

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНЖЕНЕРІЇ ТА ТРАНСПОРТУ
КАФЕДРА АВТОМАТИЗАЦІЇ, РОБОТОТЕХНІКИ І МЕХАТРОНІКИ

Кваліфікаційна робота бакалавра

на тему: «Програмно-апаратний (електронний) комплекс вимірювання кутів з
використанням акселерометра і гіроскопу в сфері будівництва»
« Software and hardware (electronic) complex for measuring angles using an
accelerometer and gyroscope in construction »

Виконав: студент 4 курсу, групи 4А
спеціальність 151 – «Автоматизація та
комп'ютерно-інтегровані технології»
Деркаченко Т. О.

Керівник: к.т.н., доцент Сарафаннікова Н.В.

Рецензент _____
(прізвище та ініціали)

Херсон – 2024 рік

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ІНЖЕНЕРІЇ ТА ТРАНСПОРТУ

КАФЕДРА АВТОМАТИЗАЦІЇ, РОБОТОТЕХНІКИ І МЕХАТРОНИКИ

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Програмно-апаратний (електронний) комплекс вимірювання кутів з

використанням акселерометра і гіроскопу в сфері будівництва»

« Software and hardware (electronic) complex for measuring angles using an
accelerometer and gyroscope in construction »

Виконав: студент 4 курсу, групи 4А
спеціальність 151 – «Автоматизація та
комп'ютерно-інтегровані технології»

Деркаченко Т.О.

Керівник: к.т.н., доцент Сарафаннікова Н.В.

Рецензент _____

(прізвище та ініціали)

Херсон – 2024 рік

	Херсонський національний технічний університет
Факультет	Інженерії та транспорту
Кафедра	Автоматизації, робототехніки і мехатроніки
Ступінь вищої освіти	бакалавр
Спеціальність	151 – «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ:
Завідувач кафедри автоматизації,
робототехніки і мехатроніки
Дмитрієв Д.О.
 «__» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Деркаченко Тимофію Олеговичу

1. Тема проекту: Програмно-апаратний (електронний) комплекс вимірювання кутів з використанням акселерометра і гіроскопу в сфері будівництва

Software and hardware (electronic) complex for measuring angles using an accelerometer and gyroscope in construction

керівник проекту: к.т.н., доцент Сарафаннікова Н.В.

затверджена наказом вищого навчального закладу від 15.01.2024 р. № 20-с

2. Строк подання студентом проекту «10» червня 2024 р.

3. Вихідні дані до проекту: Довідникова література

4. Зміст розрахунково-пояснювальної записки: 1. ЗАГАЛЬНИЙ ОГЛЯД ТЕОДОЛІТУ ТА ЙОГО ХАРАКТЕРИСТИК; 2. МАТЕМАТИЧНА ЧАСТИНА. ОТРИМАННЯ КУТА ОБЕРТУ. КВАТЕРНІОНИ; 3. АПАРАТНА ЧАСТИНА ПАК

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): 1. Структурна схема ПАК, 2. Структурна схема ПАК, 3. Функціональна схема ПАК, 4. Принципова схема

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Сарафаннікова Н.В., к.т.н., доцент		

7. Дата видачі завдання

«12» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломоно проекту	Строк виконання етапів проекту	Примітка
1	Підбір і огляд літератури	12.02.24 – 31.03.24	
2	Аналіз існуючих рішень	01.04.24 - 14.04.24	
3	Розробка математичної моделі	15.04.24 - 30.04.24	
4	Оформлення пояснювальної записки	01.05.24 - 15.05.24	
5			
6			
7			

Студент

Деркаченко Т. О.

(підпис)

Керівник проекту

Сарафаннікова Н.В.

(підпис)

[illegible]

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: «Програмно-апаратний (електронний) комплекс вимірювання кутів з використанням акселерометра і гіроскопу в сфері будівництва»

Область застосувань: вимірювання кутів у сфері будівництва

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки «Програмно-апаратного (електронний) комплекс вимірювання кутів з використанням акселерометра і гіроскопу в сфері будівництва» є завдання на дипломне проектування, затверджене кафедрою Херсонського національного технічного університету.

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою дослідження є розробка програмно-апаратного комплексу, який може вимірювати кути з використанням акселерометра та гіроскопу.

Призначення даної розробки в сторені дешевого та сучасного пристрою, який надасть можливість автоматизувати вимірювання кутів.

4. ДЖЕРЕЛО РОЗРОБКИ

Джерелом розробки є науково технічна література з технічних питань та статі в інтернеті за даною темою.

					ХНТУ 151.КРБ.24.022 ТЗ							
Зм.	Арк.	№ докум	Підпис	Дата								
Розроб.	Держаченко Т.О.				Технічне завдання					Лім.	Аркуш	Аркушів
Перевір.	Сарафанікова Н.В.										8	68
Реценз.										Error! Unknown document property name.		
Н. Контр.	Поліщук В.М.											
Затверд.	Error! Unknown											

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розроблюваного продукту

Вимоги до розробки:

- висока точність у вимірюванні кутів
- зручне у сприйнятті виведення значень кутів
- невеликі розміри розробки

					ХНТУ 151.КРБ.24.02 ТЗ	Арк.
						9
Зм.	Арк.	№ докум	Підпис	Дата		

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 68 сторінок, 18 рисунки, 5 таблиці.

Кваліфікаційна робота бакалавра присвячена вдосконаленню методів проведення вимірювань кутів в будівельній сфері, за допомогою акселерометра та гіроскопа.

Проведений огляд теодолітів як методу вимірювання кутів та аналіз існуючих рішень з приводу автоматизації вимірювань та збереження вимірювань. За результатами аналізу було прийнято рішення проведення автоматизації процесу вимірювання кутів за допомогою MEMS акселерометра, гіроскопа та керування обертанням за допомогою сферичного паралельного механізму. Найбільш оптимальною для визначення кутів за допомогою такого апаратного забезпечення була визначенна теорія кватерніонів.

Обрані засоби автоматизації, що задовольняють технічним умовам пред'явленим для оптимальної роботи програмно-апаратного комплексу. До цього апаратного забезпечення було розроблено програмне забезпечення, для проведення, та передачі вимірювань для відображення у необхідній формі, для передачі керуючих впливів на сервоприводи сферичного паралельного механізму.

Це дало змогу здійснювати вимірювання, обертання без втручання людини безпосередньо у прилад.

ВИМІРЮВАННЯ КУТІВ, АКСЕЛЕРОМЕТР, ГІРОСКОП, ПРОГРАМНО
АПАРАТНИЙ КОМПЛЕКС, ARDUINO PRO MICRO LEONARDO,
КВАТЕРНІОНИ, MEMS

					ХНТУ 151.КРБ.24.02 РФ						
Зм.	Арк.	№ докум	Підпис	Дата							
Розроб.		Держаченко Т.О.			Реферат Error! Unknown document property name						
Перевір.		Сарафанікова Н.В.									
Реценз.											
Н. Контр.		Поліщук В.М.									
Затверд.		Error! Unknown									
					Лім.			Аркуш	Аркушіє		
								10	68		

ABSTRACT

Bachelor's thesis: 52 pages, 1 figures, 5 tables.

The bachelor's classification work is devoted to improving the methods of measuring angles in the construction field, using an accelerometer and a gyroscope.

An overview of theodolites as a method of measuring angles and an analysis of existing solutions regarding the automation of measurements and the preservation of measurements was carried out. Based on the results of the analysis, a decision was made to automate the angle measurement process using a MEMS accelerometer, a gyroscope, and control rotation using a spherical parallel mechanism. The most questionable for determining angles with the help of such hardware was the definite theory of quaternions.

Automation tools that meet the technical conditions for optimal operation of the hardware and software complex are selected. To this hardware, software was developed to take and transmit measurements for display in the required form, to transmit control effects to the servo drives of the spherical parallel mechanism.

This made it possible to measure and rotate without human intervention directly in the device.

MEASUREMENT OF ANGLES, ACCELEROMETER, GYROSCOPE,
SOFTWARE AND HARDWARE COMPLEX, ARDUINO PRO MICRO
LEONARDO, QUATERNIONS, MEMS

					<i>XHTY 151.KPB.24.02 PΦ</i>	Арк.
						11
Зм.	Арк.	№ докум	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	14
ВСТУП.....	15
1 ЗАГАЛЬНИЙ ОГЛЯД ТЕОДОЛІТУ ТА ЙОГО ХАРАКТЕРИСТИК.....	16
1.1 Галузі застосувань	16
1.2 Огляд теодоліту	16
1.2.1 Огляд загальної побудови теодоліту	16
1.2.2 Приведення теодоліта в робоче положення	21
1.2.3 Загальні характеристики теодолітів	23
1.2.4 Огляд, перевірки та дослідження теодолітів.....	24
1.3 Вимірювання кутів	25
1.3.1 Вимірювання горизонтальних кутів	25
1.3.2 Вимірювання вертикального кута	27
1.4 Огляд існуючих аналогів.....	29
1.4.1 Оптичні теодоліти.....	29
1.4.2 Електронні теодоліти.....	31
1.5 Приклад автоматизації роботи теодолітів.....	34
1.6 Висновок до першого розділу.....	36
2 МАТЕМАТИЧНА ЧАСТИНА. ОТРИМАННЯ КУТА ОБЕРТУ.	
КВАТЕРНІОНИ.....	37
2.1 Кватерніони	37
2.2 Отримання кута оберту у градусній мірі.....	40
2.3 Інші одиниці вимірювань кутів	41
2.4 Фільтрація шуму та корекція похибок	44
2.4.1 Нормалізація кватерніонів	44
2.5 Обробка та управління об'єктами	45
2.5.1 Генерація кватерніону для обертання	45
2.5.2 Застосування обертання	46
2.6 Розрахунок керуючих впливів до СПМ	46

3 АПАРАТНА ЧАСТИНА ПАК.....	50
3.1 Структурна схема	50
3.1.1 Опис структури та роботи ПАК.....	50
3.2 Функціональна схема	52
3.3 Вибір акселерометра	54
3.4 Вибір контролера.....	59
3.5 СПМ	65
3.6 Програма контролера для вимірювання кутів та керування СПМ	66
3.6.1 Склад бібліотек до програми	66
3.6.2 Алгоритм програми для вимірювання кутів та керування СПМ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПАК – програмно-апаратний комплекс

MEMS – мікроелектромеханічні системи (micro-electro-mechanical systems)

КП – керуючий пристрій

СПМ – сферичний позиціонуючий механізм

					ХНТУ 151.КРБ.24.02 ПЗ	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		14

ВСТУП

Автоматизація проникла у багато сфер життя, і будівництво не стало винятком. Сучасний технологічний прогрес відкриває широкі можливості для автоматизації будівельних процесів, включаючи вимірювання кутів.

Одним з традиційних інструментів для вимірювання кутів є теодоліти – оптичні інструменти, які протягом багатьох років були основним інструментом геодезистів та інженерів. Однак з розвитком електроніки та програмного забезпечення почали з'являтися нові, більш досконалі методи вимірювання. Прикладом таких є технології MEMS (мікроелектромеханічні системи).

MEMS акселерометри та гіроскопи відкривають нові горизонти у вимірюванні кутів та позиціонуванні об'єктів у просторі. Ці мініатюрні датчики завдяки своїм малим розмірам, можуть бути інтегровані у сучасні будівельні інструменти та обладнання, забезпечуючи високу точність і надійність вимірювань.

Завдяки розвитку цих технологій, будівельна галузь отримала потужні інструменти для покращення точності та ефективності вимірювань. Саме ці технології відкривають нові можливості для автоматизації будівельних процесів.

1 ЗАГАЛЬНИЙ ОГЛЯД ТЕОДОЛІТУ ТА ЙОГО ХАРАКТЕРИСТИК

1.1 Галузі застосувань

Основні сфери застосування де теодоліт знайшов себе: астрономія, геодезія, будівництво.

Теодоліт використовується при проведенні: топографічних, геодезичних зйомках; розбивочних досліджень; землепорядних вимірювань; контролю відповідальних будівельно-монтажних операцій. Застосовується він і при вирішенні інших завдань, що вимагають високоточних вимірів положення об'єктів ситуації на місцевості, деталей будівель і інженерних споруд.

Придбати популярність у багатьох сферах теодоліту допомогли компактність і зручність в експлуатації [1].

Найбільше застосування в практиці геодезичної астрономії на сучасному етапі має астрономічний теодоліт. Він застосовується для точних вимірів горизонтальних напрямків і зенітних віддалей світил, а також може застосовуватись для спостереження проходження світил через один і той же альмукантарат і через один і той же вертикал. За допомогою окулярного мікрометра труби теодоліта можна вимірювати малі різниці зенітних віддалей і малі різниці азимутів.

Таким чином, за допомогою астрономічного теодоліта можна виконувати визначення широти, часу і азимута напрямку всіма відомими способами астрономічних визначень [2].

1.2 Огляд теодоліту

1.2.1 Огляд загальної побудови теодоліту

Теодоліт – прилад, призначений для вимірювання горизонтальних і вертикальних кутів, відстаней та кутів орієнтування. Теодоліти, які дозволяють

вимірювати ще й відстані за далекомірними нитками, а також азимути за допомогою накладної бусолі, називають теодолітами-тахеометрами [3].

