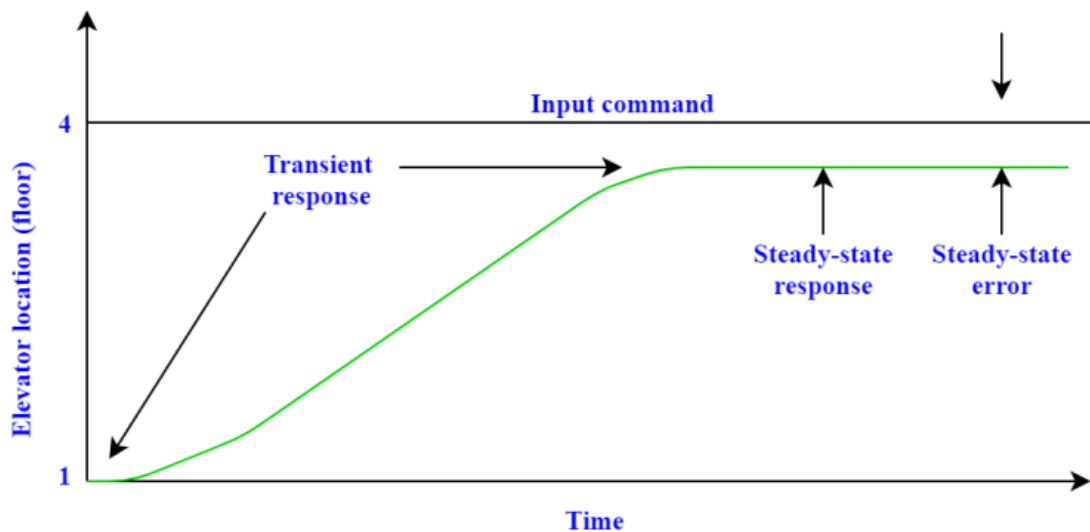


Рис.1.4 Перехідна характеристика

У нашому прикладі комфорт і терпіння пасажирів залежать від перехідної реакції. Якщо ця реакція надто швидка, комфорт пасажирів приносить жертву; якщо занадто повільно, терпіння пасажира приносить жертву. Похибка стаціонарного стану є ще однією важливою характеристикою специфікації, оскільки безпека та зручність пасажирів були б пожертвовані, якби ліфт не вирівнявся належним чином.

Аналіз - це процес, за допомогою якого визначається продуктивність системи. Наприклад, ми оцінюємо його перехідну характеристику та похибку сталого стану, щоб визначити, чи відповідають вони бажаним специфікаціям.

Дизайн - це процес, за допомогою якого продуктивність системи створюється або змінюється. Наприклад, якщо проаналізовано перехідний відгук системи та помилку в стаціонарному стані та виявлено, що вони не

відповідають специфікаціям, ми змінюємо параметри або додаємо додаткові компоненти, щоб відповідати специфікаціям.

Перехідна реакція важлива. У випадку ліфта повільна перехідна реакція робить пасажирів нетерплячими, тоді як надмірно швидка реакція викликає у них дискомфорт. Якщо ліфт коливається навколо місця прибуття довше секунди, це може призвести до збентеження. Перехідний відгук також важливий із структурних причин: надто швидкий перехідний відгук може спричинити постійне фізичне пошкодження. Інша мета аналізу та проектування зосереджена на відповіді в стаціонарному стані. Як ми бачили, ця відповідь нагадує вхід і зазвичай є тим, що залишається після того, як перехідні процеси спадають до нуля. Ми стурбовані точністю відповіді в стаціонарному стані. Ми кількісно визначаємо стаціонарні похибки, аналізуємо стаціонарні похибки системи, а потім розробляємо коригувальну дію, щоб зменшити стаціонарні похибки, наша друга ціль аналізу та проектування.

Обговорення перехідної характеристики та помилки стаціонарного стану є спірним, якщо система не має стабільності. Системи управління повинні бути розроблені таким чином стабільний. Тобто їх природна реакція повинна зменшуватися до нуля, коли час наближається до цілі, або коливатися. У багатьох системах перехідна характеристика, яку ви бачите на графіку часової характеристики, може бути безпосередньо пов'язана з природною реакцією. Таким чином, якщо природна реакція спадає до нуля, коли час наближається до нуля, перехідна реакція також згасає, залишаючи лише вимушену реакцію. Якщо система стабільна, можна розробити правильну перехідну реакцію та характеристики помилки в стаціонарному стані. Стабільність — наша третя мета аналізу та проектування.

Функція передачі системи керування

Передатна функція представляє співвідношення між вихідним сигналом системи керування та вхідним сигналом для всіх можливих вхідних значень.

					ДП. ДС-91. 07. 000 ПЗ	Ар №
Змн.	Арк.	№ докум.	Підпи	Дата		

Блок-схема — це візуалізація системи керування, яка використовує блоки для представлення функції передачі та стрілки, які представляють різні вхідні та вихідні сигнали.

Необов'язково, щоб вихід і вхід системи керування були однієї категорії. Наприклад, в електродвигунах вхідним сигналом є електричний сигнал, тоді як вихідним сигналом є механічний сигнал, оскільки для обертання двигунів потрібна електрична енергія. Подібним чином в електричному генераторі вхідним сигналом є механічний сигнал, а вихідним сигналом є електричний, оскільки для виробництва електроенергії в генераторі потрібна механічна енергія. Але для математичного аналізу системи всі види сигналів повинні бути представлені в однаковій формі. Це робиться шляхом перетворення всіх видів сигналів у форму Лапласа. Крім того, передаточна функція системи представлена формою Лапласа шляхом ділення вихідної передавальної функції Лапласа на вхідну передатну функцію Лапласа. Отже, базова блок-схема системи керування може бути представлена у вигляді схеми, що зображена на рис 1.5.



Рис.1.5 Передатна функція

Існує два основні способи отримання передатної функції для системи керування:

1. Метод блок-схеми: Незручно виводити повну передатну функцію для складної системи керування. Тому передаточна функція кожного елемента управління системи представлена блок-схемою. Методи редукції блок-схеми застосовуються для отримання потрібної функції передачі.

2. Графіки потоку сигналу: Модифікована форма блок-схеми є графом потоку сигналу. Блок-схема дає графічне зображення системи керування. Граф потоку сигналу ще більше скорочує представлення системи керування.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						10
Змн.	Арк.	№ докум.	Підпи	Дата		

Отже, для нашої системи за вимогами поставлених завдань було вибрано систему керування з відкритим контуром.

Аналіз засобів керування об'єктом

Автономія є ключовим аспектом майбутніх систем, а планування архітектури є одним з найважливіших аспектів у розробці та впровадженні таких систем. По-перше, узагальнюються поняття автономності та автономних систем у різних сферах.

У 21 столітті з розвитком інформаційних технологій було складно задовольнити вимоги до систем за допомогою лише однієї технології. Тому для автоматизації системи були розроблені інформаційно-комунікаційні технології (ІКТ), включаючи сенсорні, комп'ютерні та інтелектуальні технології, комунікаційні технології та технології управління. Однак зі зростанням складності функцій системи і розмаїттям системних рішень, технологія автоматизації стала недостатньою для вирішення виробничих потреб.

Запит на розвиток автономних систем став все більш важливим для зменшення залежності від людської взаємодії та покращення здатності індивідів і систем самостійно виконувати завдання. Процес розробки системи автоматизації представлений на рис. 1.6.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						11
Змн.	Арк.	№ докум.	Підпи	Дата		

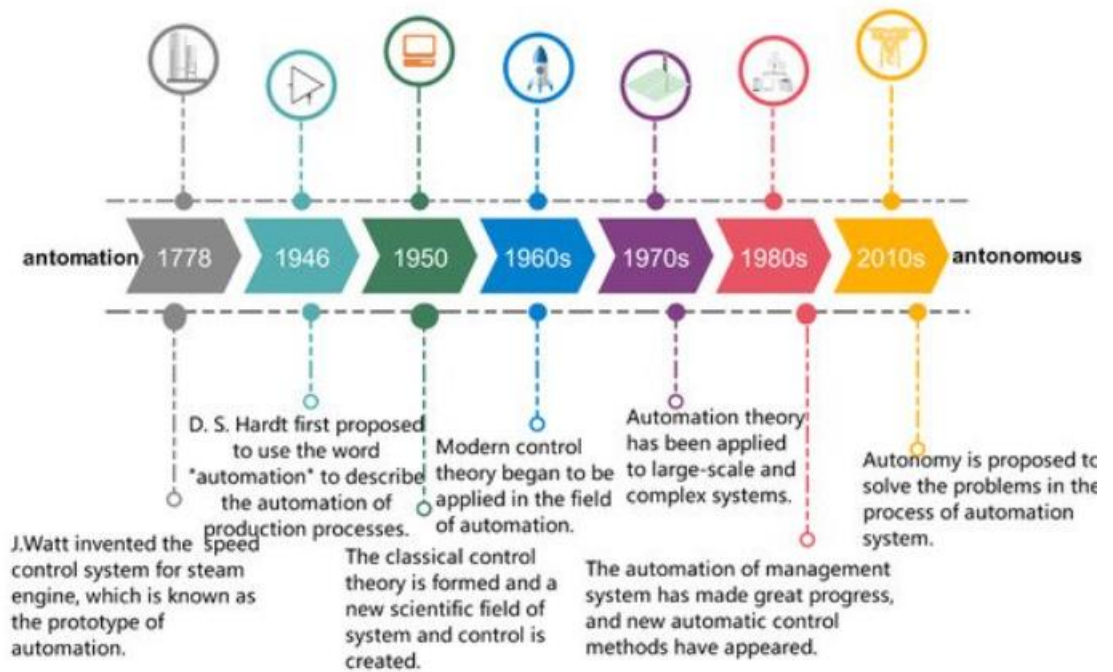


Рис.1.6 Розвиток індивідуумів і систем від автоматизації до автономної.

Автономія та автономна система

Автоматизація відноситься до автоматичного управління та функціонування пристроїв, процесів або систем за допомогою механічних або електронних пристроїв, замінюючи людську працю. Автономність, з свого боку, відрізняється тим, що передає повноваження прийняття рішень від критичних вузлів системи до довірених об'єктів [18]. Довірений об'єкт може діяти і приймати рішення на підставі своїх повноважень без зовнішнього втручання.

Одна з найважливіших відмінностей між автоматизованою та автономною системами полягає в їх здатності до самонавчання та самоevolюції. Коли система працює за заданими правилами без помилок, її можна назвати автоматизованою системою, а не автономною системою. Автономна система, для досягнення своєї мети, повинна бути здатна точно сприймати та оцінювати оточуюче середовище, стан системи, розуміти поставлену задачу, аналізувати ситуацію, самостійно формулювати та вибирати відповідні дії для досягнення поставлених цілей.

Автономія може розглядатись як розширення автоматизації з вищим рівнем інтелектуальних та функціональних можливостей. В різних областях автономія має різні визначення, які можна знайти в табл. 1.1.

Таблиця 1.1

Визначення автономії в різних сферах.

Філософія	Автономія є основною характеристикою самої себе, на яку впливає навколишнє середовище, але обмежується внутрішніми факторами.
Соціологія	Характеристика полягає в тому, що своєрідна система від внутрішнього спонтанного до зовнішнього просування, своєю дією не потрібно сприяти, спиратися на зовнішні сили, і може розібратися з ситуацією за власним бажанням.
Теорія управління	Автономія - це надання об'єкту повноважень щодо прийняття рішень, так що об'єкт має право діяти в певному обсязі, тобто автономія - це рішення, прийняте самостійно без стороннього втручання.
Робото-техніка	Автономність - це здатність роботизованої системи сприймати, спостерігати, аналізувати, планувати, приймати рішення та автоматично вживати заходів.

1.2. Аналіз засобів керування об'єктом

Архітектура системи автоматизації та автономної системи

Архітектура системи включає різні значущі моделі, які допомагають у проектуванні та моделюванні системи. Одна з таких моделей - модель системної логіки - визначає внутрішній зв'язок між фізичними елементами системи. Архітектура системи є основою для впровадження автономних систем.

У 1964 році Г. М. Амдал запропонував концепцію архітектури, після чого ця концепція здобула значний інтерес і стала основою для багатьох досліджень. За останні півстоліття архітектурна дисципліна досягла великих успіхів. Архітектура автономних систем включає різні структури, такі як фізична структура, логічна структура, програмно-апаратна структура, організаційна структура, функціональна структура, мережева структура і т. д.

Всеосяжна логічна концептуальна структура включає структуру класифікації послуг та логічну структуру, а технічна структура включає програмну та апаратну структури. Апаратна структура визначає розгортання програмного забезпечення системи. Архітектурний дизайн використовується для опису складу системи, її функцій, логіки та інших аспектів дизайну.

Фізична структура описує загальний дизайн системи, включаючи програмне та апаратне забезпечення, а програмна система використовується для розгортання системного програмного забезпечення. При проектуванні програмної системи необхідно враховувати як апаратну, так і програмну частини.

Фізична структура також включає елементи фізичної побудови та мережеву форму системи для забезпечення надійності, надійності та різноманітності завдань, пов'язаних із системою. Організаційна структура зосереджується на організаційних відносинах між компонентами системи на різних рівнях.

Логічна архітектура визначає внутрішні кореляції між об'єктами або елементами в системі. Вона фокусується на логічному зв'язку всередині системи, а логічні архітектури можуть мати "ієрархічну" структуру. Це означає, що система може мати фізичний логічний рівень, рівень бізнес-функцій і мережевий рівень даних.

Процес розробки архітектури системи включає різні етапи, які представлені на рис 1.7.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						14
Змн.	Арк.	№ докум.	Підпи	Дата		

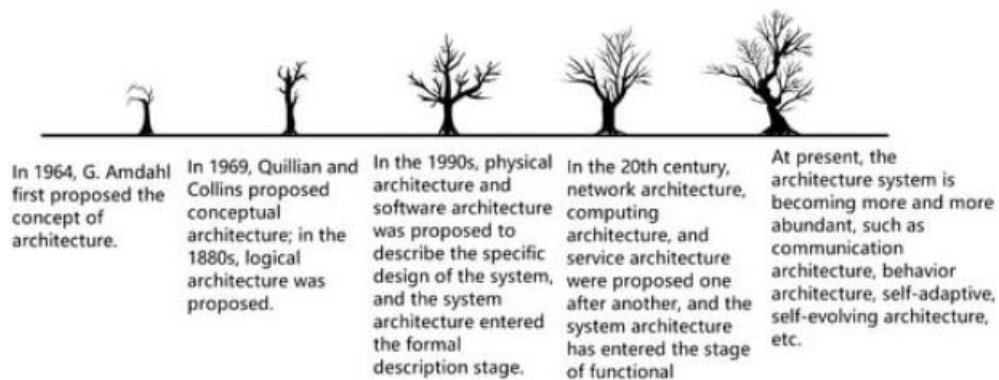


Рис.1.7 Процес розробки архітектури системи.

Таким чином, можна виявити, що основна мета побудови архітектури автономної системи є суттєвою, і особливо важливою для проектування загальної структури та реалізації очікуваних функцій системи, що, у свою чергу, підвищує автономні можливості та можливості виконання завдань системи. Для автономної системи чотири види здібностей, включаючи сприйняття навколишнього середовища та самосприйняття, планування, самовиконання та здатність до самонавчання, що обіцяють самоадаптацію та самоеволюцію.

Репрезентативна архітектура системи автоматизації

Робот-система - це стандартна і проста система автоматизації. Як репрезентативна форма системи автоматизації, архітектура робота, як правило, складається з підсистеми сприйняття, підсистеми планування та підсистеми виконання. Відповідно до різних комбінаційних форм трьох підсистем, їх можна розділити на архітектуру, засновану на знаннях (також відому як горизонтальна, децентралізована архітектура), архітектура на основі поведінки (також відома як вертикальний тип декомпозиції), а також гібридна архітектура, що поєднує знання та поведінку.

В даний час архітектура, заснована на знаннях, прийнята як видатна архітектура системи Робот, такого роду системна архітектура, натхнення

людським пізнанням знань. Пізнання людини зазвичай містить наступні процеси: сприйняття навколишньої інформації; моделювання моделі тіла для його кінематичного рівняння і динамічного рівняння; планування поставлених завдань; виконання інструктажу; і повторення з модуля сприйняття. Реалізація модуля виявляється і контролюється. Крім того, модель навколишнього середовища повинна бути точною. Архітектура, заснована на знаннях, пов'язує модулі відповідно до людського пізнання.