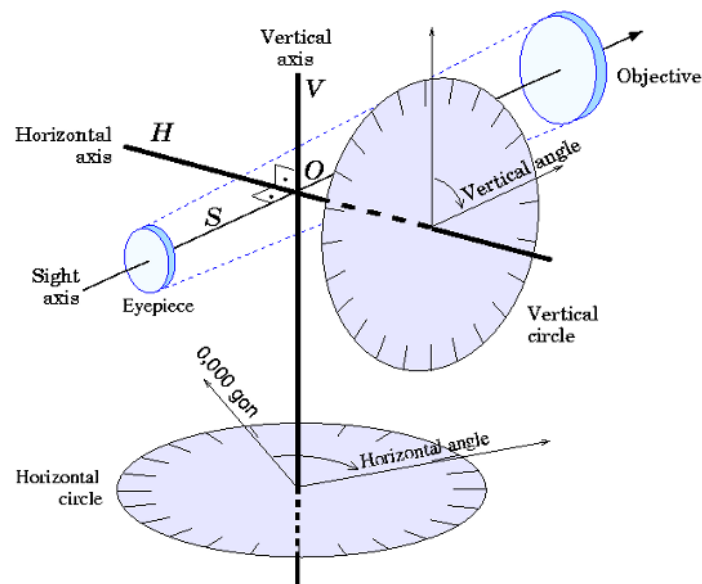


Рис. 1.1. Лімб теодоліту, осі обертання та візирна вісь

Теодоліти класифікують за багатьма критеріями:

- 1) за призначенням: геодезичні, маркшейдерські, астрономічні, фототеодоліти;
- 2) за точністю (відповідно до середньо-квадратичної помилки вимірювання кута): високоточні (Т1, Т05), точні (Т2, Т5) і технічні (Т15, Т30, 2Т30);
- 3) за типом відлікового пристрою: верньєрні, із шкаловим мікроскопом, із штриховим мікроскопом, із оптичним мікрометром, із електронною цифровою індикацією;
- 4) за конструкцією вертикальної осі: повторювальні, неповторювальні або прості;
- 5) за конструкцією оптичної відлікової системи: з двохстороннім відліком за кругами і одностороннім;
- 6) за фізичною природою носія інформації: механічні, оптичні, кодові;
- 7) за типом зорової труби: з прямим і оберненим зображенням;

8) за конструкцією стабілізації відлікового індекса: з рівнем при вертикальному крузі, з рівнем при алідаді горизонтального круга, з компенсатором кутів нахилу.

У шифрі теодоліта літера „Т” означає „теодоліт”, а цифри – середню квадратичну похибку вимірювання горизонтального кута одним прийомом (у секундах). Додаткова літера в шифрі теодоліта означає його модифікацію або конструктивне рішення: „А” – астрономічний, „М” – маркшейдерський, „К” – з компенсатором вертикального круга, „П” – зорова труба з прямим зображенням. Якщо на основі базової моделі розроблена нова модифікація, перед шифром додається цифра „2”.

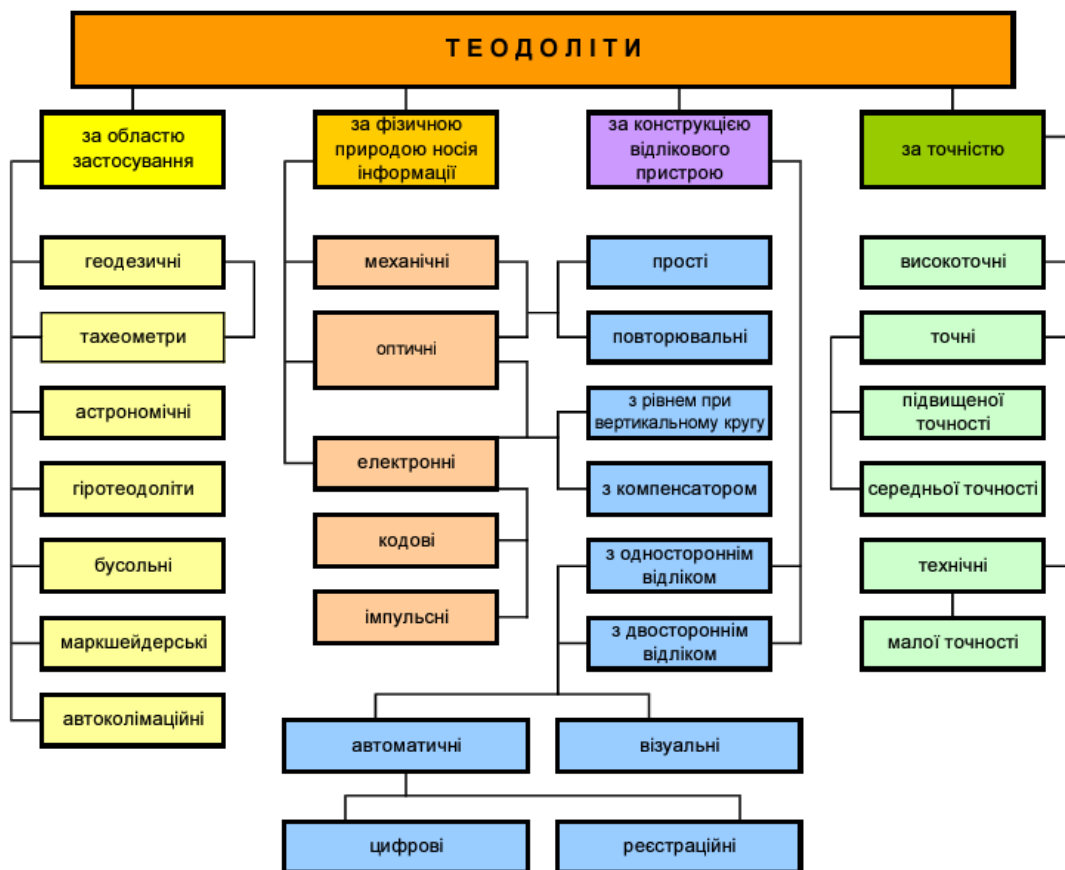


Рис. 1.2. Класифікація теодолітів

Загальний вигляд і будова теодоліта показана на рис. 1.3.

Теодоліти бувають різних конструкцій, мають різний зовнішній вигляд, але назви основних частин у всіх типів теодолітів і їх призначення однакові.

Штатив служить для встановлення приладу над вершиною кута. До верхньої частини (головки) штатива за допомогою станового гвинта (25) прикріплюється теодоліт. На головку штатива спирається основа (1) з трьома піднімальними гвинтами (15) і підставкою (13), яку ще називають трегер. Піднімальні гвинти розташовані через 120° один від одного. Їх призначення – приводити прилад у горизонтальне положення за допомогою циліндричного рівня (5).

Для вимірювання горизонтальних і вертикальних кутів слугують кутомірні круги теодоліта – відповідно горизонтальний (позначається ГК) і вертикальний (ВК) (19). Круги складаються із лімба (ГК – (21), ВК – (23)) і алідади (ГК – (22), ВК – (24)). Лімба – це скляний круг, який розмічений поділками від 0° до 360° за рухом годинникової стрілки. При вимірюванні кута лімба є нерухомим і горизонтальним. Вісь алідади вміщується у вісь лімба. На алідаді нанесено відліковий пристрій у вигляді шкали, за допомогою якої знімається відлік за лімбом.

Над трегером (13) розміщена верхня частина теодоліта, яка називається алідадною. Вона обертається навколо вертикальної осі теодоліта і складається з алідади ГК (22), колонок (12), на яких кріпиться зорова труба (26) та вертикального круга (19). Зорова труба може обертатися навколо своєї осі обертання від 0° до 360° . На одному з кінців осі обертання труби закріплений вертикальний круг. Під час вимірювань вертикальний круг може розміщуватись від зорової труби ліворуч (круг ліворуч – КЛ) або праворуч (круг праворуч – КП). Лімба (23) вертикального круга наглухо скріплений із зоровою трубою і обертається разом з нею, а алідада (24) вертикального круга – нерухома.

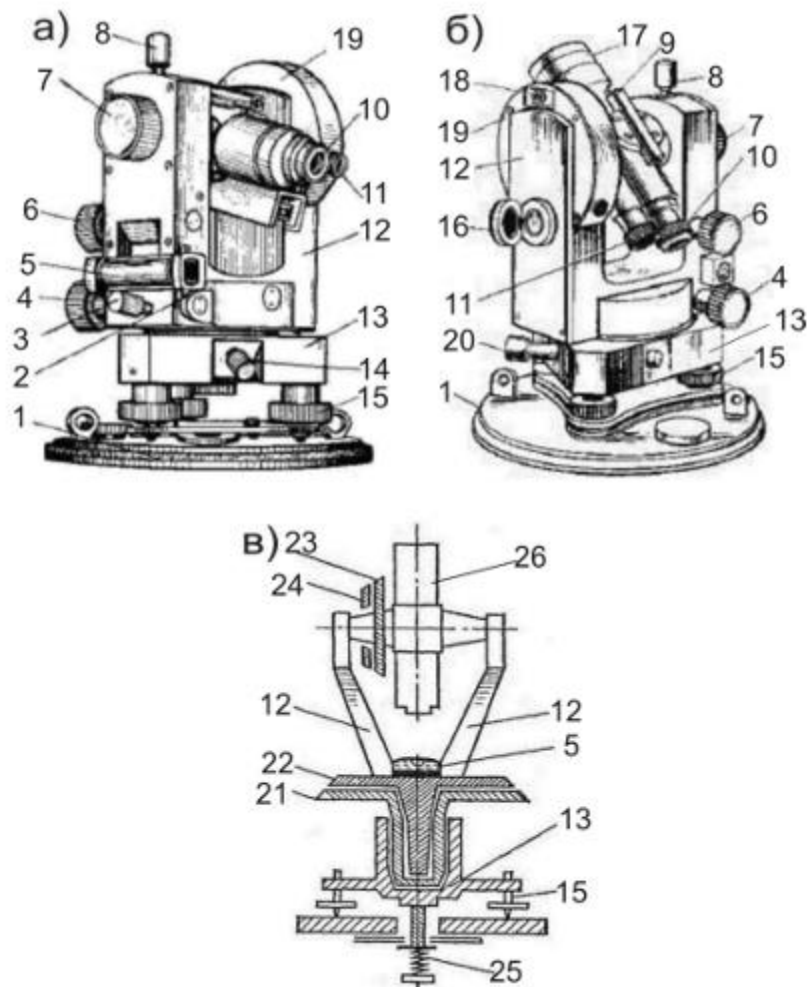


Рис. 1.3. Загальний вигляд і будова теодоліта:

а) вигляд при крузі праворуч; б) вигляд при крузі праворуч; в) схематичний розріз

Зорова труба (26) має об'єktiv (17), окуляр (10), фокусуючий гвинт (кремальєру) (7), візир (9), закріпний гвинт (8) і мікрометричний або навідний гвинт (6). За допомогою фокусуючого гвинта досягається чітке зображення предмета в полі зору труби, а за допомогою окуляра – чітке зображення сітки ниток. Візир призначений для швидкого попереднього наведення на точку. Закріпний гвинт закріплює трубу у будь-якому положенні, а мікрометричний гвинт дозволяє повільно і плавно обертати трубу при точному наведенні на

потрібну точку (навідний гвинт працює лише при закріпленому закріпному гвинті).

Закріпний і навідний гвинти мають також лімб і алідаду горизонтального круга: (3) – закріпний гвинт алідади ГК, (4) – навідний гвинт алідади ГК, (14) – закріпний гвинт лімба ГК, (20) – навідний гвинт лімба ГК.

Для підвищення точності відліку застосовують спеціальний пристрій – відліковий мікроскоп (11), у поле зору якого передається зображення штрихів лімбів ГК і ВК та шкали. Для підсвічування відліків використовується дзеркало (16).

Крім того, в комплект теодоліта входить орієнтир-бусоль, яка призначена для вимірювання магнітних азимутів. Для її кріплення в теодоліті є спеціальний паз (18) на колонці зорової труби.

1.2.2 Приведення теодоліта в робоче положення

Приведення теодоліта в робоче положення включає центрування, горизонтування приладу й фокусування зорової труби.

Центрування – це встановлення центра лімба або осі алідади на одній прямовисній лінії з вершиною кута. Для центрування використовують ниткові виски і оптичні центрири. Оскільки теодоліт 2Т30 не має оптичного центриру, ми будемо користуватись виском (рис. 1.4).

Для центрування теодоліт встановлюють над вершиною кута так, щоб головка штатива була наближено горизонтальною, а висок знаходився над кілочком, який позначає вершину кута. Ніжки штатива встановлюють у ґрунт, натиснувши ногою на металеві наконечники. Переконавшись у стійкості приладу, необхідно послабити становий гвинт і виконати більш точне центрування, переміщуючи теодоліт на головці штатива, поки висок не суміститься з точкою на місцевості. Після закінчення операції центрування закріплюють становий гвинт.

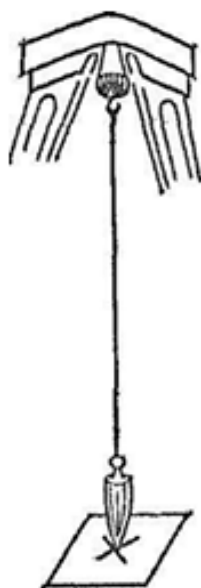


Рис. 1.4. Центрування за допомогою виска

Горизонтування – приведення площини лімба в горизонтальне положення або осі алідади в прямовисне положення піднімальними гвинтами.

Для горизонтування спочатку встановлюють циліндричний рівень горизонтального круга паралельно до двох піднімальних гвинтів і приводять його бульбашку на середину (рис. 1.5, а). Потім повертають алідаду на 90° у напрямку третього гвинта (рис. 1.5, б). Обертаючи лише третій піднімальний гвинт, знову приводять бульбашку в нуль-пункт. Ці дії повторюють декілька разів, поки бульбашка рівня не залишатиметься на середині.

Фокусування зорової труби – це отримання в полі зору труби чіткого зображення сітки ниток і предмету, який спостерігається. Зорову трубу наводять на предмет і, обертаючи кремальєру, фокусують трубу, тобто добиваються чіткої, різко окресленої видимості предмета. Далі, обертаючи окулярне кільце, добиваються чіткого зображення сітки ниток. При спостереженні різновіддалених предметів кожного разу змінюють фокусування.

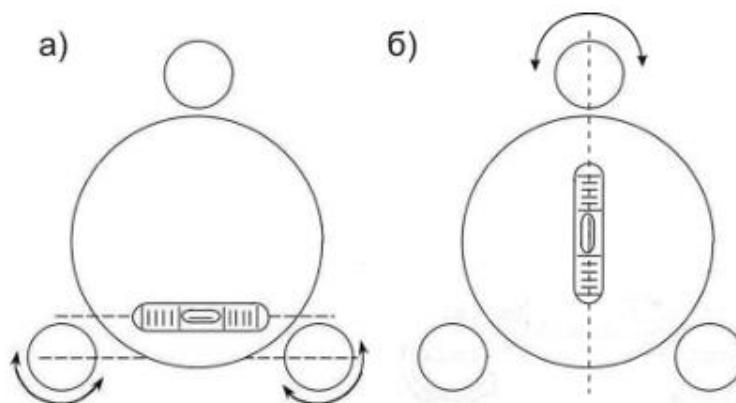


Рис. 1.5. Горизонтування теодоліта за допомогою циліндричного рівня: а) циліндричний рівень встановлений за напрямком двох піднімальних гвинтів; б) циліндричний рівень встановлений за напрямком третього піднімального гвинта

1.2.3 Загальні характеристики теодолітів

Загальні характеристики теодолітів представлені в таблиці 1.1.

Таблица 1.1. Загальні характеристики технічних теодолітів

Найменування характеристик	Типи теодолітів		
	T30	2T30	T15
1. Ціна ділення лімба	10'	шкалової мікроскоп	1
2. Відлікове пристосування	штриховий мікроскоп	шкалової мікроскоп	шкалової мікроскоп
3. Довжина шкали відлікового пристосування		10	10
4. Кількість ділень в шкалі		12	60
5. Ціна ділення шкали		5'	1'
6. Точність відліку	1'	0,5'	0,1'

1.2.4 Огляд, перевірки та дослідження теодолітів

Огляд теодолітів. До початку польових робіт роблять ретельний огляд, повірки й дослідження правильності роботи окремих вузлів теодоліта. При огляді теодолітів виявляють наявність усіх пристроїв і приладдя, зовнішні дефектні ознаки - вм'ятини, тріщини, відколи й т.п., що визначають у цілому довіру до приладу. Особлива увага приділяється перевірці роботи окремих рухливих пристроїв.

1. Піднімальні й навідні гвинти повинні мати плавний хід, без стрибків, зривів і заїдань. Для перевірки роботи гвинтів трубу теодоліта наводять звичайно на віддалену точку й, обертаючи піднімальний або навідний гвинт, стежать за плавністю переміщення зображення точки щодо ниток сітки. Хід піднімальних гвинтів регулюють їх гвинтами, а навідних - їх пружинами й гайками.

2. Навідний пристрій лімба, засувка повторювального пристрою, механізм оптичного мікрометра повинні працювати плавно й стійко.

3. Обертання теодоліта навколо вертикальної осі (лімба або алідади) та зорової труби повинне бути легким і плавним.

4. Зображення в оптичних відлікових пристроях, зорових трубах повинні бути чіткими, не деформованими. При цьому звертають увагу, щоб нерухомий індекс відлікового мікроскопа при установці на його шкалі відліку «нуль» збігався зі сполученими зображеннями діаметрально протилежних штрихів лімба, освітленість поля зору була рівномірною, було відсутнє перекривлення зображень, зображення штрихів лімба й шкали мікрометра можна було побачити без перефокусування окуляра мікроскопа.

1.3 Вимірювання кутів

1.3.1 Вимірювання горизонтальних кутів

Горизонтальні кути можна вимірювати наступними способами:

- 1) Спосіб прийомів
- 2) Спосіб кругових прийомів

Спосіб прийомів використовується, коли необхідно виміряти один кут на точці (рис. 1.6, а), а спосіб кругових прийомів – коли необхідно виміряти два або більше кутів з однієї вершини (рис. 1.6, б).

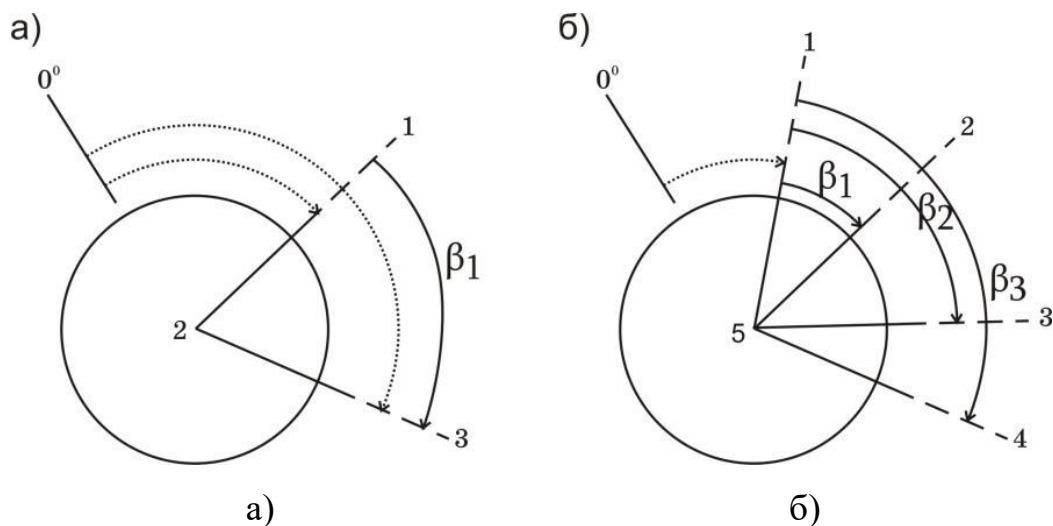


Рис. 1.6. Способи вимірювання горизонтальних кутів:

а) спосіб прийомів; б) спосіб кругових прийомів

Кути способом прийомів вимірюють, дотримуючись такої послідовності:

- 1) теодоліт встановлюють над вершиною кута і приводять у робоче положення;
- 2) визначають, яка точка буде правою, а яка лівою. Для цього стають на вершині обличчям до кута – за правою рукою буде права точка, за лівою – ліва;
- 3) закріплюють лімб у положенні круг праворуч (КП) і, повертаючи алідадну частину, наводять перехрестя сітки ниток зорової труби на низ віхи, встановленої

у правій точці 3 (рис. 1.6, а). Наведення здійснюють спочатку приблизно – за допомогою візиру, а потім точно – за допомогою мікрометричних гвинтів. Після точного наведення труби на точку візування знімають відлік за горизонтальним кругом. Відліки записують у „Журнал вимірювання горизонтальних кутів теодолітом способом прийомів” у відповідну графу (позиція (1) у табл. 2);

4) після цього відкріплюють закріпний гвинт алідади і наводять перехрестя сітки ниток зорової труби на низ віхи, встановленої у лівій точці 1 і знову знімають відлік по ГК (2);

5) величину виміряного кута (3) знаходять, віднімаючи від відліку на праву точку відлік на ліву точку, тобто за правилом – правий напрямок мінус лівий $(3)=(1)-(2)$. Якщо відлік на праву точку буде меншим, ніж відлік на ліву точку, то до нього додають 360° ;

6) переводять зорову трубу через зеніт, тобто повертають теодоліт на 180° . Внаслідок цього вертикальний круг виявиться ліворуч від зорової труби, тобто займе положення – круг ліворуч КЛ. Після цього всі дії, вказані у пунктах 3-5, повторюють при положенні КЛ і вдруге обчислюють величину виміряного кута $(6)=(4)-(5)$.

Вимірювання одного й того ж кута при двох положеннях вертикального круга КП і КЛ складає один повний прийом. Вимірювання кута при якомусь одному положенні круга називається півприйомом. При вимірюванні кута повним прийомом ліквідується колімаційна помилка і помилка від неперпендикулярності осі обертання труби до осі обертання теодоліта.

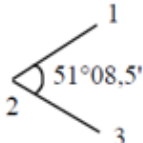
Якщо значення кута, визначені при положеннях КП і КЛ, відрізняються не більше ніж на подвійну точність відлікового пристрою, то з них виводять середнє і записують у відповідні графи журналу, вважаючи цей запис величиною кута 2. Якщо різниця між величинами кута, отриманими при положеннях КП і КЛ, більша від подвійної точності теодоліта, вимірювання кута треба провести ще раз.

Приклад заповнення журналу вимірювання горизонтальних кутів теодолітом

					ХНТУ 151.КРБ.24.02 ПЗ	Арк.
						26
Зм.	Арк.	№ докум	Підпис	Дата		

способом прийомів наведено в табл. 1.2.

Таблиця 1.2. Журнал вимірювання горизонтальних кутів теодолітом способом прийомів Теодоліт 2Т30

Станція	Точки наведення	Положення ВК	Відліки по ГК	Кути з пів-приймів	Середнє значення кута	Схема кута
2	3 1 3 1	КП КЛ	45°27' (1) 354°18' (2) 225°26' (4) 174°18' (5)	51°09' (3) 51°08' (6)	51°08,5'	

1.3.2 Вимірювання вертикального кута

Кутом нахилу ν називають кут між горизонтальною площиною і напрямком на необхідну точку (рис. 1.7). Прийнято називати кути нахилу вертикальними. Якщо точка, на яку необхідно визначити кут нахилу, розміщена вище від горизонтальної площини, то кут нахилу буде із знаком плюс, а якщо нижче – мінус.

У процесі вимірювання кутів нахилу спочатку визначають місце нуля вертикального круга ($МО ВК$). $МО ВК$ – відлік з вертикального круга, при якому візирна вісь зорової труби горизонтальна і бульбашка рівня при алідаді $ВК$ знаходиться в нуль-пункті.

Для визначення $МО$ приводять теодоліт у робоче положення, наводять центр сітки ниток на добре видиму точку і знімають відлік з вертикального круга $КЛ$. Далі переводять трубу через зеніт і проводять такі ж операції при другому положенні вертикального круга, отримуючи відлік $КП$.

Місце нуля $МО$ вертикального круга визначають за відповідними формулами, залежно від конструкції теодоліта.

Зокрема, для теодоліта 2Т30:

$$MO = \frac{KL + KP}{2}. \quad (1.1)$$

Постійне значення місця нуля є *контролем* вимірювань, тому необхідно визначити його декілька разів на різні точки місцевості. За остаточне значення місця нуля приймають середнє.

Визначивши *МО*, переходять до *вимірювання вертикальних кутів* у такій послідовності (рис. 1.7):

- 1) теодоліт встановлюють над точкою і приводять у робоче положення;
- 2) вимірюють *висоту приладу* (від верху кілочка до візирної осі) рейкою чи рулеткою;
- 3) зорову трубу у положенні круг праворуч (*КП*) наводять на віху, встановлену в кінцевій точці лінії. Причому перехрестя сітки ниток наводять на висоту приладу, відмічену на вісі;
- 4) після точного наведення зорової труби знімають відлік за вертикальним кругом. Відліки записують у „Журнал вимірювання вертикальних кутів теодолітом” у відповідну графу (табл. 1.3);
- 5) переводять трубу через зеніт і повторюють дії, зазначені у *пунктах* 3- 4 при крузі ліворуч (*КЛ*);
- 6) обчислюють *кут нахилу* v за формулами:

$$v = KL - MO = MO - KP = \frac{KL - KP}{2}. \quad (1.2)$$

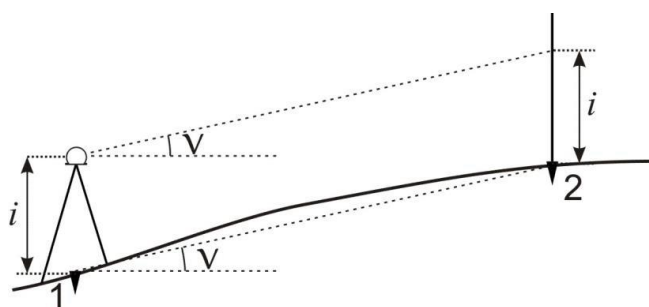


Рис. 1.7. Схема вимірювання вертикального кута

Якщо значення кута нахилу, визначені при положеннях *КП* і *КЛ*, відрізняються не більш як на подвійну точність приладу, то з них виводять середнє і записують у відповідні графи журналу, вважаючи цей запис величиною вертикального кута нахилу лінії. Якщо різниця між величинами кута, отриманими при положеннях *КП* і *КЛ*, більша від подвійної точності відлікового пристрою, то вимірювання вертикального кута необхідно провести ще раз [4].