Ілюстрація архітектури, заснованої на знаннях, представлена на рис.1.8. В архітектурі кожен модуль може працювати і розвиватися самостійно, а реалізація системи проста.

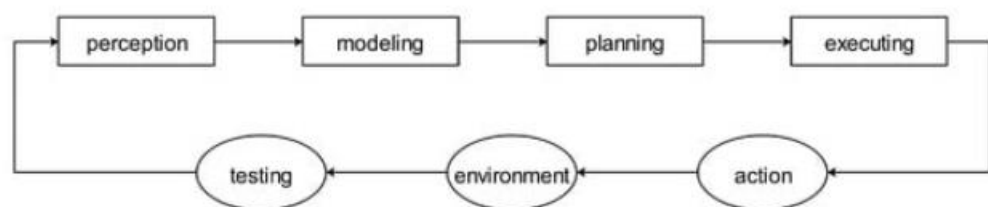


Рис.1.8 Архітектура заснована на знаннях

Побудова архітектури, заснованої на знаннях, є типовим методом будівництва зверху вниз. Метод побудови зверху вниз відноситься до генерації системних дій, які проходять ряд процесів, від сприйняття до виконання.

Модуль сприйняття також відіграє значну роль в системі. Модуль сприйняття може самостійно конструювати інформацію про глобальне середовище і зробити висновок про взаємозв'язок кожного елементного об'єкта в інформації про глобальне середовище. Однак відчуті дані не вплинуть безпосередньо на виконання дії.

Модуль планування є найважливішим компонентом в цій архітектурі, який визначає рух робота. Модуль планування впорається із завданням. Через задані цілі та системні обмеження планувальник дасть наступні інструкції щодо

дій відповідно до сприйнятої інформації та виконає все завдання у співпраці кожного модуля.

У цій структурі кожен модуль має важливе значення. При співпраці кожного модуля система виконує все завдання. Модуль сприйняття може самостійно будувати глобальну інформацію про навколишнє середовище та робити висновок про взаємозв'язок кожного елемента в глобальній екологічній інформації. Якщо інформація про глобальне середовище відсутня, модуль планування не може завершити та правильно спланувати інструкції щодо дій. Встановлення екологічної інформації залежить від апаратних умов системи. Через слабку обчислювальну потужність буде неминуча затримка контуру управління системою, що призводить до відсутності реального часу і гнучкості системи.

Через конкатенатну структуру архітектури, заснованої на знаннях, є деякі очевидні недоліки. Перший недолік - відсутність надійності системи. У конкатенатній структурі, якщо один з модулів вийде з ладу, вся система зруйнується. Крім того, виконання дій у реальному часі є недосконалим, оскільки передача інформації буде проходити через ряд процесів і не може реагувати на швидко мінливе середовище.

У 1986 році професор Родні А. Брукс з Австралії запропонував архітектуру, засновану на поведінці, також відому як інклюзивна архітектура [4]. У порівнянні з архітектурою, заснованою на знаннях, архітектура, заснована на поведінці, використовує паралельний спосіб побудови системи.

Модуль сприйняття і модуль виконання включені в структуру функцій кожного рівня архітектури, заснованої на поведінці. Відповідно до різниці кожного шару, модуль сприйняття та модуль планування активуються на основі попиту в режимі реального часу. В архітектурі, заснованій на поведінці, система планує різні рівні поведінкових можливостей відповідно до різних завдань. Можливості поведінки накладаються на кожному рівні функціональної

					ДП. ДС-91. 07. 000 ПЗ	Ар
						12
Змн.	Арк.	№ докум.	Підпи	Дата		

структури. Загалом, складна поведінка впливатиме на просту поведінку, а проста поведінка також впливатиме на складну, але вплив обмежений. Будь то низькорівневий або високий рівень, кожен рівень має незалежну поведінку і може самостійно керувати інтелектуальним роботом для генерації відповідного руху.

Архітектура, заснована на поведінці, виділяє структуру управління поведінкою від сприйняття до дії. Архітектура, заснована на поведінці, є типовою паралельною архітектурою. Кожна поведінка включає ряд можливостей від сприйняття, моделювання, планування, виконання тощо. Типова архітектура на основі поведінки продемонстрована на рис.1.9.

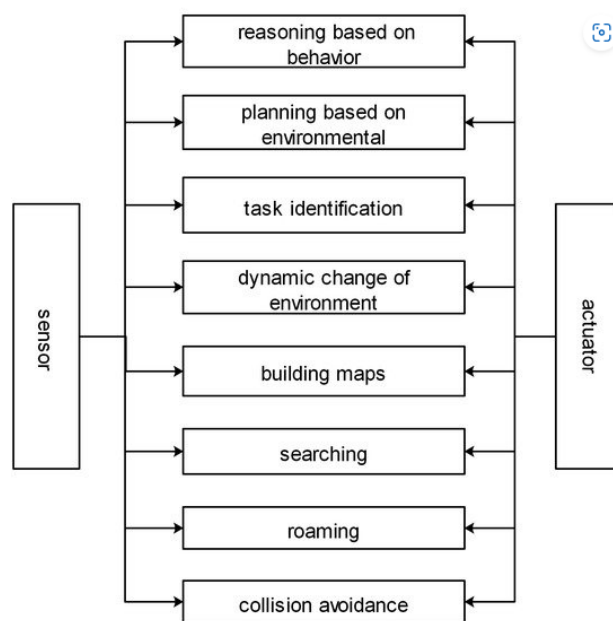


Рис.1.9 Архітектура побудована на поведінці

2. РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ РОБОТОМ

2.1 Розробка структурної схеми системи керування

Для реалізації автономної системи керування роботом було розроблено структурну схему згідно із технічним завданням до складу якої входить блок керування, блок живлення, система повороту голови манекену, тепловипромінюючі елементи та блок моніторингу параметрів системи (рис.2.1).

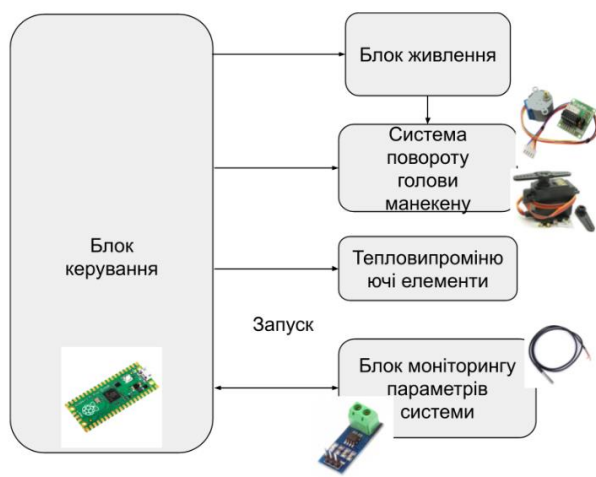


Рис. 2.1 Структурна схема АСК

Пропорційна інтегральна похідна (ПІД) регулятор часто називають «тричленним» контролером [1]. На даний момент це один із найбільш часто використовуваних контролерів у переробній промисловості. У ПІД-регуляторі керуюча змінна генерується з члена, пропорційного похибці, члена, який є інтегралом помилки, і члена, який є похідною помилки.

- Пропорційний: помилка множиться на приріст $K_{стор}$. Дуже високий коефіцієнт посилення може спричинити нестабільність, і дуже низький коефіцієнт підсилення може призвести до відхилення системи.

- Інтеграл: береться інтеграл похибки та множиться на приріст K_i . Виграш можна регулювати для зведення помилки до нуля за необхідний час. Занадто високий коефіцієнт посилення може викликати коливання, а занадто низький коефіцієнт посилення може призвести до повільної реакції.

- Похідна: похідна похибки множиться на приріст K_d . Прибуток, якщо підсилення занадто високе, система може коливатися, а якщо підсилення занадто низьке, відгук може бути повільним.

Структурна схема класичного безперервного ПІД-регулятора (рис.2.2)

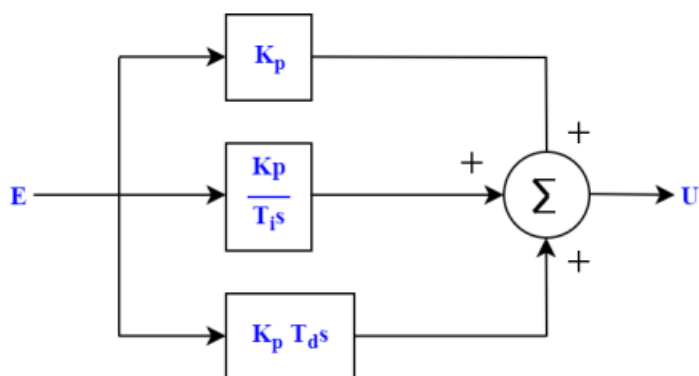


Рис.2.2 ПІД - регулювання

Налаштування ПІД

Коли в системі використовується ПІД-регулятор, важливо налаштувати контролер, щоб забезпечити необхідну реакцію. Налаштування ПІД-регулятора передбачає вибір значень для параметрів регулятора $K_{ст}, T_i, T_d$. Існує багато методів налаштування контролера, починаючи від перших методів, описаних JG Ziegler і NB Nichols (відомих як алгоритм налаштування Ziegler Nichols) у 1942 та 1943 роках, до останніх контролерів з автоматичним налаштуванням. Зіглер і Ніколс запропонували значення для параметрів ПІД установки на основі випробувань установки з відкритим або замкнутим контуром. Відповідно до Циглера та Ніколса, передавальна функція системи з відкритим

контуром може бути апроксимована із затримкою часу та системою з одним порядком.

2.2 Розробка принципової електричної схеми

Для реалізації автономної системи керування роботом було розроблено структурну схему згідно із технічним завданням до складу якої входить блок керування, блок живлення, система повороту голови манекену, тепловипромінючі елементи та блок моніторингу параметрів системи.

Для реалізації автономної системи керування роботом було розроблено схему електричну принципову з мікроконтролером Raspberry Pi Pico (рис.2.3). Був проведений огляд потрібних електронних компонентів.

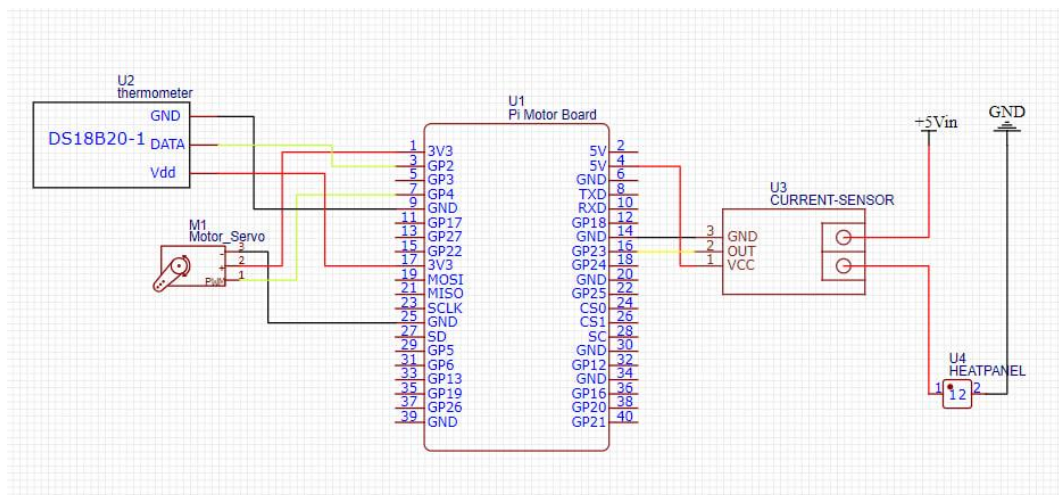


Рис.2.3 Схема електрична принципова

Датчики

Датчик - електричний/механічний/хімічний пристрій, який відображає екологічний атрибут у кількісному вимірюванні [7]. Кожен датчик заснований на принципі трансдукції - перетворення енергії з однієї форми в іншу.

Датчики можна класифікувати як аналогові або цифрові. Аналогові датчики є більш доступними, а їхні виходи є аналоговими напругами. Наприклад, на виході аналогового датчика температури може бути напруга,

пропорційна вимірянній температурі. Аналогові датчики можна підключити до комп'ютера лише за допомогою аналого-цифрового перетворювача. Цифрові датчики не дуже поширені, і вони мають виходи логічного рівня, які можна безпосередньо підключити до вхідного порту комп'ютера. Вибір датчика для конкретного застосування залежить від багатьох факторів, таких як вартість, надійність, необхідна точність, роздільна здатність, діапазон і лінійність датчика.

Датчик температури

Температура є однією з фундаментальних фізичних змінних у більшості застосувань хімічних речовин і керування процесами. Точне й надійне вимірювання температури є важливим майже в усіх програмах керування процесом. Датчики температури можуть бути аналоговими і цифровими. Одними з найбільш часто використовуваних аналогових датчиків температури є термопари, датчики температури опору (RTD) і термістори. Цифрові датчики мають форму інтегральних схем. Вибір датчика залежить від точності, температурного діапазону, швидкості реакції, теплового зв'язку, середовища (хімічного, електричного чи фізичного) і вартості.

Популярним аналоговим вихідним датчиком температури з інтегральною схемою є LM35, виготовлений компанією National Semiconductors Inc. (рис. 2.4). Це 3-контактний датчик аналогового виходу, який забезпечує лінійну вихідну напругу 10 мВ/°С. Діапазон температур 0°С до +100°, з точністю до 1,5°С

					ДП. ДС-91. 07. 000 ПЗ	Ар
						22
Змн.	Арк.	№ докум.	Підпи	Дата		

[illegible]

Під час вимірювання температури може бути кілька джерел похибок. Деякі важливі можливі помилки описані нижче.

- Датчик самонагрівання. Для роботи резистивних датчиків температури, терморезисторів і датчиків інтегрованих схем потрібне зовнішнє джерело живлення. Джерело живлення може призвести до нагрівання датчика, що призведе до помилки у вимірюванні. Ефект самонагрівання залежить від розміру датчика та кількості потужності, що розсіюється датчиком. Самонагрівання можна уникнути, використовуючи мінімально можливу зовнішню потужність або враховуючи ефект нагрівання під час вимірювання.

-Електричний шум. Електричний шум може внести помилки у вимірювання. Термопари виробляють дуже низьку напругу, і, як наслідок, шум може легко потрапити під час вимірювання. Зазвичай цей шум можна звести до мінімуму, використовуючи фільтри низьких частот і тримаючи провід датчика якомога коротшими та подалі від двигунів та інших електричних машин.

- Механічна напруга. Деякі датчики, такі як RTD, чутливі до механічних впливів, тому їх слід використовувати обережно. Механічні навантаження можна мінімізувати, уникаючи деформації датчика.

- Термомуфта. Важливо, щоб для точних і швидких вимірювань датчик мав хороший контакт з вимірювальною поверхнею. Якщо поверхня мас

температурний градієнт, неправильне розташування датчика може призвести до помилок. Якщо датчик використовується в рідині, рідину слід перемішувати, щоб забезпечити рівномірний розподіл тепла. Датчики з інтегральною схемою зазвичай страждають від теплового зв'язку, оскільки їх важко встановити на поверхні.

- Постійна часу датчика. Ще одним джерелом помилок може бути час відгуку датчика. Кожному типу сенсора потрібен обмежений час, щоб відреагувати на зміну середовища. Помилки через постійну часу датчика можна мінімізувати шляхом покращення зв'язку між датчиком і поверхнею вимірювання.

Датчик положення

Датчики положення використовуються для вимірювання положення рухомих об'єктів. Ці датчики в основному бувають двох типів: датчики для вимірювання лінійного руху та датчики для вимірювання кутового руху. Найпоширенішим з усіх датчиків положення є потенціометр, оскільки це недорогий і простий у використанні датчик положення. Потенціометри доступні в лінійній і поворотній формах.

Потенціометр має контакт склоочисника, з'єднаний з механічним валом, який може бути кутовим (обертальний) або лінійним (тип повзунка) у своєму русі, і який викликає зміну значення опору між склоочисником/повзунком і двома кінцевими з'єднаннями, надаючи електричний сигнал вихід, який має пропорційне співвідношення між фактичним положенням склоочисника на резистивній доріжці та значенням його опору. Іншими словами, опір пропорційний положенню.