Таблиця 1.3. Журнал вимірювання вертикальних кутів теодолітом Теодоліт 2Т30

Станції	Точки наведення	Відліки за ВК		МО	Кути нахилу		Кут нахилу середній
		КП	КЛ		МО-КП	КЛ-МО	
1	2	+1°31'	-1°25'	+0°03'	-1°28'	-1°28'	-1°28,0'
2	3	-2°23'	+2°28'		+2°26'	+2°25'	+2°25,5'
3	4	+3°44'	-3°39'		-3°41'	-3°42'	-3°41,5'

1.4 Огляд існуючих аналогів

1.4.1 Оптичні теодоліти

Оптичний теодоліт - це прилад, що оснащений оптичним відліковим пристроєм (оптична візирна труба) для визначення кутів (вертикальних і горизонтальних) і відстаней (за допомогою далекомірних ниток або додатково встановленого далекоміра).

В оптичних теодолітах застосовуються скляні лімби і оптичні системи, що дозволяють знімати відліки по горизонтальному і вертикальному кругам за допомогою одного мікроскопа, розташованого поруч з окуляром зорової труби. Оптичні теодоліти відносять до більш ранніх моделей теодолітів.

Вони мають свої переваги і недоліки. До переваг можна віднести: надійність; широкий діапазон робочих температур; робота без живлення; точні вимірювання за будь-яких погодних умов. До недоліків відносять: для точних вимірювань

потрібна висока кваліфікація фахівця; вимірювання займають відносно тривалий час [5].



а)

(випускався до 1972р.)



б)

(2007р.)



в)



г)

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 151.КРБ.24.02 ПЗ

Арк.

30



д)

Рис. 1.8. Оптичні теодоліти: а) ТВ-3, б) 3Т5КП, в) 4Т30П, г) Т15МКП, д) 2Т30П

1.4.2 Електронні теодоліти

Електронний теодоліт - це сучасний різновид теодоліта. У ньому кутомірні градуйовані круги замінені на електронні штрих-кодові диски з датчиками абсолютного кута повороту, а шкали мікроскопа - на зручний графічний дисплей.

Електронні теодоліти також мають електронний блок управління і компенсатор вертикального круга. А також можуть бути оснащені лазерним виском і мати можливість запису даних вимірювань у власну пам'ять або на зовнішній накопичувач.

Основна перевага електронних теодолітів - це простота і точність зняття відліків з дисплея, ймовірність помилки при цьому максимально мала. Деякі моделі цифрових теодолітів оснащуються лазерним покажчиком цілей для виконання розбивочних робіт або перевірки вертикальності споруд. Також з'являються нові моделі електронних теодолітів, оснащених вбудованим далекоміром - що перетворює цей прилад майже на тахеометр [6].



а)



б)



в)



г)

Рис. 1.9. Електронні теодоліти: а) Prexiso TO2, б) Nivel System DT-5, в) BOSCH CST / berger DGT 2, г) Nivoline DT02L

У шифрі теодоліта літера „Т” означає „теодоліт”, а цифри – середню квадратичну похибку вимірювання горизонтального кута одним прийомом (у секундах). Додаткова літера в шифрі теодоліта означає його модифікацію або конструктивне рішення: „А” – астрономічний, „М” – маркшейдерський, „К” – з компенсатором вертикального круга, „П” – зорова труба з прямим зображенням. Характеристики наведених вище теодолітів містяться у таблиці 1.4.

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 151.КРБ.24.02 ПЗ

Арк.

32

Таблиця 1.4

Найменування· характеристик			Типи теодолітів								
			Оптичні					Цифрові			
			TB-3	3T5KP	4T30P	T15MKP	2T30P	LEICA· PREXISO· T02	NIVEL· SYSTEM· DT-5	BOSCH· CST·/ berger· DGT-2	NIVOLINE· DT02L
Середньо- квадратична· похибка· вимірювання			2-3''	5''	30''	15''	30''	2''	5''	2''	2''
Поле зору			1°30'	1°35'	2°	□	2°	1°30'	□	1°30'	□
Ціна поділу	Лімбус	Горизонталь	20'	1°	1°	□	1°	□	□	□	□
		Вертикаль	20'	1°	1°	□	1°	□	□	□	□
	шкала· мікрометра		1''	1''	5'	1''	5'	□	□	□	□
	Циліндричного·рівня· алідади· горизонтального·		20''	□	□	□	□	□	□	□	□
	Циліндричного· рівня·вертикального· кола		25''	□	□	□	□	□	□	□	□
Найменша· відстань· візування, м			1,2	0,9	1,2	□	1,2	1,35	1,5	1,3	1,5

Якщо на основі базової моделі розроблена нова модифікація, перед шифром додається цифра „2” [2].

1.5 Приклад автоматизації роботи теодолітів

Електронні теодоліти – найновіші досягнення електроніки і приладобудування. В нових високоточних теодолітах замість традиційної відлікової системи з скляним лімбом і оптичним мікрометром використовується оптична система відліку по кругах з оптико-електронним скануванням, яка дозволяє автоматизувати процес вимірювання кутів і одночасно підвищити їх приборну точність [7]. До таких теодолітів відноситься теодоліт Nivel System DT-2.



Рис. 1.10. Nivel System DT-2

Наприклад він має такі можливості:

1 Запис і виведення даних

Ця серія теодолітів забезпечує функцію запису даних вимірювань. Дані про кут і дані про відстань можна зберігати в пам'яті приладу (до 1000 груп) або виводити через інтерфейс зв'язку. Записані дані містять інформацію про час. Перед записом даних слід вибрати метод запису. Якщо вибрано запис даних через інтерфейс зв'язку, потрібно правильно налаштувати зв'язок.

2 Інтерфейс зв'язку RS-232C

Підключивши прилад до комп'ютера або КПК за допомогою кабелю, дані вимірювань можна передати на комп'ютер або обладнання для збору даних. Пам'ятайте, що інтерфейс знаходиться під вертикальною ручкою.

3 Запис даних вимірювання

В іншому режимі вимірювання натисніть SFT, а потім натисніть (REC), дані вимірювання можна вивести на комп'ютер чи КПК (при виборі методу запису даних через інтерфейс зв'язку) або зберегти в пам'яті пристрою. інструмент (при виборі способу запису даних у пам'ять) [8].

На сучасному ринку представлено дуже багато виробників високоточних електронних теодолітів, їхні можливості з кожним роком зростають і покращуються.

Електронні теодоліти мають вертикальну, горизонтальну та візуючу осі. як звичайні теодоліти. І площини основних кругів в них також розташовані так само. Тобто механічна частина традиційних і електронних теодолітів - однакова. Основною відмінністю електронних теодолітів від оптичних є електронна система визначення кутів і напрямків замість візуального зчитування відліків. Для цього, традиційні круги з поділками замінюють кодовими кругами, або кругами з растрами та вмонтовують в теодоліт електронні зчитувачі, процесори, міні-ЕОМ і запам'ятовуючі пристрої. Такі зміни дозволили автоматизувати процес відчитування відліків з кругів і "запам'ятовувати" їх на електронних носіях інформації.

В більшості сучасних електронних теодолітів точність вимірювання горизонтальних і вертикальних кутів однакова [9].

Електронні теодоліти мають такі переваги:

- автоматизація зняття відліків;
- надання даних у зручному для обробки форматі;
- збільшення швидкості вимірів у зв'язку з їхньою автоматизацією;
- простота в налаштуванні та використанні;

- виключення помилок, пов'язаних із людським фактором.

Електронні теодоліти мають підвищену точність вимірів. У них є пристрої контролю вихідних даних, автоматично виправляються дрібні помилки і можуть бути вирішені прості геометричні завдання. Сьогодні багато компаній, які проводять різні інженерні дослідження, прагнуть купити електронні теодоліти. Вони значно збільшують швидкість проведення типових вимірювальних робіт. Однак не варто забувати, що купити оптичні теодоліти просто необхідно тим, хто має намір проводити вимірювання у важких кліматичних умовах. Крім того, оптичні пристрої набагато дешевші, а якість одержуваних даних досить висока для деяких галузей [10].

1.6 Висновок до першого розділу

Виконаний огляд будови, класифікації, проведення вимірювань і огляд існуючих теодолітів та приклад автоматизації цих вимірювальних приладів. У основу теми роботи лягла ідея створення більш дешевої, та автоматизованої версії теодоліту для того, щоб в подальшому використовувати для проведення вимірювань розмірів, об'ємів та іншого безконтактним методом.

2 МАТЕМАТИЧНА ЧАСТИНА. ОТРИМАННЯ КУТА ОБЕРТУ. КВАТЕРНІОНИ

2.1 Кватерніони

Для того щоб реалізувати вимірювання кутів за допомогою електронних компонентів акселерометра та гіроскопа, ми скористаємося теорією кватерніонів.

Кватерніон - це впорядкована четвірка дійсних чисел

$$q = [q_0, q_1, q_2, q_3], \quad (2.1)$$

що містить в собі інформацію про скаляр і тривимірний вектор. В компонентах кватерніон будемо зображати у вигляді рядка або стовпця (залежно від зручності написання) у квадратних дужках.

Ця представлення дозволяє ефективно виконувати операції обертання, враховуючи усі аспекти орієнтації об'єкту в просторі. Вони дозволяють зручно представляти та обчислювати обертання об'єктів в тривимірному просторі.

У кватерніоні розрізняють скалярну частину

$$\text{scal } q = q_0, \quad (2.2)$$

яка представляє собою скаляр, і векторну частину

$$\text{vect } q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (2.3)$$

що являє собою тривимірний вектор.

Кватерніон, що має тільки скалярну частину ,

$$q = [q_0, 0, 0, 0], \quad (2.4)$$

називається чисто скалярним.

Кватерніон , що має тільки векторну частину ,

$$q = [0, q_1, q_2, q_3], \quad (2.5)$$

називається чисто векторним.

Для кватерніона визначено також декілька скалярних характеристик.

1. Норма:

$$\|q\| = q_0^2 + q_1^2 + q_2^2 + q_3^2. \quad (2.6)$$

2. Модуль:

$$|q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \quad (2.7)$$

3. Модуль векторної частини:

$$\langle q \rangle = \sqrt{q_1^2 + q_2^2 + q_3^2}. \quad (2.8)$$

для позначення якого будемо використовувати ідентифікатор кватерніона в кутових дужках .

4. Аргумент – кутова величина, що дорівнює аргументу комплексного числа з дійсною частиною q_0 та уявною частиною $\langle q \rangle$ (рис. 2.1), що приймає значення із замкнутого інтервалу $[0, \pi]$.

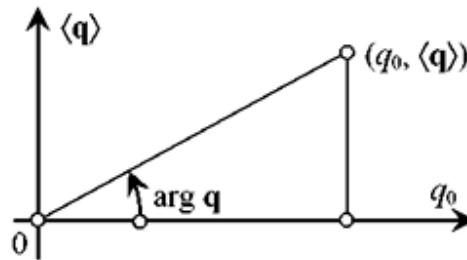


Рис. 2.1. Аргумент кватерніона

Аргумент кватерніона, зазвичай, визначають формулою:

$$\arg q = \arccos \frac{q_0}{|q|}. \quad (2.9)$$

Однак ця формула погано обумовлена поблизу країв інтервалу. Для обчислень краще використовувати вбудовані функції $\arg(x+iy)$ (комплексного числа) або $\text{atan2}(x,y)$, наявні в багатьох математичних програмах, які використовують не тільки скалярну частину кватерніона і його модуль, але і модуль векторної його частини [11]:

$$\arg q = \arg[q_0 + i\langle q \rangle] = \text{atan2}[q_0; \langle q \rangle]. \quad (2.10)$$

Як випливає з формул (2.6) і (2.7), норма і модуль кватерніона дорівнюють нулю тільки для нульового кватерніона

$$0 = [0,0,0,0]. \quad (2.11)$$

Аргумент нульового кватерніона не визначений.

5. Кватерніон нормований, якщо його норма дорівнює одиниці. Верзором чи знаком кватерніона будемо називати нормований кватерніон, отриманий на його основі:

$$\bar{q} = [q_0, -q_1, -q_2, -q_3]. \quad (2.12)$$

2.2 Отримання кута оберту у градусній мірі

Якщо кватерніон представлений як:

$$qw + i qx + j qy + k qz. \quad (2.13)$$

Тоді:

$$q = [w, x, y, z]. \quad (2.14)$$

Квадрати скалярної та векторної частини

$$q \cdot w^2 = q \cdot w \cdot q \cdot w \quad (2.15)$$

$$q \cdot x^2 = q \cdot x \cdot q \cdot x$$

$$q \cdot y^2 = q \cdot y \cdot q \cdot y$$

$$q \cdot z^2 = q \cdot z \cdot q \cdot z.$$

Визначення векторів (аргументів):

$$\vec{x} = \text{atan2}(2.0 * (q.x * q.y + q.z * q.w), (q.x^2 - q.y^2 - q.z^2 + q.w^2)), \quad (2.16)$$

$$\vec{y} = \text{asin}(2.0 * (q.x * q.z + q.y * q.w), (q.x^2 + q.y^2 + q.z^2 + q.w^2)), \quad (2.17)$$

$$\vec{z} = \text{atan2}(2.0 * (q.y * q.z + q.x * q.w), (-q.x^2 - q.y^2 + q.z^2 + q.w^2)). \quad (2.18)$$

Переведення значення аргументів (радіанів) у градуси:

$$angle.x = \vec{x} \cdot 180/\pi, \quad (2.19)$$

$$angle.y = \vec{y} \cdot 180/\pi, \quad (2.20)$$

$$angle.z = \vec{z} \cdot 180/\pi, \quad (2.21)$$

2.3 Інші одиниці вимірювань кутів

Дві найпоширеніші одиниці вимірювання кутів – градуси і радіани. Ступені базуються на давньому месопотамському присвоєнні 360° до повного кола. Це бере свій початок у поділі горизонту нічного неба, оскільки землі потрібно 365 днів, щоб подорожувати навколо сонця.

Вимірювання значення аргументу можливе не лише за радіанами та градусами. Також можливі та поширені вимірювання за DMS (градуси°, хвилини', секунди") та GRD gon (grad).

Оскільки градуси спочатку були розроблені месопотамцями, вони часто також розбиваються на 60 одиниць вимірювання хвилин і секунд. Шістдесят секунд роблять одну хвилину, а шістдесят хвилин - один градус.

$$60 \text{ секунд} = 1 \text{ хвилинка (або } 60'' = 1'), \quad (2.22)$$

$$60 \text{ хвилин} = 1 \text{ градусів (або } 60' = 1^\circ), \quad (2.23)$$

Кути, виміряні в градусах, також можуть бути виражені за допомогою десяткових частин градуса, наприклад:

$$72.5^\circ = 72^\circ 30'$$

Перетворення з десяткового в DMS

Перетворення між градусами, вираженими десятковими знаками, і форматом градусів, хвилин, секунд (DMS) є відносно простим. Якщо ми

перетворюємо з градусів, виражених десятковими знаками в DMS, просто візьмемо частину кута за десятковою крапкою і помножимо на 60. У нашому попередньому прикладі, ми б взяли 0.5 з 72.5° і помножили це на 60:

$$0.5 * 60 = 30.$$

Так, кут в одиницях DMS буде $72^\circ 30'$.

Приклади

Перетворити 21.85° в DMS одиниці виміру

$$0.85 * 60 = 51$$

Отже, $21.85^\circ = 21^\circ 51'$.

Перетворення одиниць 143.27° у DMS

$$0.27 * 60 = 16.20.27 * 60 = 16.2$$

Отже, для $143.27^\circ = 143^\circ 16.2'$ того щоб обчислити кількість секунд, необхідне для вираження цього кута в одиницях DMS, беремо десяткову частину хвилин і множимо на 60:

$$0.2 * 60 = 12.$$

Отже, $143.27^\circ = 143^\circ 16.2' = 143^\circ 16' 12''$.

У прикладі вище все закінчилося цілою кількістю секунд. Якщо ми не отримаємо ціле число за секунди, ми можемо залишити секунди з десятковою частиною. Наприклад, якщо ми хочемо перетворити 22.847° в одиниці виміру DMS:

$$22.847^{\circ} = 22^{\circ} 50.82' = 22^{\circ} 50.82' 49.2''$$

Для переведення градусів у гони необхідно скористатися таким співвідношенням:

$$1 \text{ гон} = 0,9 \text{ градусів (або } 1 \text{ гон} = 0,9^{\circ}\text{)}.$$

Тобто, для переведення градусів у гони:

$$\text{гон} = \frac{\text{градус}}{0,9}.$$

Приклади

Перший приклад з дробовою частиною:

$$\text{гон} = \frac{72,5}{0,9} = 80, \bar{5}.$$

Другий приклад переводиться без дробової частини:

$$\text{гон} = \frac{27}{0,9} = 30 \text{ гон}$$

Тепер в подальшому ми можемо використовувати ці співвідношення для виведення виміряного кута у необхідному форматі, забезпечуючи точність і зручність у роботі з різними системами вимірювання кутів.

2.4 Фільтрація шуму та корекція похибок

Розглянемо потребу у фільтрації шуму та корекції похибок, які є важливими аспектами при обробці даних у реальному часі. Давйте приділимо увагу нормалізації кватерніонів, що використовуються для орієнтації об'єктів у тривимірному просторі.

2.4.1 Нормалізація кватерніонів

Нормалізація кватерніонів є необхідною для забезпечення їхньої правильної величини, яка для одиничних кватерніонів дорівнює одиниці. Це допомагає уникнути математичних помилок під час операцій з кватерніонами

Це означає, що оптимізація вже потрібна. Числа з плаваючою комою не ідеальні. Згодом ми втратите точність з ними, і ми можете перевірити, чи потрібно нам повторно нормалізувати одиницю кватерніону, щоб запобігти математичним помилкам. Це можна зробити, перевіривши величину, яка має бути одиницею але якщо ми перевіряємо, чи дорівнює вона одиниці, у `sqrt()` взагалі немає потреби ($\text{sqrt}(1) = 1$).

Так, якщо $(w^2 + x^2 + y^2 + z^2)$ більше ніж певний допуск від 1, ми будемо знати, що наш кватерніон потрібно нормалізувати, і ми заощадимо на витягуванні квадратного кореня у випадках, коли він знаходиться в межах допуску. Кожні кілька тактів мають значення. Тоді процес нормалізації виконується наступним чином:

1. Розрахунок величини кватерніона за допомогою формули:

$$\text{величина} = \sqrt{w^2 + x^2 + y^2 + z^2} \quad (2.24)$$

2. Нормалізація кватерніона, виконується розділивши кожен його компонент на величину:

$$w = \frac{w}{\text{величина}}, \quad (2.25)$$

$$x = \frac{x}{\text{величина}}, \quad (2.26)$$

$$y = \frac{y}{\text{величина}}, \quad (2.27)$$

$$z = \frac{z}{\text{величина}}. \quad (2.28)$$

Це дозволяє забезпечити, що кватерніони завжди мають одиничну величину і готові для подальших операцій.

2.5 Обробка та управління об'єктами

Для ефективного управління об'єктами використовуються кватерніони, які дозволяють уникнути карданного блокування та забезпечують плавні рухи об'єктів. Ключовою операцією є обертання об'єктів навколо заданих вісей.

2.5.1 Генерація кватерніону для обертання

Перш ніж почати обертання об'єкта, необхідно створити локальний кватерніон обертання (*local_rotation*), який представляє зміну орієнтації об'єкта. Для цього використовується формула:

$$local_rotation_w = \cos(fAngle2), \quad (2.29)$$

$$local_rotation_x = axis\ x \cdot \sin(fAngle2), \quad (2.30)$$

$$local_rotation_y = axis\ y \cdot \sin(fAngle2), \quad (2.31)$$

$$local_rotation_z = axis\ z \cdot \sin(fAngle2). \quad (2.32)$$

де *angle* - кут обертання, *axisx*, *axisy*, *axisz* - компоненти вектора вісі обертання.

2.5.2 Застосування обертання

Отримавши *local_rotation*, ми можемо застосувати обертання до об'єкта, помноживши його на поточний кватерніон орієнтації (*total*). Це можна зробити наступним чином:

$$total = local_rotation \cdot total, \quad (2.33)$$

Ця операція дозволяє обертати об'єкт навколо будь-якої визначеної вісі на довільний кут, уникаючи карданного блокування [12].

2.6 Розрахунок керуючих впливів до СПМ

Для використання СПМ нам необхідно розраховувати керуючі впливи для сервоприводів. Для розрахунку цих керуючих впливів використаємо метод заснований на розрахунку оберненої кінематичної задачі для СПМ.

Радіуси, що описують обмеження положення шарнірів В, В2, В3 по осі z, обчислюються лише один раз за початковими координатами, заданими у формулах R2, R3 та R4:

$$\begin{cases} R_2 = \sqrt{R_C^2 - z_0^2}, \\ R_3 = \sqrt{R_C^2 - z_2^2}, \\ R_4 = \sqrt{R_C^2 - z_4^2}. \end{cases} \quad (2.34)$$

де R_C – довжина ланки між шарнірами С і В, С2 і В2, С3 і В3, z_0, z_2, z_3 – положення шарнірів С, С2, С3.

Щоб визначити кут повороту шестерень А, А2 і А3, необхідно знати положення шарнірів В, В2, В3 уздовж осі х або у. Наступні формули описують ці позиції:

$$B_X = \frac{x_0^4 + R_1^2 \cdot x_0^2 - R_0^2 \cdot x_0^2 + y_0^2 \cdot x_0^2 - \sqrt{F} \cdot y_0}{2 \cdot x_0 \cdot (x_0^2 + y_0^2)}, \quad (2.35)$$

$$B_Y = \frac{y_0}{2} + \frac{R_1^2 \cdot y_0 - R_0^2 \cdot y_0 + \sqrt{F}}{2 \cdot (x_0^2 + y_0^2)}, \quad (2.36)$$

$$B_{2X} = \frac{x_{02}^4 + R_1^2 \cdot x_2^2 - R_2^2 \cdot x_2^2 + y_2^2 \cdot x_2^2 - \sqrt{F} \cdot y_2}{2 \cdot x_2 \cdot (x_2^2 + y_2^2)}, \quad (2.37)$$

$$B_{2Y} = \frac{y_2}{2} + \frac{R_1^2 \cdot y_2 - R_2^2 \cdot y_2 + \sqrt{F}}{2 \cdot (x_2^2 + y_2^2)}, \quad (2.38)$$

$$B_{3X} = \frac{x_3^4 + R_1^2 \cdot x_3^2 - R_3^2 \cdot x_3^2 + y_3^2 \cdot x_3^2 - \sqrt{F} \cdot y_3}{2 \cdot x_3 \cdot (x_3^2 + y_3^2)}, \quad (2.39)$$

$$B_{3Y} = \frac{y_3}{2} + \frac{R_1^2 \cdot y_3 - R_3^2 \cdot y_3 + \sqrt{F}}{2 \cdot (x_3^2 + y_3^2)}, \quad (2.40)$$

де

$$\begin{aligned} F = & -R_1^4 \cdot x_0^2 - x_0^6 - y_0^4 \cdot x_0^2 - 2 \cdot x_0^4 \cdot y_0^2 + \\ & + 2 \cdot x_0^4 \cdot R_2^2 - x_0^2 \cdot R_2^4 + 2 \cdot x_0^4 \cdot R_1^2 + \\ & + 2 \cdot y_0^2 \cdot R_2^2 \cdot x_0^2 + 2 \cdot R_1^2 \cdot y_0^2 \cdot x_0^2 + 2 \cdot x_0^2 \cdot R_1^2 R_2^2, \end{aligned} \quad (2.41)$$

де R1 — довжина ланки, що з'єднує шарнір В і шестерню А, x0 — положення шарніра С уздовж осі х, y0 — положення шарніра С уздовж осі у, z0 — положення шарніра С уздовж вісь z, R2, R3, R4 – радіуси, які описують обмеження шарнірів В, В2, В3 у положенні вздовж осі z і обчислюються лише один раз у початкових координатах.

Останні формули є результатом розрахунку оберненої кінематичної задачі для СПМ.

Повні кути, що описують положення шестерень A , A_2 , A_3 , вказують на те, який кут має бути від умовного «нуля», який пов'язаний з «нульовим» кутом сервоприводу:

$$\begin{aligned} A &= 180 / \pi \cdot \arccos (B_x/R_1), \\ A_2 &= 180 / \pi \cdot \arccos (B_{2x}/R_1), \\ A_3 &= 180 / \pi \cdot \arccos (B_{3x}/R_1). \end{aligned} \quad (2.42)$$

Кути повороту навколо «нульової» точки положення грані, яка є початком координат, для розрахунку кутів повороту можуть бути позитивними або негативними. Тобто кути A , A_2 , $A_3 \pm 90^\circ$ залежать від знака, з яким отримано вхідне значення кута з розрахунку оберненої задачі кінематики:

$$\begin{aligned} A &= 180 / \pi \cdot \arccos (B_y/R_1), \\ A_2 &= 180 / \pi \cdot \arccos (B_{2y}/R_1), \\ A_3 &= 180 / \pi \cdot \arccos (B_{3y}/R_1). \end{aligned} \quad (2.43)$$

Обертання в тривимірному просторі представляється як композиція обертань навколо трьох ортогональних осей. Це відповідає матриці, яка дорівнює добутку відповідних трьох матриць обертання.

Матриці обертання в тривимірному просторі задані у формулах (2.44-2.46) СПМ (рис. 3.5)

По осі X:

$$M_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}. \quad (2.44)$$

По осі Y:

$$M_Y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix}. \quad (2.45)$$

По осі Z:

$$M_Z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.46)$$

Отримані матриці перетворюються за початковими координатами для точок C, C2, C3 в координати для точок C', C'2, C'3 за такою формулою, а всі наступні повороти здійснюються з уже розрахованими координатами:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_X(\alpha) \cdot M_Y(\alpha) \cdot M_Z(\gamma) \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}, \quad (18)$$

де α, β, γ – кути повороту навколо осей x, y, z , x', y', z' – попередні координати по осі x, y, z [13].