При використанні як датчика положення рухомий об'єкт підключається безпосередньо до обертового валу або повзунка потенціометра (рис.2.5).

					ДП. ДС-91. 07. 000 ПЗ	Ар
						24
Змн.	Арк.	№ докум.	Підпи	Дата		

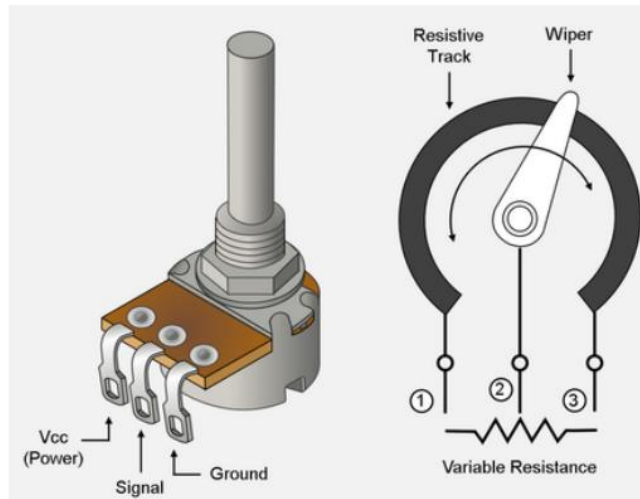


Рис.2.5Потенціометр як датчик положення

Ця конфігурація створює вихідний сигнал типу дільника потенціалу або напруги, який пропорційний положенню вала. Наприклад, якщо ви прикладаєте напругу, скажімо, 10 В до резистивного елемента потенціометра, максимальна вихідна напруга дорівнюватиме напрузі живлення при 10 Вольт, а мінімальна вихідна напруга дорівнюватиме 0 Вольтам. Тоді скоочисник потенціометра буде змінювати вихідний сигнал від 0 до 10 Вольт, причому 5 В вказує на те, що скоочисник або повзунок перебувають на півдорозі або в центральному положенні.

Поворотний потенціометр можна використовувати для вимірювання кутового положення.

Датчики положення потенціометричного типу недорогі, але їх недолік полягає в обмеженому діапазоні, а також у тому, що датчик може бути зношений через надмірне переміщення вала. Серед інших типів датчиків положення є ємнісні датчики, індуктивні датчики, лінійні змінні диференціальні трансформатори (LVDT) і оптичні кодери.

Електродвигуни

Електричний двигунце електрична машина, яка перетворює електричну енергію в механічну [5]. Більшість електродвигунів працюють через взаємодію

					ДП. ДС-91. 07. 000 ПЗ	Ар
						25
Змн.	Арк.	№ докум.	Підпи	Дата		

між магнітним полем двигуна та електричним струмом у обмотці дроту для створення сили у формі обертання вала. Електродвигуни можуть живитися від джерел постійного струму (DC), наприклад від акумуляторів, автомобілів або випрямлячів, або відзмінний струм(джерела змінного струму, наприклад електромережі, інвертори або електричні генератори. Анелектричний генератор механічно ідентичний електродвигуну, але працює у зворотному напрямку, перетворюючи механічну енергію в електричну.

Керування двигуном постійного струму

Двигун постійного струму(Двигун постійного струму – це широко використовуваний пристрій, який перетворює електричні імпульси в механічний рух. У двигуні постійного струму ми маємо тільки + і - виводи. Підключення їх до джерела постійної напруги рухає двигун в одному напрямку. Змінюючи полярність, двигун постійного струму буде рухатися в протилежному напрямку. Наприклад, маленькі вентилятори, які використовуються в багатьох материнських платах для охолодження ЦП, працюють від двигунів постійного струму. Коли проводи підключені до джерела напруги + і -, двигун постійного струму рухається безперервно.

Максимальна швидкість двигуна постійного струму вказується в об/хв (оборотів за хвилину) і наведено в таблиці даних. Двигун постійного струму має дві частоти обертання: без навантаження і під навантаженням. У технічному паспорті виробника вказано кількість обертів на холостому ході. Обороти на холостому ході можуть становити від кількох тисяч до десятків тисяч. Число обертів зменшується під час переміщення вантажу і зменшується зі збільшенням вантажу.

Наприклад, свердло, що обертає гвинт, має набагато нижчу швидкість обертів, ніж у режимі холостого ходу. Двигуни постійного струму також мають значення напруги та струму. Номінальна напруга - це напруга для цього двигуна за нормальних умов і може становити від 1 до 150 В, залежно від

					ДП. ДС-91. 07. 000 ПЗ	Ар
						26
Змн.	Арк.	№ докум.	Підпи	Дата		

двигуна. Коли ми збільшуємо напругу, оберти зростають. Номінальний струм — це споживаний струм, коли номінальна напруга подається без навантаження може бути від 25 мА до кількох ампер. Зі збільшенням навантаження кількість обертів на хвилину зменшується, якщо не збільшити струм або напругу, що подається на двигун, що, у свою чергу, збільшує крутний момент.

При фіксованій напрузі зі збільшенням навантаження струм (потужність), споживаний двигуном постійного струму, збільшується. Якщо ми перевантажимо двигун, він заглухне, і це може пошкодити двигун через тепло, яке утворюється внаслідок високого споживання струму. На рис.2.6 показано обертання двигуна постійного струму за годинниковою стрілкою (CW) і проти годинникової стрілки (CCW).

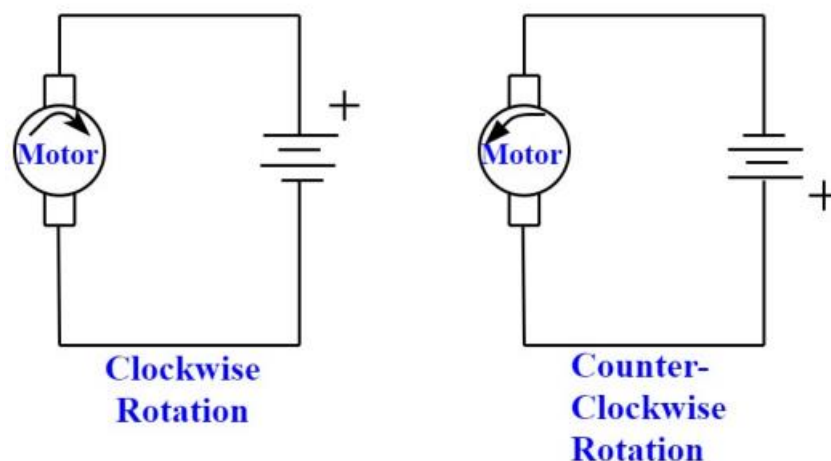


Рис.2.6 Обертання двигуном постійного струму

Керування кроковим двигуном

Асинхронний і безщітковий двигун постійного струму, який перетворює електричні імпульси в механічні рухи і, таким чином, обертається ступінчасто з певним кутом між кожним кроком для повного обертання, називається кроковим двигуном. Кут між кроками обертання крокового двигуна називається кутом кроку двигуна.

Крокові двигуни - це безщітчні двигуни постійного струму, які можуть обертатися від 0° до 360° кроками. Кроковий двигун використовує електронні сигнали для крокового обертання двигуна, і кожен сигнал обертає вал із фіксованим кроком (один крок). Кут повороту контролюється подачею певної послідовності сигналів. На відміну від серводвигуна, крокові двигуни можуть керуватися за допомогою контактів GPIO мікроконтролера, а не контактів ШІМ, і можуть обертатися ($+360^{\circ}$) і (-360°). Порядок сигналів визначає напрямок руху крокового двигуна за годинниковою стрілкою та проти неї. Щоб керувати швидкістю двигуна, нам просто потрібно змінити швидкість прикладених керуючих сигналів.

Крокові двигуни обертаються кроками. Це дуже корисно, оскільки його можна точно позиціонувати без будь-якого датчика зворотного зв'язку, який представляє контролер із розімкненим контуром. Кроковий двигун складається з ротора, який зазвичай є постійним магнітом, і оточений обмотками статора.

Коли ми крок за кроком активуємо обмотки в певному порядку та пропускаємо через них струм, вони намагнічують статор і створюватимуть електромагнітні полюси відповідно, що спричинятиме рух двигуна. Отже, це основний принцип роботи крокових двигунів. Існує кілька режимів кроків для роботи крокового двигуна, таких як повний крок, півкроку та мікрокрок.

Перший - це хвильовий період або збудження з однією котушкою (рис.2.7). У цьому режимі ми активуємо лише одну котушку за раз, що означає, що для цього прикладу двигуна з 4 котушками ротор здійснить повний цикл за 4 етапи.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						28
Змн.	Арк.	№ докум.	Підпи	Дата		

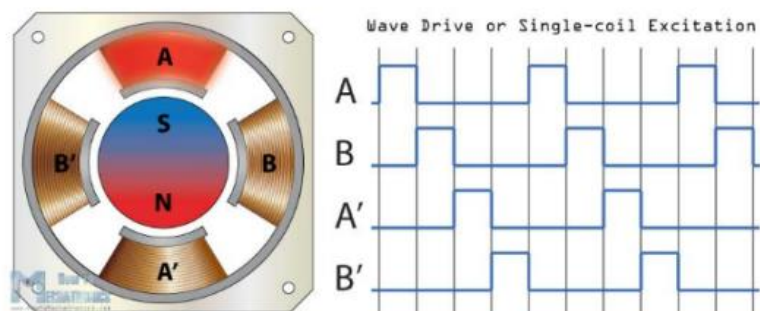


Рис.2.7 Хвильовий період

Повний крокрежим, який забезпечує набагато більший вихідний момент, оскільки ми завжди маємо 2 активні котушки в певний момент часу (рис.2.8). Однак це не покращує роздільну здатність крокового кроку, і знову ротор зробить повний цикл у 4 етапи.

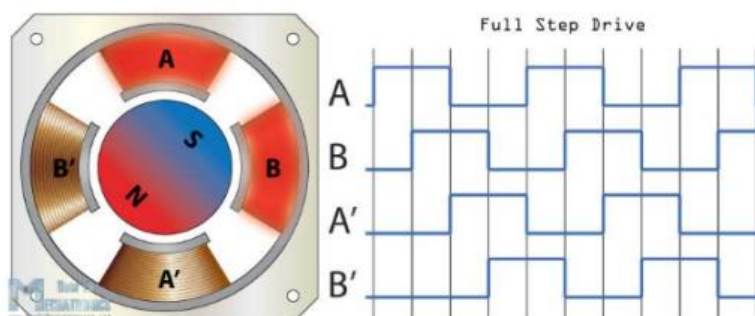


Рис.2.8 Повний крок

Для збільшення роздільної здатності степпера ми використовуємо напівкроковий драйв режим (рис.2.9). Цей режим насправді є комбінацією двох попередніх режимів. Тут ми маємо одну активну котушку, за якою йдуть 2 активні котушки, а потім знову одну активну котушку, за якою йдуть 2 активні котушки і так далі. Тож у цьому режимі ми отримуємо подвійну роздільну здатність із тією самою конструкцією. Тепер ротор зробить повний цикл у 8 кроків.

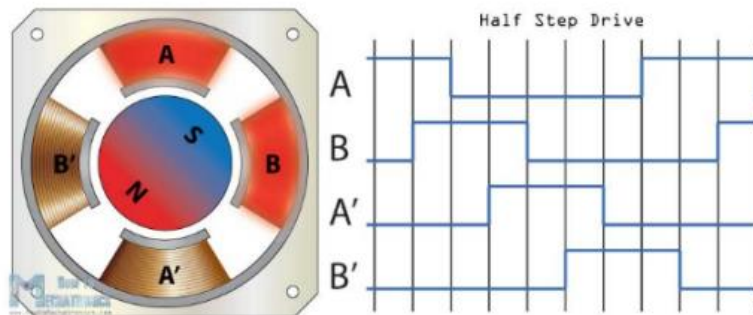


Рис.2.9 Напівкрок

Однак найпоширенішим способом керування кроковими двигунами в даний час є мікростепінг(рис.2.10). У цьому режимі ми забезпечуємо змінний контрольований струм катушок у формі синхвилі. Це забезпечить плавність руху ротора, зменшить напругу деталей і підвищить точність крокового двигуна.

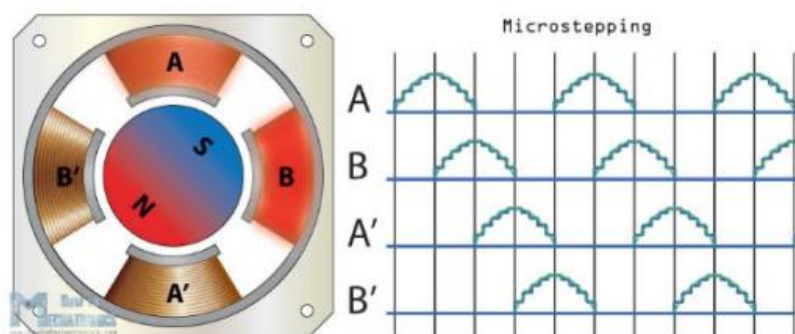


Рис.2.10. Мікрокрок

Іншим способом збільшення роздільної здатності крокового двигуна є збільшення кількості полюсів ротора та кількості полюсів статора(рис.2.11).

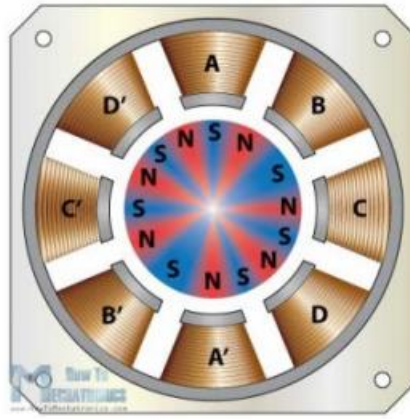


Рис.2.11 Збільшення кількості полюсів

На основі отриманих даних було обрано серводвигун MG90S та датчик температури та контролер струму.

2.3 Огляд та підбір комплектуючих

В ході роботи було розглянуто декілька варіантів мікропроцесорів та мікроконтролерів.

Мікропроцесор - це електронний компонент, який використовується комп'ютером для виконання його роботи. Це центральний процесор на одній інтегральній мікросхемі, що містить мільйони дуже маленьких компонентів, включаючи транзистори, резистори та діоди, які працюють разом [6]. Мікропроцесор — це багатоцільова цифрова інтегральна схема на основі регістрів, керована тактовою частотою, яка приймає двійкові дані як вхідні дані, обробляє їх відповідно до інструкцій, збережених у його пам'яті, і надає результати (також у двійковій формі) як вихідні дані. Мікропроцесори містять як комбінаційну, так і послідовну цифрову логіку. Мікропроцесори оперують числами і символами, представленими в двійковій системі числення.

Реєстр загального призначення

					ДП. ДС-91. 07. 000 ПЗ	Ар
						31
Змн.	Арк.	№ докум.	Підпи	Дата		

В архітектурі комп'ютера реєстр процесора це швидкодоступне місце, доступне для центрального процесора (ЦП) комп'ютера. Регістри зазвичай складаються з невеликої кількості швидкого зберігання, хоча деякі регістри мають спеціальні апаратні функції, і, можливо, лише читання або лише запис. Регістри зазвичай звертаються за допомогою механізмів, відмінних від основної пам'яті.

Регістри процесора зазвичай знаходяться на вершині ієрархії пам'яті і забезпечують найшвидший спосіб доступу до даних. Регістри зазвичай вимірюються кількістю бітів, які вони можуть містити, наприклад, «8- бітний регістр», «32-бітний регістр» або «64-бітний регістр» або навіть більше. Процесор часто містить кілька типів регістрів, які можна класифікувати відповідно до їх вмісту або інструкцій, які з ними працюють:

1. Доступні для користувача регістри можна читати або записувати машинними інструкціями. Найпоширенішим поділом доступних для користувача реєстрів є регістри даних і регістри адрес.

2. Регістри даних можуть містити числові значення даних, такі як цілі числа та, в деяких архітектурах, значення з оцінкою, а також символи, малі бітові масиви та інші дані. У деяких старих і недорогих ЦП спеціальний регістр даних, відомий як накопичувач, використовується неявно для багатьох операцій.