3 АПАРАТНА ЧАСТИНА ПАК

3.1 Структурна схема

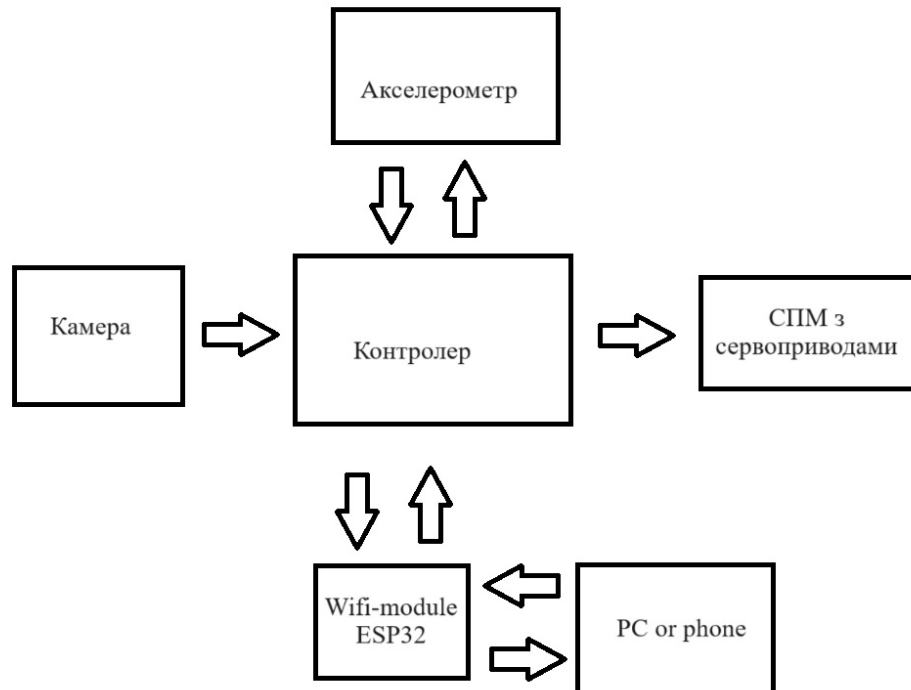


Рис.3.1 Структурна схема ПАК

3.1.1 Опис структури та роботи ПАК

ПАК представляє собою інтегровану систему для проведення точних вимірювань та спостережень. Його основні компоненти включають камеру, контролер Arduino, датчик MPU9250, модуль ESP32 для бездротового зв'язку, та (СПМ) до якої входять сервоприводи для управління обертанням. Робота ПАК починається з подачі живлення до всіх компонентів, що активує систему та ініціює процес вимірювань і спостережень.

При увімкненні живлення всі компоненти проходять етап ініціалізації. Камера підключається до керуючого пристрою (КП) за допомогою бездротового зв'язку по Wi-Fi. Контролер Arduino проводить перевірку підключених датчиків і готовність системи до роботи. Датчик MPU9250, який

включає в себе акселерометр, гіроскоп і магнітометр, також проходить калібрування, щоб забезпечити точність подальших вимірювань.

Після успішної ініціалізації, камера починає передавати відеоряд до КП через Wi-Fi. Це відео використовується для наведення камери на об'єкт вимірювань. Контролер Arduino періодично опитує датчик MPU9250, збираючи дані. Ці дані обробляються контролером, переводяться у відповідну систему вимірювань і відправляються до модуля ESP32.

Модуль ESP32 грає ключову роль у передачі даних від контролера Arduino до КП. Використовуючи Wi-Fi з'єднання, ESP32 забезпечує надійну та швидку передачу даних, що дозволяє оператору на КП отримувати актуальну інформацію в реальному часі. Це важливо для прийняття рішень та коригування роботи системи.

Обертання камери та датчика здійснюється за допомогою сервоприводів (СПМ). Управління СПМ здійснюється з КП через модуль ESP32, що забезпечує двосторонній зв'язок. СПМ обертає камеру та датчик відповідно до отриманих команд, починаючи з нульового положення. Це дозволяє точно навести камеру на об'єкт вимірювань та забезпечити оптимальні умови для збирання даних.

Під час наведення камери на об'єкт вимірювань, контролер Arduino проводить активний збір даних з датчика MPU9250. Отримані дані проходять первинну обробку на рівні мікроконтролера, що включає фільтрацію шумів та корекцію похибок. Потім ці дані передаються до модуля ESP32, який відправляє їх на КП.

У процесі роботи ПАК забезпечується постійний моніторинг стану системи. Контролер Arduino відслідковує параметри роботи датчиків та сервоприводів, а також стан з'єднання з модулем ESP32. У разі виникнення будь-яких проблем або аномалій, система негайно повідомляє про це оператора на КП. Це дозволяє оперативно вжити заходів для усунення несправностей та забезпечення безперервної роботи комплексу.

Зворотний зв'язок між КП та системою ПАК забезпечується через модуль ESP32. Оператор може надсилати команди для зміни параметрів роботи, калібрування датчиків або переналаштування камери. Це дозволяє адаптувати систему до змінних умов роботи та забезпечити максимальну точність вимірювань.

Після завершення сеансу вимірювань або спостережень, система ПАК переходить у режим збереження даних. Контролер Arduino забезпечує збереження зібраних даних, а модуль ESP32 передає їх на КП для остаточного зберігання та обробки. Камера повертається у нульове положення, а система готується до вимкнення.

Увімкнення живлення припиняється для всіх компонентів, і система переходить у стан очікування до наступного сеансу роботи. Перед вимкненням проводиться остання перевірка стану системи та збереження всіх критично важливих даних, щоб уникнути їх втрати.

Робота ПАК є складним та інтегрованим процесом, що включає різні етапи від ініціалізації до завершення сеансу вимірювань. Завдяки поєднанню сучасних технологій та точного управління, ПАК дозволяє отримувати високоточні дані для аналізу та прийняття рішень. Це робить його незамінним інструментом у різних галузях, де необхідна висока точність і надійність вимірювань.

3.2 Функціональна та принципова схема

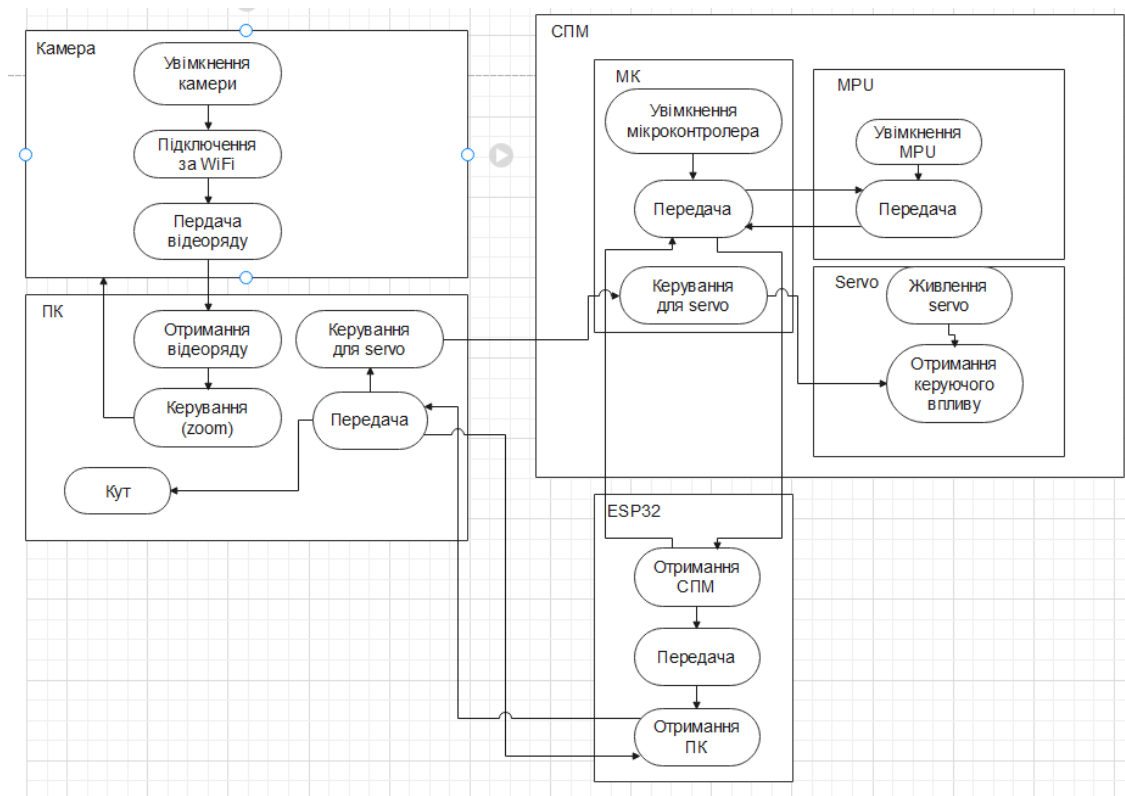


Рис. 3.2. Функціональна схема ПАК

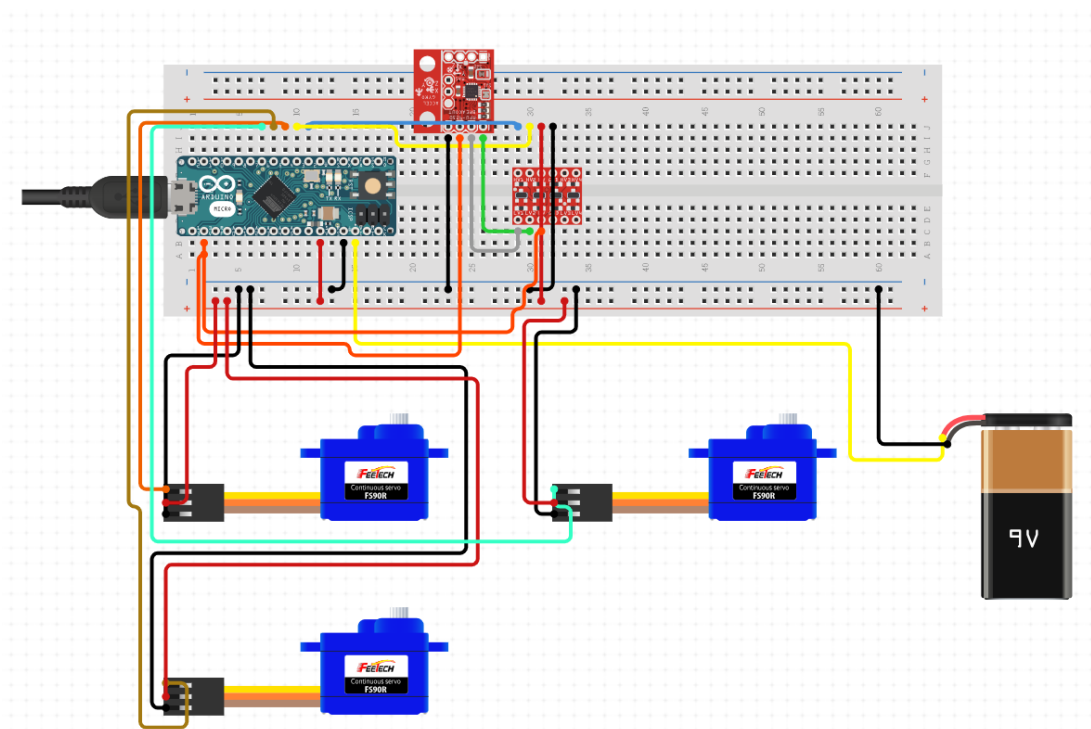


Рис. 3.2. Принципова схема ПАК

3.3 Вибір акселерометра

Гіроскоп - це механізм, який реагує на зміну орієнтації основи, на якій його встановлено, відносно інерціального простору. У класичному розумінні гіроскоп є пристроєм, що містить швидкообертове тверде тіло, яке має три ступені свободи обертання, тобто можливість обертання навколо трьох взаємно перпендикулярних осей.

Насправді, конструкція електронних гіроскопів дещо інша і вони влаштовані трохи складніше (не так просто було б впхнути обертову дзигу в мікросхему).

Акселерометр - це пристрій, що вимірює проекцію удаваного прискорення, тобто різниці між справжнім прискоренням об'єкта та гравітаційним прискоренням. Примітивний акселерометр можна уявити як вантаж, закріплений на пружині. Демпфер гасить коливання вантажу. Чим більше повне прискорення, тим сильніше деформується пружина, змінюючи показання пристрою. Таким чином, гіроскоп реагує на зміну орієнтації у просторі незалежно від напрямку руху, тоді як акселерометр дозволяє вимірювати лінійні прискорення об'єкта та розраховувати його положення в просторі. Кожен з цих пристроїв має свої переваги і недоліки.

MPU 9250 виконує функції відразу декількох сенсорів: гіроскопа, акселерометра та магнітометра - всі три пристрої в одному, що дуже зручно. У корпусі мікросхеми об'єднані два кристали: один містить 3-осьовий гіроскоп і 3-осьовий акселерометр, а інший - 3-осьовий магнітометр. Дані з сенсорів оцифровуються 16-бітними АЦП, обробляються із застосуванням алгоритмів Motion Fusion сигнальним процесором DMP (Digital Motion Processor) і передаються зовнішньому мікроконтролеру по шині I2C/SPI. Модуль MPU 9250 є найменшим у світі 9-осьовим сенсором.