3. Адресні регістри містять адреси та використовуються інструкціями, які опосередковано звертаються до первинної пам'яті. Деякі процесори містять регістри, які можуть використовуватися лише для зберігання адреси або лише для зберігання числових значень (у деяких випадках використовуються як індексний регістр, значення якого додається як набір 0 з деякої адреси), інші дозволяють регістрам зберігати будь-який тип величини . Існує велика різноманітність можливих режимів адресації, які використовуються для визначення ефективної адреси операнда.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						32
Змн.	Арк.	№ докум.	Підпи	Дата		

Блок керування

Блок керування (CU) — це компонент центрального процесора (CPU) комп'ютера, який керує роботою процесора. Він повідомляє пам'яті комп'ютера, арифметичному і логічному блоку, а також пристроям введення і виведення, як реагувати на інструкції, надіслані до процесора. Він керує роботою інших блоків, надаючи сигнали синхронізації та керування. Більшість комп'ютерних ресурсів управляються МС. Він спрямовує потік даних між ЦП та іншими пристроями.

Мікропроцесорна система

Один лише мікропроцесор нічого не може зробити. Тож для виконання певної роботи йому потрібно об'єднатися з іншими пристроями. Призначення мікропроцесора полягає в тому, щоб виконувати математичні обчислення (обчислення) штучним способом. Поєднання мікропроцесора з пам'яттю та периферійними пристроями призводить до створення системи, яка призначена для керування іншими пристроями або швидкого виконання обчислень. Усі мікропроцесорні системи виконують однакові основні функції, тобто введення, зберігання, обробку та виведення даних або сигналів (рис.2.12).

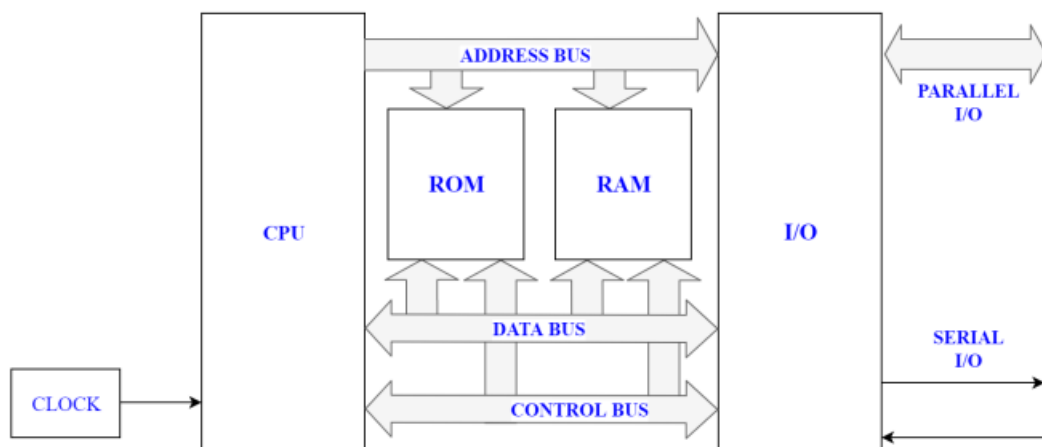


Рис.2.12 Процесорна схема

Шини

Шини - це група провідних проводів, які передають інформацію, усі периферійні пристрої підключені до мікропроцесора через шину. Є три види шин.

Адресна шина - це група провідних проводів, які несуть лише адресу. Адресна шина є односпрямованою, оскільки дані надходять в одному напрямку: від мікропроцесора до пам'яті або від мікропроцесора до пристроїв введення/виведення. Довжина шини адреси визначає обсяг пам'яті, до якої може звертатися система. Наприклад, система з 32-розрядною адресною шиною може адресувати 232 місця пам'яті. Якщо кожне місце пам'яті містить один байт, адресний простір пам'яті становить 4 ГБ.

Шина даних - це група провідних проводів, які передають лише дані. Шина даних є двонаправленою, оскільки дані надходять в обох напрямках: від мікропроцесора до пам'яті або пристроїв введення/виведення та від пам'яті або пристроїв введення/виведення до мікропроцесорів. Ширина шини даних безпосередньо пов'язана з найбільшим числом, яке може нести шина, наприклад, 8-бітна шина може представляти 2 у ступені 8 унікальних значень, це дорівнює числу від 0 до 255. 16-бітна автобус може перевозити від 0 до 65535.

Шина керування - це група провідних проводів, яка використовується для генерації синхронізації та керуючих сигналів для керування всіма пов'язаними периферійними пристроями. Мікропроцесор використовує керуючу шину для обробки даних, тобто те, що потрібно робити з вибраним місцем пам'яті. Деякі контрольні сигнали:

- Читання з пам'яті;
- Запис пам'яті;
- I/O читання;
- Запис введення/виведення;
- Отримання коду операції.

Внутрішня будова мікроконтролера

Мікроконтролер - це комп'ютерна система на чіпі, яка виконує певну роботу. Він містить інтегрований процесор, пам'ять і програмовані периферійні пристрої вводу/виводу, які використовуються для взаємодії з елементами, підключеними до мікросхеми (рис.2.13). Мікроконтролер відрізняється від мікропроцесора, який містить лише центральний процесор.

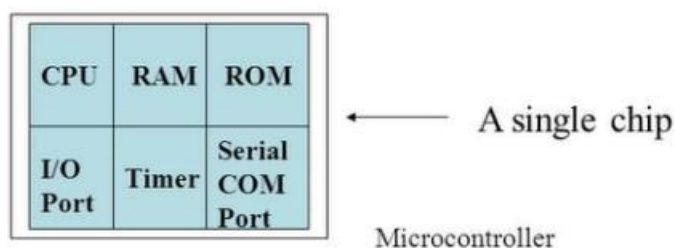


Рис.2.13 Мікроконтролер

Основними елементами мікроконтролера є:

- Процесор(CPU) - процесор можна розглядати як мозок пристрою. Це обробляє та відповідає на різні інструкції, які керують функціями мікроконтролера. Це передбачає виконання основних арифметичних, логічних операцій і операцій вводу/виводу. Він також виконує операції передачі даних, які передають команди іншим компонентам у більшій вбудованій системі.

- Пам'ять - Пам'ять мікроконтролера використовується для зберігання даних процесора отримує і використовує для відповіді на інструкції, на виконання яких він запрограмований. Мікроконтролер має два основних типи пам'яті:

1. Програмна пам'ять, який зберігає довготривалу інформацію про інструкції які виконує ЦП. Пам'ять програм є енергонезалежною пам'яттю, тобто вона зберігає інформацію протягом тривалого часу без потреби в джерелі живлення.

2. Пам'ять даних, який потрібен для тимчасового зберігання даних під час виконання інструкцій виконуються. Пам'ять даних є енергонезалежною, тобто

дані, які в ній зберігаються, є тимчасовими та зберігаються, лише якщо пристрій підключено до джерела живлення.

3. Периферійні пристрої введення/виведення

4. Пристрої введення та виведення є інтерфейсом для процесора до зовнішнього світу.

5. Вхідні порти отримують інформацію і передають її процесору у вигляді двійкових даних.

6. Процесор отримує ці дані та надсилає необхідні інструкції до пристроїв виведення, які виконують зовнішні завдання щодо мікроконтролера.

У табл. 2.1 наведено порівняльну характеристику між мікроконтролером та мікропроцесором.

Таблиця 2.1

Порівняння мікропроцесорів та мікроконтролерів

Мікропроцесор	Мікроконтролер
Це центральний процесор на одному інтегрованому чіпі на основі кремнію.	Це побічний продукт розвитку мікропроцесора з ЦП разом з іншими периферійними пристроями.
Він не має RAM, ROM, модулів вводу-виводу, таймерів та інших периферійних пристроїв на чіпі.	Він має центральний процесор разом з оперативною пам'яттю, ПЗУ та іншими периферійними пристроями, вбудованими в один чіп.
Це серце комп'ютерної системи.	Це мозок комп'ютерної системи.
Він використовує зовнішню шину для підключення до RAM, ROM, контактів вводу-виводу та інших периферійних пристроїв.	Він використовував внутрішню керуючу шину, яка недоступна для розробника плати.
Він використовується для програм загального призначення, які можуть обробляти велику кількість даних.	Використовується для систем із спеціальними програмами.
Складний у виробництві та дороговартісний.	Простий у виробництві і значно дешевший.

Порівняння мікроконтролерів

Raspberry Pi Pico - це вступ до світу Інтернету речей (IoT) та мікроконтролерів. Це дружній для початківців мікроконтролерний модуль, що використовує MicroPython. RP2040, який знаходиться в основі Pico, є добре розробленим мікропроцесором, який може бути використаний у практично будь-якому проекті IoT. Він має достатньо потужності для ефективного виконання завдань. Центральний процесор (CPU) в двоядерному процесорі - це ARM M0+. Pico є вигідним вибором для студентів, які шукають дошку для різноманітних проектів, які не потребують підключення до Wi-Fi, але мають всі необхідні можливості за доступною ціною, навіть близько \$4.

ESP32, з іншого боку, є компактною, але потужною мікроконтролерною дошкою. Завдяки своїй низькій вартості, енергоефективності та компактному розміру, вона є добре розробленою мікросхемою, яка може бути використана в різних застосуваннях IoT. Однак, завдяки можливостям бездротового з'єднання та іншим цікавим функціям, ця мікроконтролерна дошка зарекомендувала себе як одна з найкращих для розробників IoT.

Обидві мікроконтролерні дошки мають свої переваги і можуть бути використані в проектах IoT в залежності від конкретних вимог та потреб. Вибір між ними залежить від функціональності, вартості, розміру та специфічних вимог проекту.

Обидва мікроконтролери мають свої переваги і можуть бути використані в проектах Інтернету речей (IoT) в залежності від специфічних потреб і вимог. Вибір між ними залежить від функціональності, вартості, розміру та особливих вимог проекту.

Існує кілька відмінностей між мікроконтролерами RP2040 і STM32, включаючи тип мікроконтролера та серію процесора. RP2040 є двоядерним мікроконтролером з двома ядрами ARM Cortex-M0+, тоді як STM32 є одноядерним з одним ядром ARM Cortex-M.

Також RP2040 має більший обсяг оперативної пам'яті (RAM) - 264 кілобайти порівняно з 20 кілобайтами у STM32. Крім того, RP2040 має більший

					ДП. ДС-91. 07. 000 ПЗ	Ар
						37
Змн.	Арк.	№ докум.	Підпи	Дата		

обсяг флеш-пам'яті, ніж STM32, з 64 кілобайтами флеш-пам'яті. Проте, робоча температура STM32 є вищою, з діапазоном від -40°C до +85°C, порівняно з діапазоном від -20°C до +85°C у RP2040. Таким чином, можна сказати, що RP2040 переважає STM32. Варто враховувати, що вибір між RP2040 і STM32 залежить від конкретних вимог проекту і потреб користувача. Кожен з цих мікроконтролерів має свої переваги і може бути підходящим для певних сценаріїв використання.

RP2040 та ATMEGA328 мають кілька відмінностей, які варто відзначити. Щодо розміру оперативної пам'яті, ATMEGA328 має скромні 2 кілобайти, тоді як RP2040 може похвалитися набагато більшим обсягом - 264 кілобайти. Розмір флеш-пам'яті також значно відрізняється, де ATMEGA328 має 32 кілобайти, а у RP2040 - вражаючі 2 мегабайти. Ще одна помітна відмінність полягає у напрузі живлення, оскільки ATMEGA328 працює при 5 вольтах постійного струму, тоді як RP2040 працює при 3,3 вольтах постійного струму. Ці основні відмінності підкреслюють різні можливості цих двох мікроконтролерів.

ATmega328 працює на основі одноядерних 8-бітних мікросхем Atmel ATmega328P з тактовою частотою 16 МГц. Він має 2 кілобайти оперативної пам'яті SRAM і скромні 28 кілобайт флеш-пам'яті. У порівнянні з цим, RP2040 в Raspberry Pi Pico володіє потужним 32-бітним ядром Arm Cortex-M0+, здатним працювати на налаштовуваній частоті до 133 МГц. Завдяки значно більшій потужності та можливостям, RP2040 є більш розширеним і продуктивним рішенням.

Вибір плати

На основі отриманих даних та згідно вимог до функціоналу мікроконтролером було обрану плату Raspberry Pi Pico H. Дана плата має всі необхідні характеристики для нашого робота.

Raspberry Pi - це мініатюрний одноплатний комп'ютер. Незважаючи на свої компактні розміри, ця плата має високу продуктивність, яка дозволяє їй

					ДП. ДС-91. 07. 000 ПЗ	Ар
						З
Змн.	Арк.	№ докум.	Підпи	Дата		

конкурувати зі звичайними настільними ПК. Спочатку Raspberry Pi була розроблена як навчальний посібник з інформатики, але її ідея виявилася настільки успішною, що за кілька років цей міні-комп'ютер став популярним серед широкого кола користувачів. З часом Raspberry Pi пройшла кілька модифікацій, кожна з яких мала свої особливості, і ми обрали модель Raspberry Pi Pico H. Такий підхід дозволяє налаштовувати вартість пристрою залежно від потреб користувача, що також сприяє його популярності. Усі моделі Raspberry Pi використовують процесори на основі архітектури ARM, яка відзначається високою ефективністю та надійністю. На рис 2.14 показаний зовнішній вигляд однієї з популярних плат Raspberry Pi Pico H.

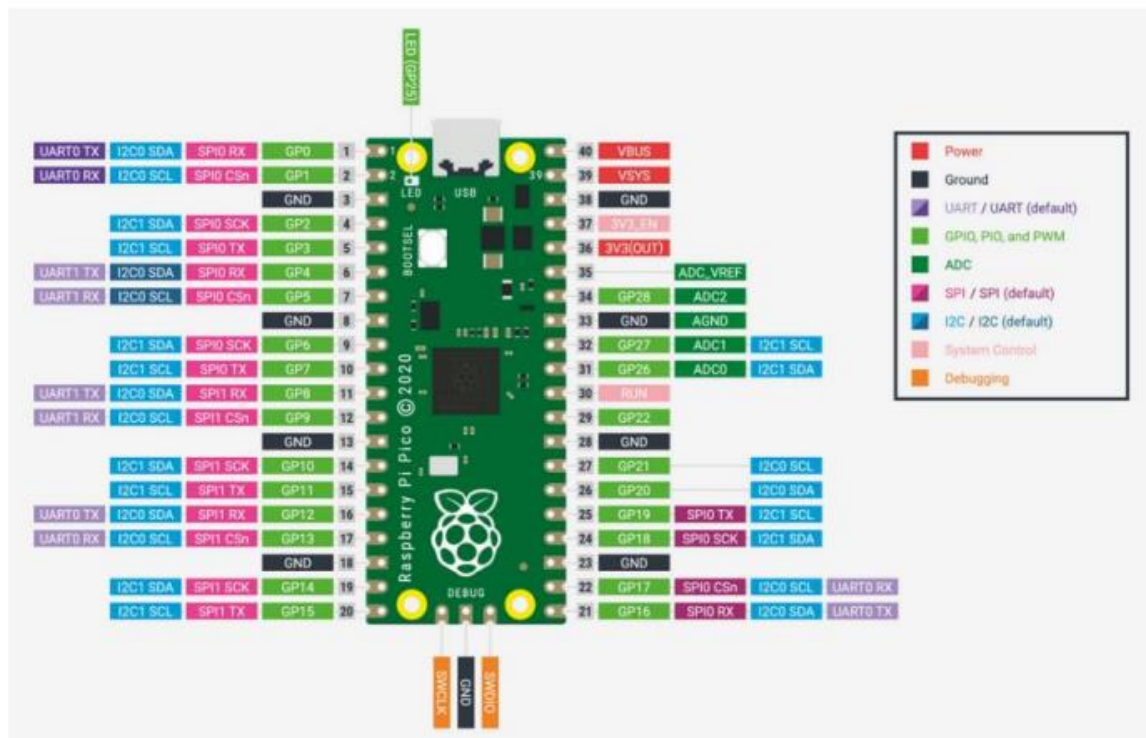


Рис.2.14 Плата Raspberry Pi Pico H.

На сьогоднішній день (період 2012-2022.) Існує 13 різновидів Raspberry Pi. Останні версії оснащені бездротовими Wi-Fi і Bluetooth модулями, що розширюють межі застосування міні-пк в області Ethernet-технологій. Нижче

наведено порівняльну таблицю, в якій відображені особливості кожної модифікації із зазначенням деяких технічних даних.