Це свідчить про високу продуктивність мікросхеми, забезпечену застосуванням технології CMOS MEMS. Живити модуль можна від 5 В, так як на платі датчика

є лінійний стабілізатор для даного типу живлення, але подавати живлення можна і при напрузі 3,3 В.

MPU-9250 також призначений для взаємодії з іншими цифровими датчиками без інерції, такими як датчики тиску, через допоміжний порт I2C. Для точного відстеження як швидких, так і повільних рухів, деталі оснащені програмованим користувачем гіроскопом із повномасштабним діапазоном ± 250 , ± 500 , ± 1000 та $\pm 2000^\circ/\text{с}$ (dps), програмованим користувачем повномасштабним акселерометром діапазонів $\pm 2g$, $\pm 4g$, $\pm 8g$ та $\pm 16g$, а також повномасштабним діапазоном магнітометра $\pm 4800\mu\text{T}$. Інші провідні функції включають програмовані цифрові фільтри, точний годинник із відхиленням 1% від -40°C до 85°C , вбудований датчик температури та програмовані переривання. Пристрій забезпечує високу міцність, витримуючи ударне навантаження 10 000 g.

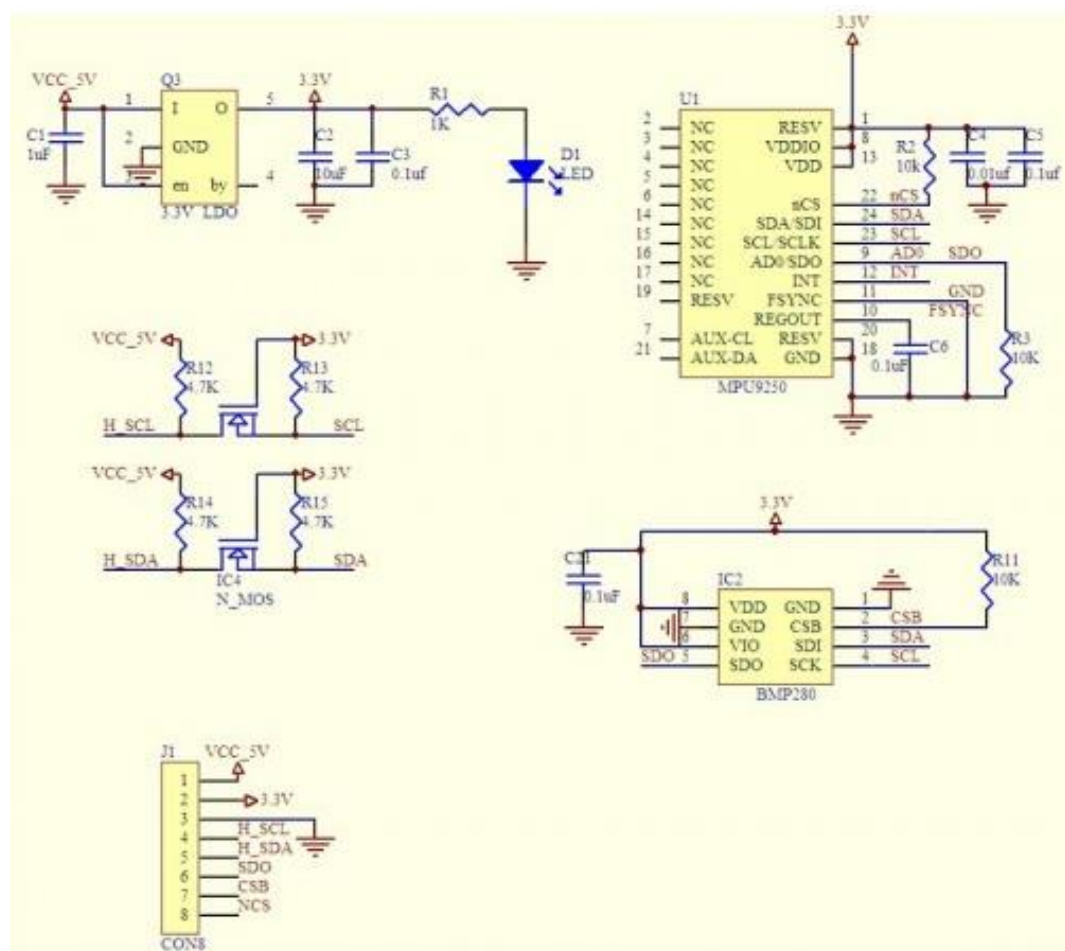


Рис. 3.3. Принципова схема MPU9250

Характеристики MPU9250:

- напруга живлення 3,3 – 5 вольт
- споживаний струм до 3,7 мА
- інтерфейс передачі даних - I2C
- максимальна швидкість I2C - 400 кГц
- вхід для інших датчиків I2C
- внутрішній генератор на 8 МГц (поза модулем є можливість підключити зовнішній кварцовий резонатор на 32,768 кГц або 19,2 МГц)
- Гіроскоп: $\pm 250 / 500 / 1000 / 2000$ °/сек
- Акселерометр: $\pm 2 / \pm 4 / \pm 8 / \pm 16$ g
- Магнітометр: $\pm 4800 \mu\text{T}$

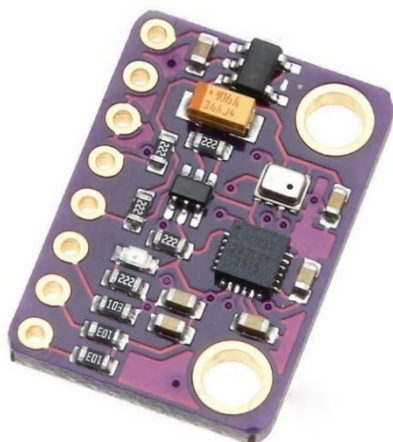
Потрібно відзначити можливість MPU9250 працювати в майстер режимі I2C для AUX виводів, до яких можна підключити ще один зовнішній датчик (наприклад датчики тиску барометр: 300-1100hPa).

Функції MPU9250

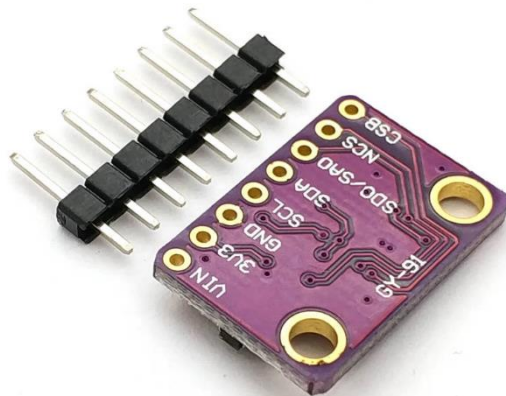
- 3-осьової MEMS гіроскоп з 16 бітовим АЦП
- 3-осьової MEMS акселерометр з 16 бітовим АЦП
- 3-осьової MEMS магнітометр з 16 бітовим АЦП
- Digital Motion Processor (DMP)
- slave I2C для підключення до мікроконтролера
- майстер I2C для підключення до мікросхеми додаткового датчика
- регістри даних датчиків
- FIFO
- переривання
- температурний сенсор
- самоперевірка гіроскопа і акселерометра

- реєстр ідентифікації пристрою

Зовнішній вигляд модуля GY-91:



а) вид зверху



б) вид з низу

Рис. 3.4 Зовнішній вигляд модуля GY-91

У комплекті додаються штирьові з'єднання прямі, пайку яких було виконано.

Дані вимірювань сенсорів можна зчитувати як з реєстрів зберігання, так і використовувати функції FIFO. Окремий висновок є виходом переривань, який налаштовується реєстрами під конкретні події.

Сенсори гіроскопа та акселерометра виготовлені як MEMS (мікроелектромеханічна система) - зовнішній вплив на сенсор спочатку змінює стан механічної частини, що потім призводить до зміни сигналу електричної частини. Отже, в одному корпусі зібрані не лише електронні, але й механічні компоненти. У мікросхемі MPU9250 міститься одразу три MEMS-сенсори, і виробник запевняє, що їхній взаємний вплив зведено до мінімуму.

Розглянемо, як можна використовувати сенсори акселерометра і гіроскопа. Температурний сенсор не будемо чіпати. Гіроскоп видає значення миттєвої кутової швидкості з дозволом, заданим в налаштуваннях, наприклад 2000 градусів на секунду. Якщо прошити мікроконтролер і спостерігати за цими даними, то спочатку побачимо лише нулі. При початку обертання сенсора

отримаємо миттєві значення кутової швидкості. Зверніть увагу, що швидкість ми отримуємо в градусах за секунду, а це означає, що лінійні швидкості не впливають на ці свідчення - значення змінюватимуться лише при обертанні сенсора у просторі. Використовуючи ці дані, можна визначити орієнтацію об'єкта у просторі. Для цього потрібно отримати миттєве значення кутової швидкості і помножити його на проміжок часу між опитуваннями сенсора. Наприклад, при здатності 2000 градусів на секунду, інтервал між опитуваннями 0,1 секунда, значення миттєвої швидкості 300, це дає $300 * 0,1 = 30$ - за цей час вісь гіроскопа була повернута на 30 градусів. Далі кожне отримане значення потрібно додавати до попереднього. Якщо вісь рухалася в одному напрямку - значення 30 градусів, якщо в іншому то -30, таким чином, при поверненні сенсора в початкове положення завжди (в ідеалі) буде 0, а при відхиленні від початкового положення, при виконанні вищезазначених дій, отримаємо кут відхилення. Обробляючи кути трьох осей гіроскопа, можна визначити орієнтацію об'єкта в просторі.

Таким чином, при інтегруванні стану кута положення, також інтегрується і похибка - при тривалому використанні можна отримати зовсім неправильні значення. Тому часто гіроскоп використовують у парі з акселерометром, утворюючи найпростішу версію альфа-бета фільтра або комплементарного фільтра.

З акселерометром все простіше. Вимірюючи прискорення трьох осей сенсора, можна отримати дані, за допомогою яких, з використанням геометрії, також можна визначити орієнтацію об'єкта в просторі. Крім того, акселерометр вимірює лінійні прискорення, тобто орієнтація об'єкта може спотворюватися при русі сенсора у лінійних напрямках. За допомогою акселерометра також можна визначати рух об'єкта або його зіткнення. Наприклад, виявляти падіння об'єкта або удар об перешкоду, щоб обійти їх.

Дані від акселерометра завжди досить точні, тобто нуль завжди залишається нулем незалежно від впливів (це означає, що дані не залежать ні від

часу, ні від характеру впливу), проте недолік у тому, що дані можуть йти із шумом в деякому діапазоні, тобто до десятих часток градуса точно виміряти кут не вдасться. Однак, виходячи з експериментальних даних, точність до цілих значень градуса зберігається цілком стабільно. Не забуваємо про вплив лінійних прискорень.

Якщо сенсор вже придбаний, можна переходити до вивчення внутрішнього устрою модуля, а саме його головного елемента - мікросхеми MPU9250. Інформація зберігається в регістрах мікросхеми, яких понад 100. І тут виникає значна проблема. Виробник не надав у документації всієї необхідної інформації, а лише найнеобхідніше. Насправді, невідомо навіть точну кількість регістрів, доступних для читання або запису, або того й іншого. Інформація про деякі регістри відсутня, окрім їхніх назв. Тому доведеться експериментально визначати вплив значень, записаних у ці регістри.

Інтерфейс I2C працює за стандартною схемою. Адреса мікросхеми може мати два значення (без біта читання/запису) в залежності від стану виведення AD0: b1101000, якщо AD0 з'єднаний із землею, і b1101001, якщо AD0 з'єднаний з джерелом живлення, плюс біт читання або запису.

Мікросхема містить Digital Motion Processor (DMP), який необхідний для обробки даних, отриманих з датчиків гіроскопа та акселерометра. Це робиться для підвищення точності отриманих даних, оскільки при обробці на мікроконтролері точність може знижуватися через низьку швидкість обробки. Зазвичай алгоритми обробки руху повинні працювати з досить високою частотою, приблизно 200 Гц, як зазначено в документації.

Щодо регістрів, їх досить багато, і необхідна інформація знаходиться в карті регістрів на MPU9250.

3.4 Вибір контролера

Як керуючий контролер обрано плату Arduino Pro Micro Leonardo, загальний вигляд якої наведено на рис. 3.5.

Контролер Arduino Pro Micro Leonardo спроектований на базі мікроконтролера ATmega32U4. Платформа має 12 цифрових вхід/виходів (5 з яких можуть використовуватися як виходи ШІМ), 4 аналогових входи, кварцовий генератор на 16 МГц та роз'єм Mini-USB. Для роботи необхідно підключити платформу до комп'ютера через USB-кабель.