На рис 2.15 зображено останню на момент написання статті модифікацію Raspberry Pi 4B, запущену у продаж у червні 2019р. Вона має додатковий графічний процесор VideoCore VI (OpenGL ES 3.x), а також апаратний декодер 4Kp60 для відтворення HEVC відео. Два порти microHDMI з можливістю пропускати сигнал до 4K, дозволяють підключити одночасно два монітори.



Рис.2.15 Raspberry Pi 4B

Головна відмінність Raspberry Pi від звичайних комп'ютерів полягає у наявності програмованих портів введення-виведення GPIO. Ці порти дозволяють керувати зовнішніми пристроями та отримувати дані з різноманітних датчиків. Модифікація Raspberry Pi H, яку ми використовуємо, базується на тому ж процесорі, але вона не має портів для підключення периферійних пристроїв, а лише пропонує порти GPIO для взаємодії з ними.

Робота з GPIO Raspberry Pi

Як уже говорилося раніше, основною відмінністю Raspberry Pi від звичайного ПК є наявність на платі портів загального призначення GPIO (General-purpose input/output). Користувачеві доступна можливість керування

					ДП. ДС-91. 07. 000 ПЗ	Ар
						40
Змн.	Арк.	№ докум.	Підпи	Дата		

цими виводами, а це означає, що до Raspberry Pi можна підключати дисплеї, кнопки, датчики, реле та інші електронні модулі, які ми і будемо використовувати в нашому проекті.

Зовні GPIO виконаний у вигляді дворядної штирьової колодки з кроком 2,54 мм, яка розташована на краю плати. Ранні моделі, такі як В та А містять 26 виводів, а більш сучасні – 40. На рис 2.16показаний зовнішній вигляд портів загального призначення для плати Raspberry Pi 3B+ із зазначенням нумерації виводів.



Рис 2.16 Зовнішній вигляд портів на платі

Але щоб повноцінно використовувати GPIO, знати їх нумерацію недостатньо. Необхідно точно розуміти, де розташований той чи інший висновок, як він називається і за що він відповідає. На рис 2.17 наведено повне розпинання роз'ємів GPIO для різних модифікацій Raspberry Pi.

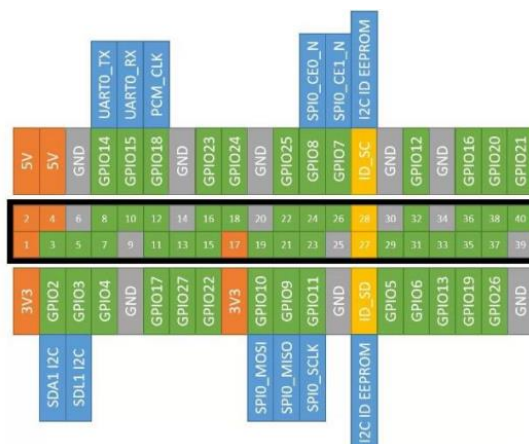


Рис. 2.17 Розміщення пінів на платі

Як видно з наведеного вище рисунка, на колодці крім самих GPIO виведені штирі з напругою 3,3V, 5V, а також виводи GND. Деякі GPIO мають альтернативні функції, призначення яких вказано у синіх блоках. До того ж не можна порушувати здатності навантаження порту, щоб не вивести Raspberry Pi з ладу. Слід пам'ятати, що GPIO працює з напругою 3,3V та максимальним струмом навантаження 50mA на один висновок. Це означає, що будь-яке перевищення зазначених параметрів негативно позначиться на працездатності плати, тому краще використовувати гальванічну розв'язку між GPIO та зовнішнім виконавчим пристроєм. Те саме стосується і вхідних ланцюгів, до яких застосовуються резистивні дільники і всілякі перетворювачі рівнів.

Електричні характеристики Raspberry Pi Pico

Таблиці та графіки споживання електроенергії показують споживання струму VBUS на трьох типових Raspberry Pi Pico пристроїв із чотирма різними варіантами використання програмного забезпечення. Ці результати не є гарантованими максимальними значеннями; вони є індикація поточного споживання, яке користувач зазвичай може очікувати від пристрою під час використання в цих сценаріях. Для отримання більш детальних даних про поточне споживання дивіться технічну таблицю RP2040.

3. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ

3.1 Опис алгоритму роботи системи керування людиноподібним роботом

Система керування корпусом людиноподібного робота реалізована за допомогою програмного забезпечення та різних компонентів. Пульти радіокерування відправляють сигнал до плати керування, яка відповідно обробляє цей сигнал і вмикає автоматичну систему керування.

Автоматична система керування включає в себе різні компоненти, зокрема нагрівачі пластини та рухомі елементи корпусу, які приводяться в рух за допомогою серводвигунів. Нагрівачі пластини відповідають за регулювання температури в роботі, а рухомі елементи корпусу дозволяють роботу здійснювати рухи і виконувати необхідні дії. На рис 3.1 наведена блок-схема принципу роботи.

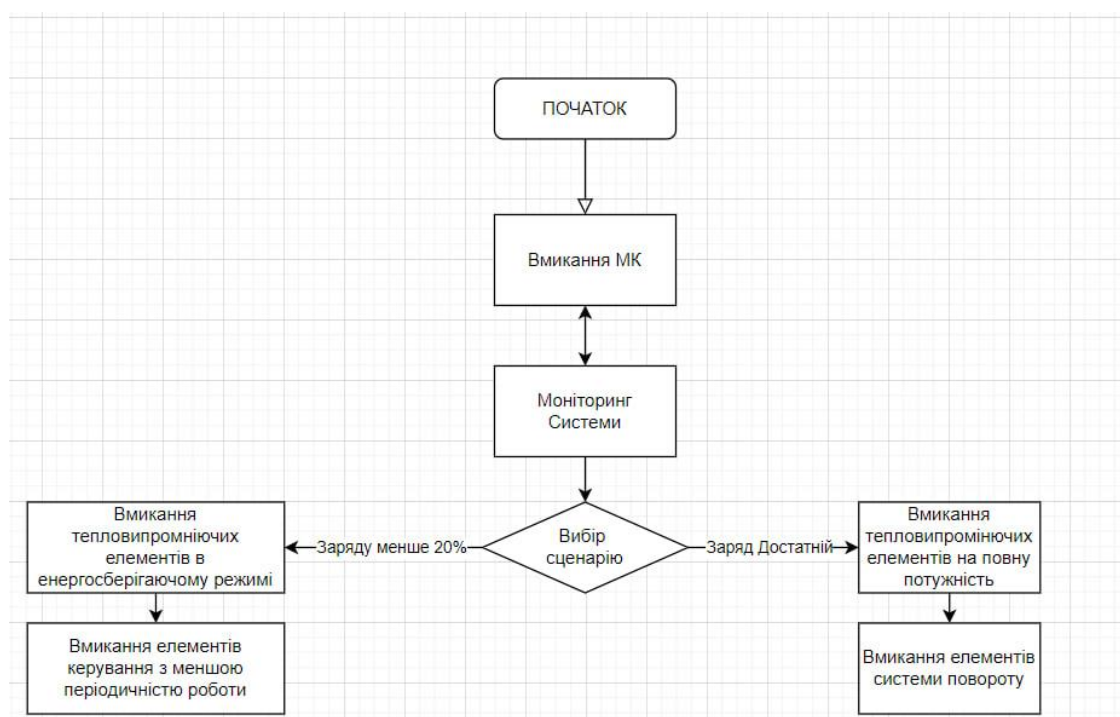


Рис.3.1. Блок-схема принципу роботи

ШИМ

Широтно імпульсна модуляція (ШИМ) - це схема, в якій цифровий сигнал забезпечує плавно змінну середню напругу. Це досягається за допомогою позитивних імпульсів з контрольованою шириною через регулярні інтервали. Відсоток часу, проведеного в стані "високий", відомий як відсоток заповнення. Це може використовуватися для наближення аналогового виходу або керування комутаційною електронікою.

Блок ШИМ RP2040 має 8 ідентичних сегментів. Кожен сегмент може керувати двома вихідними сигналами ШИМ або вимірювати частоту або відсоток заповнення вхідного сигналу. Це дозволяє мати до 16 керованих вихідних сигналів ШИМ. Всі 30 виводів GPIO можуть бути керовані блоком ШИМ.

Кожен сегмент ШИМ оснащений наступним:

- 16-розрядним лічильником
- Дробовим годинниковим дільником 8.4
- Двома незалежними вихідними каналами з можливістю налаштування коефіцієнта заповнення від 0% до 100% (включно)
- Подвійною зміною нахилу і зворотнім краєм модуляції
- Режимом вхідного сигналу, чутливим до фронту, для вимірювання частоти
- Режимом вхідного сигналу, чутливим до рівня, для вимірювання коефіцієнта заповнення
- Налаштовуваним значенням обгортки лічильника
- Регістри обгортки і рівня мають подвійне буферування і можуть змінюватись без переконання під час роботи ШИМ
- Запит на переривання та запит DMA при обгортанні лічильника
- Фазу можна точно переміщати вперед або назад під час роботи (в наростаннях одиниць лічильника)

					ДП. ДС-91. 07. 000 ПЗ	Ар
						44
Змн.	Арк.	№ докум.	Підпи	Дата		

Сегменти можуть бути ввімкнені або вимкнені одночасно за допомогою єдиного глобального регістра керування. При цьому сегменти працюють у синхронному режимі, щоб виходи кількох сегментів могли перемикаєти більш складні схеми живлення.

Апаратна реалізація ШІМ працює шляхом постійного порівняння вхідного значення з лічильником, що безперервно працює. Це призводить до переключення вихідного сигналу, де час, проведений на високому рівні вихідного сигналу, пропорційний вхідному значенню. Відсоток часу, проведеного на високому рівні сигналу, відомий як відсоток заповнення (duty cycle) сигналу.

Період лічення контролюється регістром TOP, з максимально можливим періодом у 65536 тактів, оскільки розмір лічильника і TOP становить 16 біт. Вхідні значення налаштовуються за допомогою регістра CC (рис.3.2).

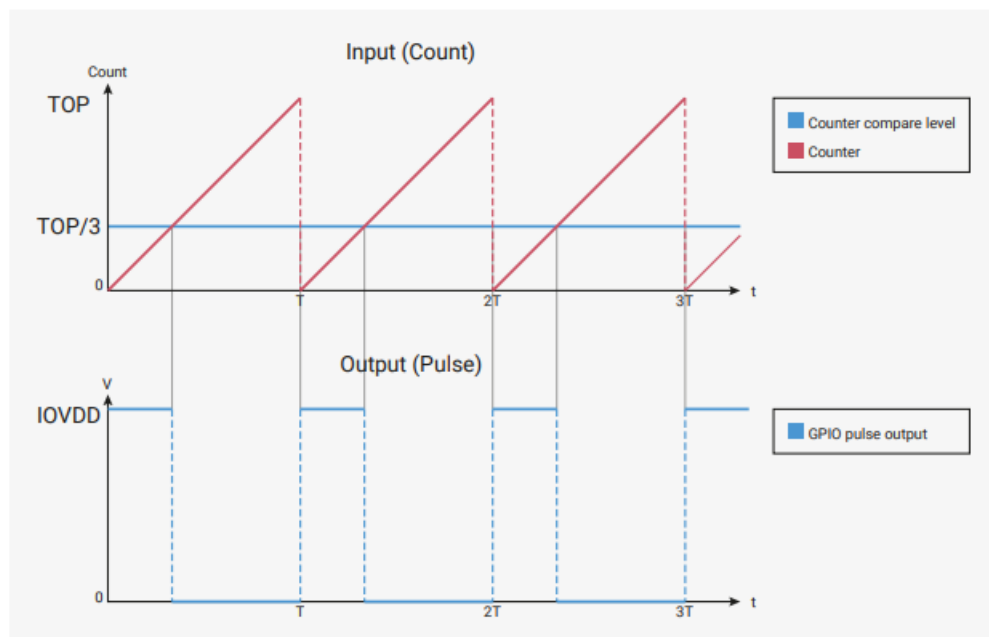


Рис.3.2 Період лічення

Цей приклад показує налаштування періоду лічення, а також рівнів порівняння лічильників A і B на одному з сегментів ШІМ у мікросхемі RP2040.

```

15 // Tell GPIO 0 and 1 they are allocated to the PWM
16 gpio_set_function(0, GPIO_FUNC_PWM);
17 gpio_set_function(1, GPIO_FUNC_PWM);
18
19 // Find out which PWM slice is connected to GPIO 0 (it's slice 0)
20 uint slice_num = pwm_gpio_to_slice_num(0);
21
22 // Set period of 4 cycles (0 to 3 inclusive)
23 pwm_set_wrap(slice_num, 3);
24 // Set channel A output high for one cycle before dropping
25 pwm_set_chan_level(slice_num, PWM_CHAN_A, 1);
26 // Set initial B output high for three cycles before dropping
27 pwm_set_chan_level(slice_num, PWM_CHAN_B, 3);
28 // Set the PWM running
29 pwm_set_enabled(slice_num, true);

```

Стандартна поведінка сегмента ШІМ полягає в тому, що лічильник збільшується досягненням значення регістра TOP, а потім миттєво обгортається до 0 (рис.3.3). Сегменти ШІМ також пропонують режим фазової корекції, який активується шляхом встановлення значення 1 для регістра CSR_PH_CORRECT, при якому лічильник починає зменшуватися після досягнення значення TOP і продовжує зменшуватися до досягнення значення 0.

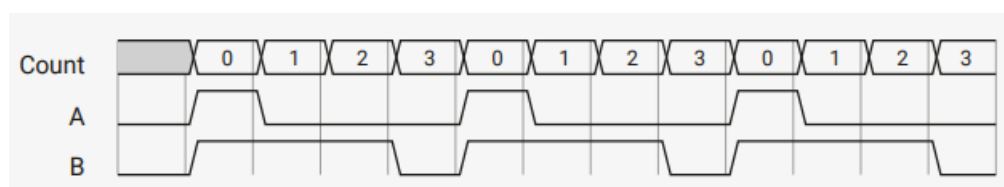


Рис.3.3 Апаратний модуль ШІМ

Режим фазової корекції названий таким, оскільки пульс завжди центрований в одній точці, незалежно від відсотка заповнення. Іншими словами, його фаза не залежить від відсотка заповнення. Частота виходу зменшується вдвічі, коли включений режим фазової корекції.

Дільник тактової частоти

Дільник тактової частоти (Clock Divider) — це елемент апаратного забезпечення, який використовується для поділу вхідного тактового сигналу на меншу частоту. Він зазвичай використовується для зниження швидкості роботи системи або для синхронізації різних компонентів.

Після налаштування дільника тактової частоти, він починає ділити вхідний тактовий сигнал на більш низьку частоту відповідно до заданого дільника. Результатом є новий тактовий сигнал з меншою частотою, який може бути використаний для синхронізації роботи різних компонентів системи.

Наприклад, якщо вхідний тактовий сигнал має частоту 100 МГц, і ви встановлюєте дільник тактової частоти на значення 10, вихідний тактовий сигнал буде мати частоту 10 МГц ($100 \text{ МГц} / 10$). Це означає, що кожен перший тактовий сигнал буде виконуватись лише після десяти вхідних тактових сигналів.

Використання дільника тактової частоти дозволяє контролювати швидкість роботи системи та узгоджувати роботу різних компонентів, що забезпечує ефективну та синхронізовану роботу апаратного забезпечення.

Кожний сегмент має дробовий дільник тактової частоти, який налаштовується за допомогою регістра DIV. Це 8-бітовий цілий і 4-бітовий дробовий дільник тактової частоти, який дозволяє зменшити швидкість рахунку на до 256 разів. Дільник тактової частоти дозволяє досягти набагато нижчих вихідних частот, наприклад, близько 7.5 Гц з системним тактовим сигналом 125 МГц. Для досягнення ще нижчих частот потрібен переривання системного таймера.

					ДП. ДС-91. 07. 000 ПЗ	Ар.
						42
Змн.	Арк.	№ докум.	Підпи	Дата		

Для цього генерується сигнал увімкнення (enable signal), який вимикає роботу лічильника. Цей сигнал дозволяє зупинити лічильник і забезпечити зниження частоти виходу (рис.3.4)

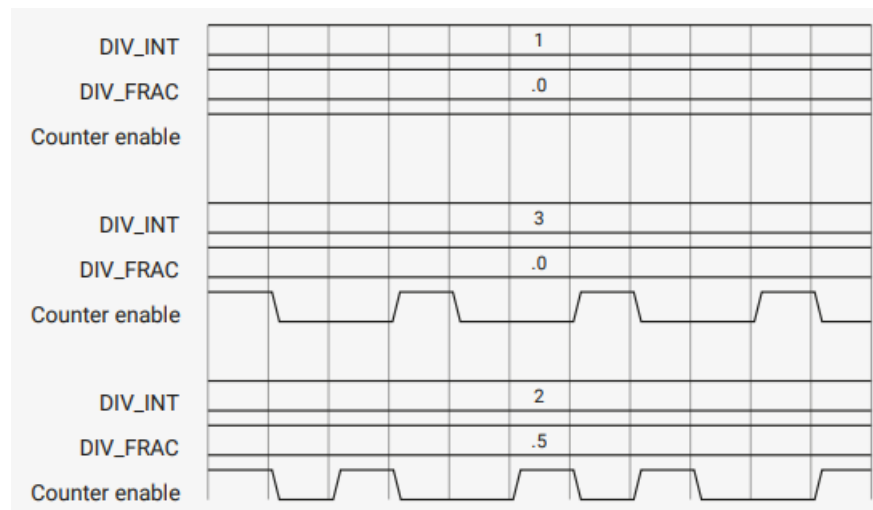


Рис.3.4 Сигнал увімкнення

3.2 Розробка програмного забезпечення

Як програмувати RP2040

RP2040 можна програмувати за допомогою одного з двох підходів. Інструкції створені для розробницької плати Pico для Raspberry Pi, але вони також підходять для інших плат з RP2040. Просто переконайтеся, що ви змінили визначення пінів, відображаючи розподіл GPIO.

Python і C/C++ можна використовувати для програмування нової плати Pico H. Ви можете програмувати у спрощеному варіанті Python за допомогою середовища розробки Thonny або Raspbian. Також можна використовувати середовище розробки Arduino IDE для програмування плати Pico H.

Використання АЦП

АЦП RP2040 не має вбудованого опорного джерела, тому використовує власне джерело живлення як опорний. На Піко висновок ADC_AVDD

(живлення АЦП) генерується з SMPS 3,3 В за допомогою R-C фільтра (201 Ом на 2,2 мкФ). Це просте рішення, але має такі недоліки:

1. Ми покладаємося на вихідну точність 3,3 В SMPS, яка не є великою
2. АЦП споживає струм (приблизно 150 мкА, якщо діод датчика температури вимкнено, але це змінюється від мікросхеми до мікросхеми) і тому буде властиве зміщення приблизно $150 \text{ мкА} \cdot 200 = \sim 30 \text{ мВ}$. Є невелика різниця в струмі малюють, коли АЦП виконує вибірку (приблизно +20 мкА), так що зміщення також змінюватиметься залежно від вибірки, а також роботи температура.

Зміна опору між ADC_VREF і виводом 3V3 може зменшити зміщення за рахунок збільшення шуму - що може бути добре, особливо якщо варіант використання підтримує усереднення за кількома зразками. Переведення високого рівня на контакт режиму SMPS (GPIO23), щоб перевести джерело живлення в режим ШІМ, може значно зменшити внутрішній пульсації SMPS при невеликому навантаженні, і, отже, пульсації на живленні АЦП. Це зменшує енергоефективність плата при невеликому навантаженні, тому режим PFM з низьким енергоспоживанням можна повторно ввімкнути між нечастими вимірюваннями АЦП GPIO23 знову низький. Зміщення АЦП можна зменшити, прив'язавши другий канал АЦП до землі та використовуючи це нульове вимірювання як наближення до зсуву. Для значно покращеної продуктивності АЦП зовнішній шунтовий джерело опору 3,0 В, наприклад LM4040, можна підключити до Вивід ADC_VREF на землю. Зауважте, що якщо це зробити, діапазон АЦП обмежено сигналами 0–3,0 В (а не 0–3,3 В), а опорний шунт пропускатиме безперервний струм через резистор фільтра $200 \text{ Ом} \cdot (3,3 - 3,0 \text{ В}) / 200 = \sim 1,5 \text{ мА}$. Зауважте, що резистор 1 Ом на Pico (R9) розроблений, щоб (можливо) допомогти з опорним шунтом, який інакше став би нестабільний при безпосередньому підключенні до 2,2 мкФ. Це також забезпечує невелику фільтрацію навіть у випадку, якщо 3,3 В і ADC_VREF замикаються разом (що є дійсним, якщо ви не дбаєте про шум і хочете зменшити властиве зміщення).

					ДП. ДС-91. 07. 000 ПЗ	Ар 42
Змн.	Арк.	№ докум.	Підпи	Дата		

Нарешті, R7 є фізично великим метричним резистором 1608 (0603), тому його можна відносно легко видалити, якщо користувач хоче ізолювати ADC_VREF і робити власну справу з напругою АЦП, наприклад живлячи його від повністю окремої напруги (наприклад, 2,5 В). Зверніть увагу, що АЦП на RP2040 був кваліфікований лише на 3,0/3,3 В, але повинен працювати приблизно до 2 В.

Вибір мови програмування та середовища

Хоча Raspberry Pi зовні може нагадати нам Arduino, він таки використовує кардинально інший метод функціонування. Ця плата, як і стандартний ПК, працює під керівництвом однієї зі спеціалізованих операційних систем. Залежно від сфери застосування або особистих симпатій, кожен може вибрати для себе свою. Нижче наведено перелік найпопулярніших «операцій» для Raspberry Pi з їх коротким описом.

Raspbian - ця операційна система в 2015 році була представлена як основна для Raspberry Pi. Вона максимально оптимізована для процесорів з ARM-архітектурою і досить активно продовжує розвиватися. Основою операційної системи є Debian GNU/Linux. Середовище робочого столу складається з LXDE (середовище для UNIX та інших POSIX-сумісних систем типу Linux та BSD), а також менеджера вікон Openbox (безкоштовний менеджер для X Window System). До складу дистрибутива входять: програма комп'ютерної алгебри Mathematica; модифікована версія Minecraft PI; урізана версія Chrome.

Debian – це операційна система з відкритим вихідним кодом. До складу Debian входить понад 59 000 пакетів вже скомпільованого ПЗ. Система використовує ядро Linux чи FreeBSD. У стандартний дистрибутив включені: середовище робочого столу GNOME з набором найпопулярніших програм, таких як Firefox, LibreOffice, Evolution та інший набір для роботи з мультимедіа. Також є можливість установки образів із середовищем робочих столів KDE, Xfce, LXDE, MATE і Cinnamon.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						50
Змн.	Арк.	№ докум.	Підпи	Дата		

Ubuntu – система заснована на Debian GNU/Linux. За популярністю Ubuntu посідає перше місце серед дистрибутивів Linux, призначених для веб-серверів. До складу дистрибутива входять: - програма для перегляду Інтернет; офісний пакет, програми для комунікації та ін.

Fedora – ця операційна система ґрунтується на дистрибутиві Linux від відомої фірми Red Hat. До складу дистрибутива входять LibreOffice, Mozilla Firefox, а також інше програмне забезпечення, яке можна додатково встановити через Центр Додатків GNOME.

Arch Linux – це дистрибутив GNU/Linux, що вільно розповсюджується, загального призначення. Особливістю даної системи є відсутність графічного установника, що може неабияк потренувати навички затятих дослідників Linux.

Gentoo Linux – один із популярних дистрибутивів GNU/Linux із гнучкою технологією управління пакетами. У системі передбачено можливість максимальної оптимізації під конкретне апаратне рішення. Алгоритм керування пакетами дає можливість легко реалізувати як робочу станцію, і сервер. RISC OS - операційна система спеціально розроблялася для процесорів з архітектурою ARM. Особливості ядра.

RISC OS дозволяють системі проводити прискорений запуск рахунків зберігання даних у ПЗУ. Такий підхід також допомагає захистити дані при різноманітних збоях та впливу шкідливого ПЗ.

OpenELEC – це програмний комплекс для організації домашнього кінотеатру під керуванням GNU/Linux.

OSMC – ще один комплекс для реалізації домашнього кінотеатру.

У мережі Інтернет, крім перерахованих операційних систем, можна знайти ще безліч модифікацій для різних призначень. Але оскільки Raspbian є основним середовищем для Raspberry Pi, то надалі спиратимемося саме на неї.

Встановлення Raspbian

					ДП. ДС-91. 07. 000 ПЗ	Ар
						51
Змн.	Арк.	№ докум.	Підпи	Дата		

Для встановлення операційної системи необхідно підключити до плати Raspberry Pi мінімальний набір периферії, а саме: монітор, клавіатуру та мишу. Далі, необхідна SD-карта із записаним чином Raspbian. Саме з неї і буде провадитися установка.

Щоб записати образ на карту пам'яті, її необхідно вставити в комп'ютер і відформатувати в системі FAT32. Зробити це можна як стандартними засобами Windows, і сторонніми програмами, наприклад – SD Memory Card Formatter. Після чого завантажуюмо дистрибутив операційної системи з офіційного сайту Raspberry. Для недосвідчених користувачів доступна спрощена версія установника NOOBS. Після завантаження архів необхідно розпакувати в корінь карти пам'яті. На цьому підготовчий етап закінчено.

Вставляємо картку пам'яті в плату Raspberry Pi (клавіатура, миша та монітор вже підключені) та подаємо живлення через роз'єм micro-USB. Починається встановлення Raspbian, яке триває близько 10 хвилин. У цей час від користувача практично нічого не потрібно, крім найпростіших та інтуїтивно зрозумілих дій, таких як вибір мови, введення пароля тощо. На завершальному етапі з'явиться меню, в якому можна вибрати тип інтерфейсу користувача (консольний або графічний). Вибираємо графічний та завершуємо встановлення натисканням кнопки Finish. Система попросить перезавантажитись і як наслідок запуститися вже у більш привабливому вигляді.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						52
Змн.	Арк.	№ докум.	Підпи	Дата		

4. АПРОБАЦІЯ ДІЮЧОЇ МОДЕЛІ СИСТЕМИ КЕРУВАННЯ ЛЮДИНОПОДІБНИМ РОБОТОМ

4.1 Розрахунок енергоефективності системи

Живлення мікроконтролеру

Найпростіший спосіб живлення Pico - це підключити його до роз'єму micro-USB, що живить VSYS (і тим самим систему) від 5-вольтового напруги USB VBUS через D1 (таким чином, VSYS стає VBUS мінус напруга прямого провідника Шотткі).

Якщо порт USB є єдиним джерелом живлення, VSYS і VBUS можна безпечно замкнути разом, щоб усунути падіння напруги на діоді Шотткі (що покращує ефективність та зменшує коливання на VSYS).

Якщо порт USB не буде використовуватися, безпечно живити Pico, з'єднавши VSYS з вашим обраним джерелом живлення (у діапазоні приблизно від 1.8V до 5.5V).

Найпростіший спосіб безпечного додавання другого джерела живлення до Pico полягає в підключенні його до VSYS за допомогою іншого діода Шотткі (рис.4.1). Це дозволить "OR" дві напруги, дозволяючи використовувати в якості живлення VSYS вищу з двох напруг - зовнішню або VBUS, при цьому діоди запобігатимуть зворотному живленню одного джерела живлення в інше. Наприклад, одна літій-іонна акумуляторна батарея* (напруга батареї ~3,0 V до 4,2 V) або 3 батарейки AA в серії (~3,0 V до ~4,8 V) та будь-яке інше джерело живлення з фіксованою напругою в діапазоні від ~2,3 V до 5,5 V будуть працювати добре. Однак, недоліком цього підходу є те, що друге джерело живлення також матиме падіння напруги на діоді, так само як і VBUS, і це може бути небажаним з точки зору ефективності, або якщо напруга вже наближається до нижнього діапазону вхідної напруги, дозволеної для RT6150.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						52
Змн.	Арк.	№ докум.	Підпи	Дата		

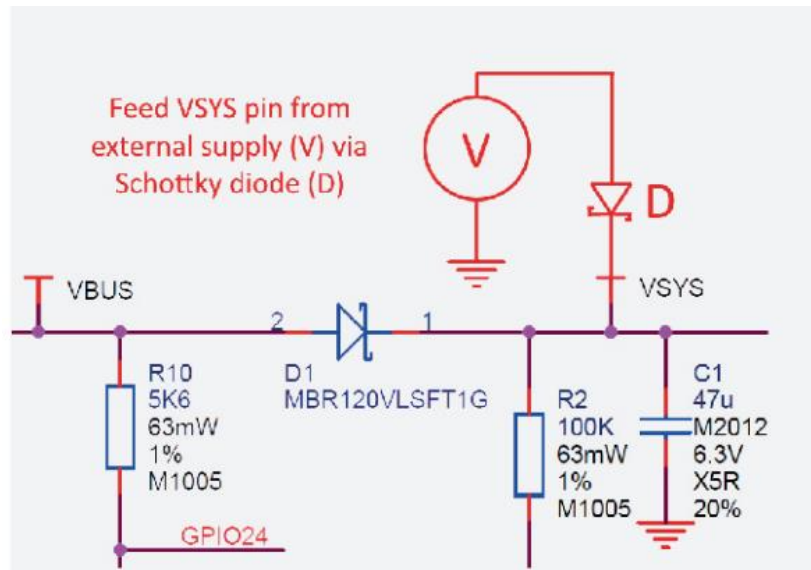


Рис.4.1 Живлення плати діодами

Акумуляторна батарея є перезаряджуваним джерелом електричного струму. Її особливість полягає у можливості циклічного використання: після заряджання батарею можна використовувати впродовж заданої кількості годин, визначеної виробником. Розряджену батарею легко можна зарядити за допомогою спеціального зарядного пристрою. Час заряду залежить від характеристик джерела, а кількість циклів "заряд-розряд" безпосередньо залежить від типу і особливостей акумуляторної батареї.

Емкість акумуляторної батареї відображає тривалість, протягом якої батарея забезпечує повноцінне живлення відповідного пристрою та його автономну роботу при заданому навантаженні. Іншими словами, емкість АКБ визначає максимальну кількість електричної енергії, яку батарея може зберегти під час повного циклу заряду. Ця характеристика впливає на вартість пристрою, його область застосування та термін служби.

Автономність (t) робота визначається як час, протягом якого робот може працювати від свого джерела живлення без підзарядки. Щоб розрахувати автономність, ми використовуємо наступну формулу:

$$t = C / P, \quad (4.1)$$

де t - автономність (у годинах), C - ємність акумулятора (у Ампер-годинах), а P - потужність живлення (у Амперах).

У нашому випадку ємність акумулятора складає 20 Ампер-годин, що означає, що акумулятор може постачати струм в розмірі 2 Ампера протягом 20 годин без підзарядки.

Підставляючи відповідні значення у формулу 4.1:

$$t = 20 \text{ Ah} / 2.1 \text{ A} \approx 9.52.$$

Розрахунок демонструє, що величина струму (2 Ампера) використовується як потужність живлення, а не вати. Це важливо враховувати при обчисленнях.

Отже, після обчислення ми отримуємо параметри автономності більш, ніж 9 годину.

Це означає, що при таких параметрах, робот може працювати без підзарядки протягом приблизно 9.5 годин, використовуючи струм споживання 2.1 Ампера та акумулятор ємністю 20 Ампер-годин. Звичайно, це лише орієнтовне значення, оскільки фактична автономність може

4.2 Підбір акумуляторів для забезпечення автономності системи

Порівняння джерел живлення

Блоки живлення змінного струму можуть бути двох типів: нерегульовані і регульовані. Нерегульовані джерела живлення є основним типом і не можуть постачати постійну напругу на навантаження, у порівнянні з регульованими джерелами живлення, які мають багато різних конструкцій.

Лінійні перетворювачі є простішими, але вони створюють більше тепла, тоді як комутовані перетворювачі більш складні і мають кращу ефективність охолодження, але можуть створювати більше шуму. Акумулятори, як правило, використовують комутовані перетворювачі. Кожен з цих типів має свої

					ДП. ДС-91. 07. 000 ПЗ	Ар
						55
Змн.	Арк.	№ докум.	Підпи	Дата		

переваги та недоліки, і вибір буде залежати від конкретної програми та умов її використання.

Табл. 4.1 ілюструє, як класифікуються типи блоків живлення та підсумовує багато плюсів і мінусів кожного типу.

Таблиця 4.1

Порівняння типів джерел живлення

Лінійна		Батарея	
переваги	недостатки	переваги	недостатки
Безпечно і надійно	Низька ефективність	Портативні	Вхідна напруга фіксована
Стабільність роботи	Габаритний	Невелика вартість	Вихідна напруга падає в міру використання запасів енергії
Регулювання навантаження	Дорогий	Не потребує живлення на місці	Обмежений ресурс використання

Вибір блоку живлення

При виборі блоку живлення є кілька вимог, які необхідно враховувати.

Вимоги до потужності навантаження або ланцюга, включають:

1. Напруги ;
2. Струм;
3. Функції безпеки, такі як обмеження напруги та струму для захисту ;
4. Навантаження;
5. Фізичні розміри і працездатність;
6. Перешкодозахищеність системи.

Хоча всі специфікації джерела живлення є важливими, деякі з них мають більший ступінь значимості. Ось кілька специфікацій, на які варто звернути увагу:

1. Вихідний струм: Це максимальний струм, який може бути поданий на навантаження. Важливо визначити, чи вистачить цього струму для живлення вашого пристрою.

2. Регулювання навантаження: Це вимірювання, яке показує, наскільки добре джерело живлення підтримує постійну напругу при зміні струму навантаження. Воно вимірюється у мілівольтах (mV) або як максимальна вихідна напруга. Чим менше змін в напрузі, тим краще.

3. Шум і пульсація: Шум вказує на додаткові електронні перешкоди, а пульсація вимірює невеликі зміни напруги при перетворенні змінного струму в постійний струм. Ці параметри часто об'єднуються в одне вимірювання. Бажано, щоб ці значення були якомога меншими.

4. Захист від перенапруги: Цей захист забезпечує відключення джерела живлення в разі перевищення номінальних значень напруги на виході. Це захищає навантаження від пошкоджень.

5. Захист від перевантаження: Цей захист активується при короткому замиканні або перевищенні допустимого струму. Він призначений для захисту джерела живлення та навантаження від пошкоджень.

6. Ефективність: Коефіцієнт корисної дії (ККД) відображає ефективність перетворення електричної потужності в постійний струм. Бажано, щоб ККД був якомога вищим, оскільки це означає менше генерації тепла і більшу економію енергії.

Отже, виходячи з отриманих даних можна обрати портативну батарею ємністю від 10Аг до 20Аг, чого буде більш ніж достатньо враховуючи технічне завдання

					ДП. ДС-91. 07. 000 ПЗ	Ар
						52
Змн.	Арк.	№ докум.	Підпи	Дата		

ВИСНОВКИ

У дипломному проєкті був проведений огляд літератури на тему аналізу засобів керування об'єктами, а також розглянуто різні типи систем керування роботами. Після цього було розроблено принципову електричну схему, що детально відображає принцип роботи системи керування. Для докладнішого опису компонентів та їх взаємозв'язків була розроблена структурна електрична схема. На основі отриманих даних було проведено аналіз різних комплектуючих, щоб вибрати найкращі з них для системи.

Далі був розроблений алгоритм роботи для людино-подібного робота, який описує послідовність дій, необхідних для виконання конкретних завдань. Цей алгоритм детально розкриває функції та режими роботи робота.

Для ефективного керування людино-подібним роботом було розроблене програмне забезпечення. Воно базується на отриманих даних та алгоритмі роботи, і дозволяє забезпечити потрібну функціональність та взаємодію з роботом.

В результаті було створено та випробувано систему керування корпусом людино-подібного робота з вражаючими характеристиками. Ця система має високу автономність, здатність працювати у різних сценаріях, включаючи важкі погодні умови, і забезпечує стабільну роботу робота.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						58
Змн.	Арк.	№ докум.	Підпи	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ying Bai and Zvi S. Roth. Classical and Modern Controls with Microcontrollers. In: Springer (2019).
2. Norman S. Nise. Control Systems Engineering. In: (2013).
3. Nadim Maluf and Kirt Williams. An Introduction to Microelectromechanical Systems Engineering. In: Artech House, Inc (2004).
4. Hanheide, M.; Gbelbecker, M.; Horn, G.S.; Pronobis, A.; Sj, K.; Aydemir, A.; Jensfelt, P.; Gretton, C.; Dearden, R.; Janicek, M. Robot task planning and explanation in open and uncertain worlds. Artif. Intell. 2017, 247, 119–150.
5. Sepehr Naimi Muhammad Ali Mazidi Sarmad Naimi. The AVR microcontroller and embedded system using assembly and c. In: Pearson (2009).
6. Yui May L. Chang and James Y. Shih. Microprocessor Applications and Building Control Systems to Achieve Energy Conservation. In: NATIONAL BUREAU OF STANDARDS (1980)
7. Приступа С. О. Мікропроцесори та ЕОМ [Електронний ресурс] / Станіслав Олексійович Приступа // Луцьк. – 2018. – Режим доступу до ресурсу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/16%2C05/other/tema_10_mikr_ontroleri.pdf.
8. Joe Pardue. C programming for Microcontrollers. In: Published by Smiley Micros (2005).
9. Koovappady P O. Lab manual Embedded Systems AVR.
10. Aravind E Vijayan. A beginners Guide to AVR. In: Technical Report (2014).
11. Programming with AVR Microcontroller. In: Research Design Lab (2014).
12. Gokaraju Rangaraju. Microcontrollers Laboratory , work book. In: Gokaraju Rangaraju Institute of Engineering and Technology (1997).
13. Panayotis Papazoglou. An Educational Guide to the AVR Microcontroller Programming. In: Kessariani, Athens, Greece (2018).

					<i>ЛП. ЛС-91. 07. 000 ПЗ</i>	Ар
						59
Змн.	Арк.	№ докум.	Підпи	Дата		

14. Shinsuke Hara. Digital System Design Use of Microcontrolle. In: River Publishess (2010).

15. Tellez, R. "A History of ROS (Robot Operating System)." The Construct (2019).

16. O'Kane, Jason M. "A gentle introduction to ROS." (2014).

17. Gill, J. S. "Setup and Configuration of the Navigation Stack on a Robot." (2018).

18. Marder-Eppstein, E., et al. "Move_Base—Ros Wiki." [online] Available: http://wiki.roseorg/move_base (2016).

19. Котвицкий, Р. С., Александр Васильевич Збруцкий, and Анна Владимировна Сарибога. "Повышение точности инерциальной навигационной системы средствами технического зрения." Механіка гіроскопічних систем 35 (2018): 12-18.

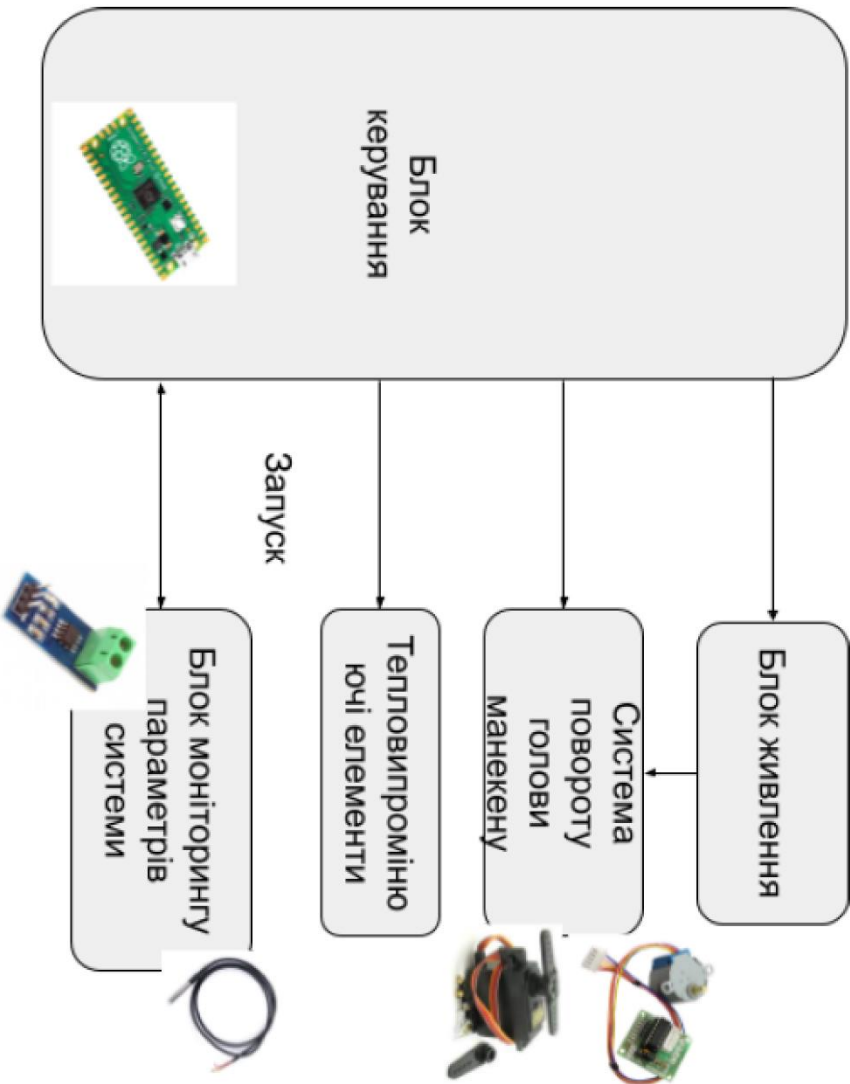
20. Thomas, Dirk. "Changes between ROS 1 and ROS 2." Retrieved on April 17 (2018).

21. Янчевський, Ігор, et al. "АНАЛІЗ ОСНОВНИХ ТИПОРОЗМІРІВ ТА КОНСТРУКЦІЙ НАЗЕМНИХ РОБОТИЗОВАНИХ КОМПЛЕКСІВ." Молодий вчений 5 (69) (2019): 7-10.

					ДП. ДС-91. 07. 000 ПЗ	Ар
						68
Змн.	Арк.	№ докум.	Підпи	Дата		

ДОДАТОК А

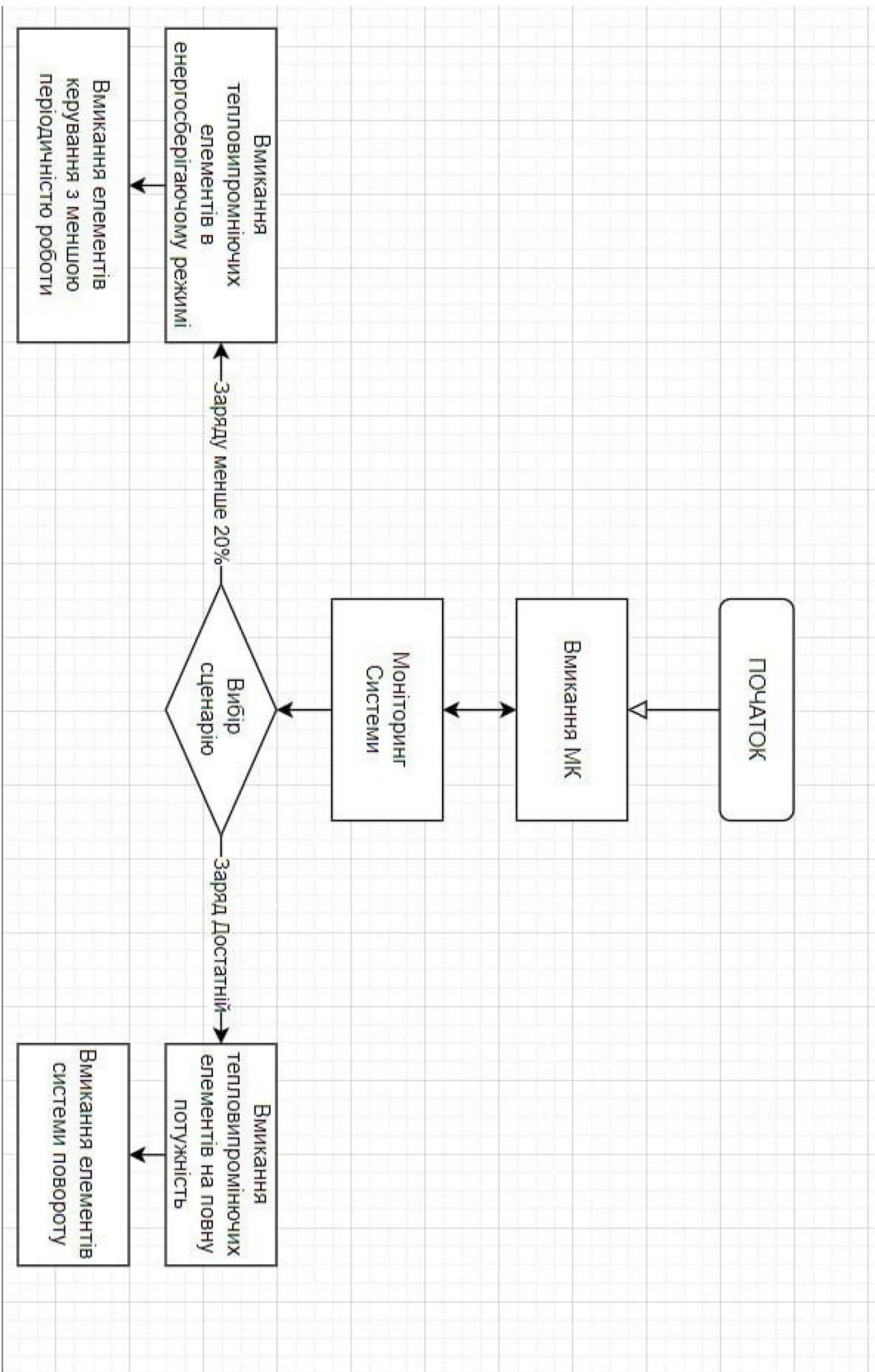
Позна- чення		Найменування			Кі ль к.	Примітки					
		Мікроконтролер									
U1		Raspberry Pi Pico H			1						
		Термометр									
U2		DS18B20-1			1						
		Контролер струму									
U3		ACS712 5A			1						
		Серводвигун									
M1		MG90S			1						
		Нагрівачий елемент									
U4		Heatpanel			1						
		Джерело живлення									
V1		Павербанк 20Аг			1						
					ДП. ДС-91. 006. 000ПЕ						
Із м	Лис	№ докум.	Підпис	Дата							
Розроб.	Андрій КОРЖ				Пристрій узгодження комплексу командних приладів і комп'ютера. Перелік елементів			Літ.	Лист	Листів	
Перевір	Володимир РОМАШко								1	2	
								КПІ ім. Ігоря Сікорського, ФЕЛ, ЕПС, гр. ДС-91			
Н.контр	Лариса БАТРАК										
Затверд	Володимир РОМАШКО										



Рівень 1	Рівень 2	Рівень 3	Рівень 4	Рівень 5

Рівень 6	Рівень 7

ДП ДС-91.006.000 Е1									
Головний конструктор: [підпис]									
Спеціаліст: [підпис]									
Інженер: [підпис]									
ФЕЛ ДС-91									



ЛІСТИНГ ПРОГРАМИ

Програмний код для електронної системи моніторингу

```
import logging
```

```
import time
```

```
import math
```

```
# Based on Adafruit Lib:
```

```
#
```

```
https://github.com/adafruit/Adafruit\_Python\_PCA9685/blob/master/Adafruit\_PCA9685/PCA9685.py
```

```
#          Default      address:  
PCA9685_ADDRESS = 0x40
```

```
#          Registers/etc:  
MODE1 = 0x00
```

```
MODE2 = 0x01
```

```
SUBADR1 = 0x02
```

```
SUBADR2 = 0x03
```

```
SUBADR3 = 0x04
```

```
PRESCALE = 0xFE
```

```
LED0_ON_L = 0x06
```

```
LED0_ON_H = 0x07
```

```
LED0_OFF_L = 0x08
```

```
LED0_OFF_H = 0x09
```

```
ALL_LED_ON_L = 0xFA
```

```
ALL_LED_ON_H = 0xFB
```

```
ALL_LED_OFF_L = 0xFC
```

```
ALL_LED_OFF_H = 0xFD
```

```
# Bits:
```

```
    RESTART      = 0x80
    SLEEP         = 0x10
    ALLCALL       = 0x01
    INVRT         = 0x10
    OUTDRV        = 0x04
```

Channels

```
    CHANNEL00    = 0x00
    CHANNEL01    = 0x01
    CHANNEL02    = 0x02
    CHANNEL03    = 0x03
    CHANNEL04    = 0x04
    CHANNEL05    = 0x05
    CHANNEL06    = 0x06
    CHANNEL07    = 0x07
    CHANNEL08    = 0x08
    CHANNEL09    = 0x09
    CHANNEL10    = 0x0A
    CHANNEL11    = 0x0B
    CHANNEL12    = 0x0C
    CHANNEL13    = 0x0D
    CHANNEL14    = 0x0E
    CHANNEL15    = 0x0F
```

```
class PCA9685(object):
```

```
    def __init__(self, i2cBus, address=PCA9685_ADDRESS):
```

```
        self.i2cBus = i2cBus
```

```
        self.address =
```

```
        addressself.begin()
```

```
    def begin(self): """Initialize
```

```
        device"""
```

```
        self.set_all_pwm(0, 0)
```

```
            self.i2cBus.write_byte_data(self.address, MODE2, OUTDRV)
```

```
        self.i2cBus.write_byte_data(self.address, MODE1, ALLCALL)
```

```
        time.sleep(0.005) # wait for oscillator
```

```
        mode1 = self.i2cBus.read_byte_data(self.address, MODE1)
```

```

        mode1 = mode1 & ~SLEEP                                # wake up (reset sleep)
self.i2cBus.write_byte_data(self.address, MODE1, mode1) time.sleep(0.005) #
wait for oscillator

```

def reset(self):

```

        self.i2cBus.write_byte_data(self.address,  MODE1,  RESTART)
time.sleep(0.01)

```

def set_address(self, address):

```

    """Sets device address."""
    self.address = address

```

def set_i2c_bus(self, i2cBus):

```

    """Sets I2C Bus."""

```

```

        self.i2cBus = i2cBus

```

def set_pwm(self, channel, on, off): """Sets
a single PWM channel."""

```

self.i2cBus.write_byte_data(self.address, LED0_ON_L + 4 * channel, on & 0xFF)

```

```

        self.i2cBus.write_byte_data(self.address, LED0_ON_H + 4 * channel, on >> 8)

```

```

        self.i2cBus.write_byte_data(self.address, LED0_OFF_L + 4 * channel, off &
0xFF)

```

```

        self.i2cBus.write_byte_data(self.address, LED0_OFF_H + 4 * channel, off >>
8)

```

def set_all_pwm(self, on, off):

```

    """Sets all PWM channels."""

```

```

        self.i2cBus.write_byte_data(self.address,  ALL_LED_ON_L,  on  &  0xFF)

```

```

        self.i2cBus.write_byte_data(self.address,  ALL_LED_ON_H,  on  >>  8)

```

```

        self.i2cBus.write_byte_data(self.address,  ALL_LED_OFF_L,  off  &  0xFF)

```

```

        self.i2cBus.write_byte_data(self.address, ALL_LED_OFF_H, off >> 8)

```

def set_pwm_freq(self, freq_hz):

```

    """Set the PWM frequency to the provided value in hertz."""

```

```

    prescaleval = 25000000.0                                # 25MHz

```

```

    prescaleval /= 4096.0                                    # 12-bit prescaleval

```

```

    /= float(freq_hz)

```

```

    prescaleval -= 1.0

```

```

prescale = int(math.floor(prescaleval + 0.5))
oldmode = self.i2cBus.read_byte_data(self.address, MODE1)
newmode = (oldmode & 0x7F) | 0x10 # sleep
self.i2cBus.write_byte_data(self.address, MODE1, newmode) # go to sleep
self.i2cBus.write_byte_data(self.address, PRESCALE, prescale)
self.i2cBus.write_byte_data(self.address, MODE1, oldmode) time.sleep(0.005)

self.i2cBus.write_byte_data(self.address, MODE1, oldmode | 0x80)

```

```
def __enter__(self):
```

```
    return self
```

```
def __exit__(self, exception_type, exception_value, traceback):
```

```
    self.reset()
```

Опис методів (функцій)

__init__()

Конструктор класу. **__init__**(self, i2cBus, address=PCA9685_ADDRESS)

Параметри

i2cBus — Об'єкт типа PCA9685.
address — I2C адрес устройства. За замовчуванням PCA9685_ADDRESS = 0x40.*begin()*

Ініціалізація устройства.

begin(self)

set_address()

Установка адреси устройства.

set_address(self, address)

Параметри

address — I2C адрес устройства.

set_i2c_bus()

Установка I2C шини.

set_i2c_bus(self, i2cBus)

Параметри

i2cBus — Об'єкт типу PCA9685.

set_pwm()

Встановлює ШІМ одного із виводів PCA9685.

```
set_pwm(self, channel, on, off)
```

Параметри

channel — Один из выводов PWM от 0 до 15.

on — В який момент цикла из 4096 частей ввімкнути ШИМ. off —

В який момент цикла из 4096 частей вимкнути ШИМ.set_all_pwm()

Встановлює ШИМ на всі виводи PCA9685.

```
set_all_pwm(self, on, off)
```

Параметри

on — У який момент циклу з 4096 частин включити ШИМ. off —

В який момент цикла из 4096 частей вимкнути ШИМ.set_pwm_freq()

Встановлює частоту ШИМ для всего чипа, до $\sim 1,6$ кГц.

```
set_pwm_freq(self, freq_hz)
```

Параметри

freq_hz — Частота в Герцах.

3.Дослідження стійкості замкненої дискретної системи за допомогою аналогу критерію стійкості Михайлова або аналогу критерію стійкості Найквіста в середовищі Матлаб.

```
clear;
```

```
clc;
```

```
% Передавальна функція розімкненої системи
```

```
W=tf(20,[5 10 1])
```

```
% Передавальна функція розімкненої системи
```

```
Wclose=feedback(W,1)
```

```
figure(1);
```

```
step(Wclose);
```

```
hold on;
```

```
% Перетворення безперервної функції у дискретну
```

```
WDclose=c2d(Wclose,0.2)
```

```
step(WDclose);
```

```
hold off;
```

```
grid;
```

```
figure(2);
```

```
hold on;
```

```
% Побудова годографа Михайлова
```

```
for w=0:0.01:pi
```

```

A=exp(j*w*2)-1.534*exp(j*w)+0.6703;
P=real(A);
Q=imag(A);
plot(P,Q,'k. ');
end hold
off;grid;
title('Годограф Михайлова для дискретної системи');
WDopen=c2d(W,0.2);
figure(3);
% Побудова годографа Найквіста
nyquist(WDopen);
grid;

```

$$z^2 - 1.534 z + 0.6703$$

Sample time: 0.2 seconds

Discrete-time transfer function.

S U M M A R Y

Humanoid robot body control system

The diploma project of first educational level "Bachelor" by specialty 171 Electronics, specialization Electronic devices and systems. Korzh Andrii Andriyivich. National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute». Faculty of Electronics, Department of Electronic Devices and Systems. Academic group DS-91. - Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2019. - 2023. pages 60, tables 3, figures 31.

In the diploma project, a literature review was conducted on the topic of object control analysis and various types of robot control systems were considered. An electrical schematic diagram and a structural electrical diagram were developed, and based on the obtained data, an analysis and selection of components were carried out. An algorithm for the operation of a humanoid robot was created and its execution was described. Based on the collected data, software was developed for the control system of the humanoid robot.

As a result, a control system for the body of a humanoid robot was created and tested, with the following characteristics:

- Autonomy of up to 9 hours.
- Variability of work scenarios.
- Operation in harsh weather conditions.
- Stability of operation.

System Design

The control system consists of subsystems and processes (or tasks) assembled to achieve a desired outcome with the required performance, taking into account the specified output signal. Figure 1.1 illustrates the control system in its simplest form, where the input represents the desired output.

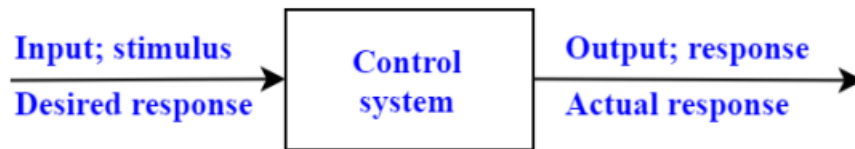


Fig. 1.1 Control System

For example, let's consider an elevator. Pressing the button for the fourth floor serves as an input signal, representing our desired outcome shown as a step function in Figure 1.2. The effectiveness of the elevator can be observed from its response curve shown in the figure.

Two primary performance indicators are evident:

1. Transient response.
2. Steady-state error.

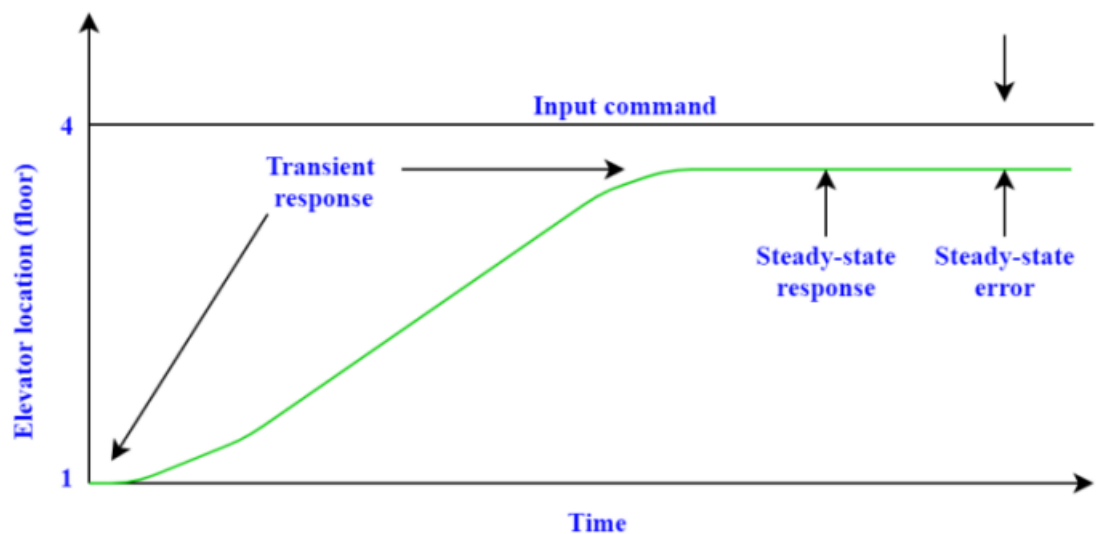


Fig. 1.2 Transient Response

In our example, passenger comfort and patience depend on the transient response. If this response is too fast, passenger comfort is sacrificed; if it's too slow, passenger patience is compromised. Steady-state error is another important specification characteristic, as safety and passenger convenience would be compromised if the elevator fails to properly align.

Analysis is the process by which the system's performance is evaluated. For instance, we assess its transient response and steady-state error to determine if they meet the desired specifications. Design, on the other hand, is the process by which the system's performance is created or modified. For example, if the analyzed transient response and steady-state error do not meet the specifications, we change parameters or add additional components to align with the specifications.

Transient response is critical. In the case of an elevator, a slow transient response makes passengers impatient, while an excessively fast response causes discomfort. If the elevator oscillates around the arrival position for longer than a second, it can lead to confusion. Transient response is also important for structural reasons: an excessively fast transient response can cause permanent physical damage. Another goal of analysis and design is focused on steady-state response. As we have seen, this response resembles the input and is typically what remains after the transient processes have decayed to zero. We are concerned with the accuracy of the steady-state response. We quantitatively determine steady-state errors, analyze system steady-state errors, and then develop corrective action to reduce steady-state errors—our second goal of analysis and design.

Discussions of transient response and steady-state error are contentious if the system lacks stability. Control systems must be designed to be stable. That is, their inherent response must decay to zero as time approaches infinity or oscillate. In many systems, the transient response you see on the time response plot can be directly related to the inherent response. Thus, if the inherent response decays to zero as time approaches zero, the transient response will also decay, leaving only the forced response. If the system is stable, we can design proper transient response and steady-state error characteristics. Stability is our third goal of analysis and design.

During the project, several options of microprocessors and microcontrollers were considered.

A microprocessor is an electronic component used by a computer to perform its tasks. It is a central processing unit on a single integrated circuit that contains

millions of very small components, including transistors, resistors, and diodes, working together. A microprocessor is a versatile digital integrated circuit based on registers, clock-driven, which takes binary data as input, processes them according to instructions stored in its memory, and provides results (also in binary form) as output data. Microprocessors contain both combinational and sequential digital logic. Microprocessors operate on numbers and characters represented in the binary number system.

Power Supply for Microcontroller

The simplest way to power the Pico microcontroller is to connect it to the micro-USB connector, which powers the VSYS (and therefore the system) from the 5-volt USB VBUS voltage through D1 (thus, VSYS becomes VBUS minus the voltage drop across the Schottky diode).

If the USB port is the only power source, VSYS and VBUS can be safely shorted together to eliminate voltage drop across the Schottky diode (improving efficiency and reducing fluctuations on VSYS).

If the USB port will not be used, you can safely power the Pico by connecting VSYS to your chosen power source (ranging from approximately 1.8V to 5.5V).

The simplest way to safely add a second power source to the Pico is to connect it to VSYS using another Schottky diode (Figure 4.1). This will allow "OR"ing of the two voltages, allowing you to use the higher of the two voltages as the power source for VSYS - either the external source or VBUS, while the diodes prevent reverse powering from one power source to the other. For example, a single lithium-ion battery* (battery voltage ~3.0V to 4.2V) or 3 AA batteries in series (~3.0V to ~4.8V) and any other fixed voltage power source in the range of approximately 2.3V to 5.5V will work well. However, a drawback of this approach is that the second power source will also have a voltage drop across the diode, similar to VBUS, and this may be undesirable in terms of efficiency, or if the voltage is already approaching the lower allowable input voltage range for the RT6150.

A rechargeable battery is a source of electrical power that can be cyclically used. After charging, the battery can be used for a specified number of hours determined by the manufacturer. A discharged battery can easily be recharged using a dedicated charging device. The charging time depends on the characteristics of the power source, and the number of charge-discharge cycles depends directly on the type and features of the rechargeable battery.

The capacity of a rechargeable battery reflects the duration for which the battery can provide full power to the corresponding device and its autonomous operation under a given load. In other words, the battery capacity determines the maximum amount of electrical energy the battery can store during a full charge cycle. This characteristic affects the cost of the device, its application range, and its service life.

Autonomy (t) is defined as the time during which a device can operate from its power source without recharging. To calculate autonomy, we use the following formula:

$$t = C / P$$

where t is the autonomy (in hours), C is the battery capacity (in ampere-hours), and P is the power consumption (in amperes).

In our case, the battery capacity is 20 ampere-hours, which means the battery can supply a current of 2 amperes for 20 hours without recharging.

$$t = 20 \text{ Ah} / 2 \text{ A} = 10 \text{ hours}$$

The calculation demonstrates that the current value (2 amperes) is used as power consumption, not watts. It is important to consider this when performing calculations.

Therefore, after calculation, we obtain an autonomy parameter of more than 10 hours. This means that under these parameters, the device can operate without recharging for approximately 10 hours, using a power consumption of 2.1 amperes

and a battery capacity of 20 ampere-hours. Of course, this is only an approximate value as the actual autonomy may vary depending on various factors.

The diploma project included a literature review on the topic of object control analysis and a discussion of various types of robot control systems. Following that, a conceptual electrical circuit was developed, providing a detailed representation of the control system's operation. To provide a more detailed description of the components and their interconnections, a structural electrical diagram was created. Based on the obtained data, an analysis of different components was conducted to select the best ones for the system.

Subsequently, a working algorithm for a humanoid robot was developed, describing the sequence of actions required to perform specific tasks. This algorithm elaborates on the functions and operating modes of the robot in detail.

To enable efficient control of the humanoid robot, software was developed. It is based on the gathered data and the robot's operating algorithm, providing the necessary functionality and interaction with the robot.

As a result, a control system for the humanoid robot was created and tested, exhibiting impressive characteristics. This system demonstrates high autonomy and the ability to operate in various scenarios, including challenging weather conditions, while ensuring stable robot performance.

Keywords: control system, autonomous control system, humanoid robot, microcontrollers

Summary of the project: 6 pages