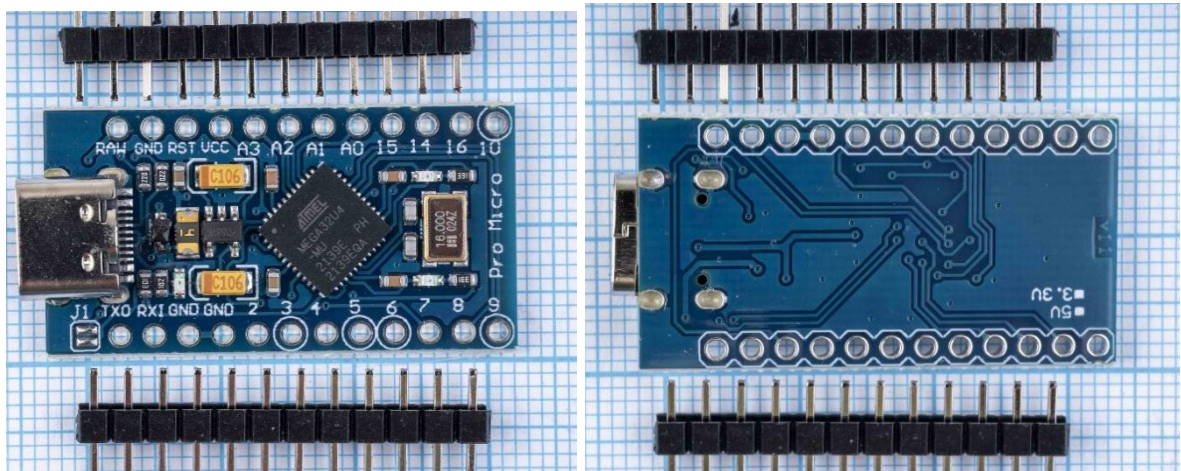


Рис. 3.5 Загальний вигляд плати мікроконтролера Arduino Pro Micro Leonardo

Arduino Pro Micro Leonardo може отримувати живлення через підключення USB. Джерело живлення вибирається автоматично.

Таблиця 3.1. Характеристики контролера Arduino Uno R3

Мікроконтролер	ATmega32U4
Робоча напруга	5 В
Вхідна напруга (рекомендована)	7-12 В
Вхідна напруга (гранична)	6-20 В

Цифрові Входи/Виходи	12 (5 з яких можуть використовуватись як виходи ШИМ)
Аналогові входи	4
Постійний струм через вхід/вихід	40 мА
Постійний струм для виводу 3.3 В	50 мА
Флеш-пам'ять	32 Кб (ATmega32U4) з яких 4 Кб використовуються для завантажувача
ОЗУ	2.5 Кб (ATmega32U4)
EEPROM	1 Кб (ATmega32U4)
Тактова частота	16 МГц

Платформа може функціонувати при зовнішньому живленні в діапазоні від 6 до 20 В. При напрузі живлення менше 7 В, висновок 5V може видавати менше 5 В, що може призвести до нестабільної роботи платформи. Якщо використовувати напругу вище 12 В, регулятор напруги може перегрітися і пошкодити плату. Рекомендований діапазон живлення – від 7 до 12 В.

Виводи живлення:

- VIN: Вхід для подачі живлення від зовнішнього джерела (без 5 В від роз'єму USB або іншого регульованого джерела живлення). Напруга живлення подається через цей вивід.
- 5V. Регульоване джерело напруги для живлення мікроконтролера та компонентів на платі. Живлення може подаватися від виводу VIN через регулятор напруги, або від роз'єму USB, або іншого регульованого джерела напруги 5 В.
- 3V3. Напруга на виведенні 3.3 В генерується вбудованим регулятором на платі. Максимальне споживання струму 50 мА.
- GND. Виводи заземлення.

Мікроконтролер ATmega32U4 має 32 кБ флеш пам'яті, з яких 4 кБ використовується для зберігання завантажувача, а також 2.5 кБ ОЗУ (SRAM) та 1 Кб EEPROM (яка читається та записується за допомогою бібліотеки EEPROM).

Кожен з 12 цифрових висновків Pro Micro може бути налаштований як вхід або вихід, використовуючи функції `pinMode()`, `digitalWrite()`, і `digitalRead()`. Виводи працюють при напрузі 5 В. Кожен вивід має резистор навантаження (за замовчуванням відключений) 20-50 кОм і може пропускати до 40 мА. Деякі виводи мають спеціальні функції:

Послідовна шина: 0 (RX) та 1 (TX). Виводи використовуються для отримання (RX) та передачі (TX) даних TTL. Ці виводи підключені до відповідних виводів мікросхеми послідовної шини ATmega32U4 USB-to-TTL.

- Зовнішнє переривання: 2 і 3. Ці виводи можуть бути налаштовані на виклик переривання на молодшому значенні, передньому або задньому фронті, або при зміні значення. Детальна інформація міститься в описі функції `attachInterrupt()`.

- ШІМ: 3, 5, 6, 9, 10 і 11. Будь-який з цих виводів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції `analogWrite()`.

- SPI: 10 (SS), 16 (MOSI), 14 (MISO), 15 (SCK). За допомогою цих виводів здійснюється зв'язок SPI, для чого використовується бібліотека SPI.

- LED: 13. Вбудований світлодіод підключений до цифрового виводу 13. Якщо значення виводу має високий потенціал, то світлодіод горить.

На платформі Pro Micro встановлено 4 аналогові входи (позначені як A0..A3), кожен із роздільною здатністю 10 біт (тобто може набувати 1024 різних значень). Стандартно виводи мають діапазон вимірювання до 5 В щодо землі, проте є можливість змінити верхню межу за допомогою виведення AREF та функції `analogReference()`. Деякі виводи мають додаткові функції:

- I2C: 2 (SDA) та 3 (SCL). За допомогою цих виводів здійснюється зв'язок I2C (TWI), для створення якої використовується бібліотека Wire.

- Додаткова пара висновків платформи:

- AREF. Опорна напруга аналогових входів. Використовується з функцією analogReference().

- Reset. Низький рівень сигналу на виводі перезавантажує мікроконтролер.

На платформі Arduino Pro Micro Leonardo встановлено кілька пристроїв для зв'язку з комп'ютером, іншими пристроями Arduino або мікроконтролерами. ATmega32U4 підтримує послідовний інтерфейс UART TTL (5), здійснюваний виводами 0 (RX) і 1 (TX). Встановлена на платі мікросхема ATmega32U4 спрямовує цей інтерфейс через USB, програми на боці комп'ютера "спілкуються" з платою через віртуальний порт COM. Прошивка ATmega32U4 використовує стандартні драйвери USB COM, ніяких сторонніх драйверів не потрібно, але на Windows для підключення знадобиться файл Arduino.inf. Монітор послідовної шини (Serial Monitor) Arduino дозволяє надсилати і отримувати текстові дані при підключенні до платформи. Світлодіоди RX і TX на платформі будуть блимати під час передачі даних через мікросхему FTDI або USB підключення (але не при використанні послідовної передачі через виводи 0 і 1).

Бібліотека SoftwareSerial дозволяє створити послідовну передачу даних через будь-який із цифрових виводів Pro Micro Leonardo.

ATmega32U4 підтримує інтерфейси I2C (TWI) та SPI. В Arduino включено бібліотеку Wire для зручності використання шини I2C.

Платформа програмується за допомогою програмного забезпечення Arduino. З меню Tools > Board вибирається «Arduino Leonardo» (відповідно до встановленого мікроконтролера).

Мікроконтролер ATmega32U4 поставляється із записаним завантажувачем, що полегшує запис нових програм без використання зовнішніх програматорів. Зв'язок здійснюється оригінальним протоколом STK500.

Pro Micro розроблено таким чином, щоб перед записом нового коду перезавантаження здійснювалося самою програмою Arduino на комп'ютері, а не натисканням кнопки на платформі. Одна з ліній DTR мікросхеми ATmega32U4, що управляють потоком даних (DTR), підключена до виведення

					ХНТУ 151.КРБ.24.02 ПЗ	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		63

перезавантаження мікроконтролера ATmega32U4 через 100 нФ конденсатор. Активація цієї лінії, тобто подача сигналу низького рівня, перезавантажує мікроконтролер. Програма Arduino, використовуючи цю функцію, завантажує код одним натисканням кнопки Upload у середовищі програмування. Подача сигналу низького рівня по лінії DTR скоординована з початком запису коду, що скорочує тайм-аут завантажувача.

Функція має ще одне застосування. Перезавантаження Pro Micro відбувається щоразу при підключенні до Arduino на комп'ютері з ОС Mac X або Linux (через USB). Наступні півсекунди після перезавантаження працює завантажувач. Під час програмування відбувається затримка кількох перших байтів коду, щоб уникнути отримання платформою некоректних даних (всіх, крім коду нової програми). Якщо на комп'ютері налаштовується разове налагодження скетчу, записаного в платформу, або введення будь-яких інших даних при першому запуску, переконайтеся, що програма на комп'ютері очікує протягом секунди перед передачею даних.

На Pro Micro можна відключити лінію автоматичного перезавантаження розривом відповідної лінії. Контакти мікросхеми з обох кінців лінії можуть бути з'єднані з метою відновлення. Лінія маркована "RESET-EN". Вимкнути автоматичне перезавантаження також можливо підключивши резистор 110 Ом між джерелом 5 В і даною лінією.

В Arduino Pro Micro вбудований запобіжник (автомат), що самозупиняється, захищаючи порт USB комп'ютера від струмів короткого замикання і надструмів. Хоча практично всі комп'ютери мають подібний захист, проте цей запобіжник забезпечує додатковий бар'єр. Запобіжник спрацьовує при проходженні струму більше 500 мА через USB порт і розмикає ланцюг до тих пір, поки нормальні значення струмів не будуть відновлені.

Довжина та ширина друкованої плати Pro Micro становлять 4.5 см та 2 см відповідно. Роз'єм USB виходить за межі даних розмірів. Чотири отвори у платі

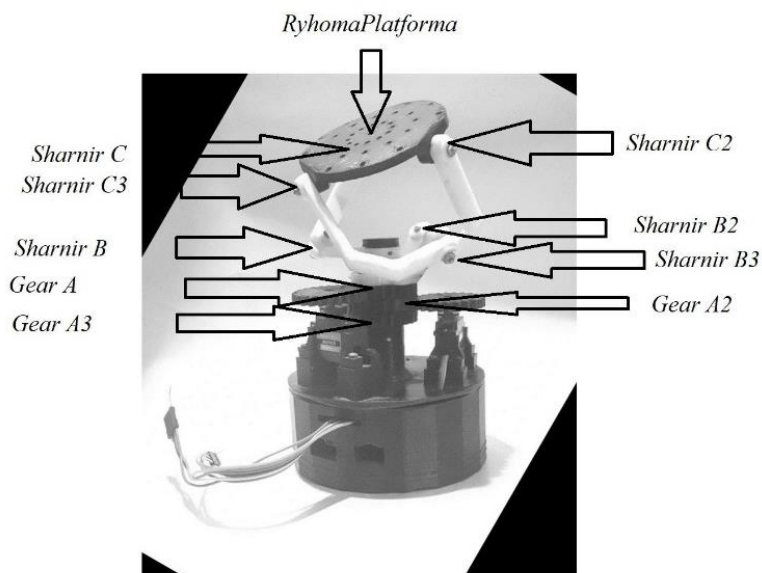
дозволяють закріпити її на поверхні. Відстань між цифровими виводами 7 та 8 дорівнює 0,4 см, хоча між іншими виводами воно становить 0,254 см.

3.5 СПМ

Огляд будови сферичного механізму з позиціонуючою платформою

Сферичний механізм має наступні характерні елементи, що наведені нижче:

1. RyhomaPlatforma – центр платформи;
2. Sharnir C, Sharnir C2, Sharnir C3 – шарніри з'єднання платформи та верхніх ланок;
3. Sharnir B, Sharnir B2, Sharnir B3 – шарніри з'єднання верхніх та нижніх ланок;
4. Gear A, Gear A2, Gear A3 – шестерні, значення яких отримуються для керування дослідним зразком.



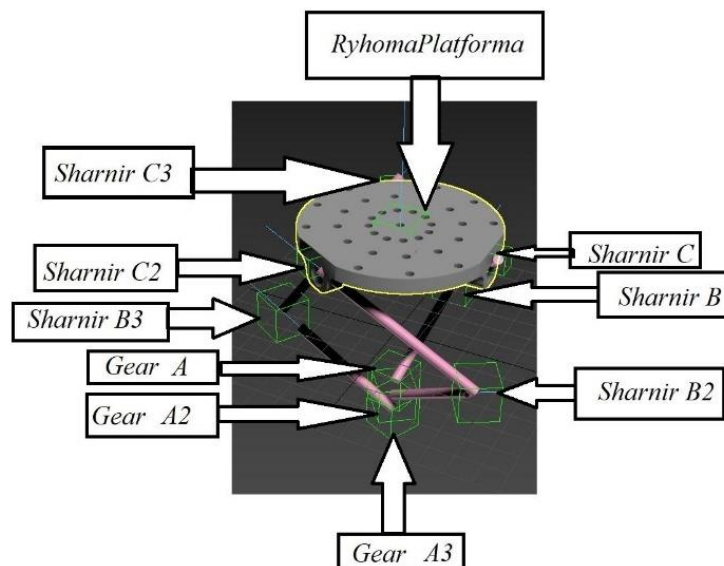
а)

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 151.КРБ.24.02 ПЗ

Арк.

65



б)

Рис. 3.6. Структура сферичного механізму з позиціонуючою платформою: а) – виготовлений дослідний зразок на 3д принтері; б) – ієрархічна тривимірна кінематична модель

3.6 Програма контролера для вимірювання кутів та керування СПМ

3.6.1 Склад бібліотек до програми

Програма керування наведена у додатку А. В ній використовуються такі бібліотеки як:

- Servo.h – дана бібліотека призначена для керування сервоприводами;
- I2Cdev.h - Ця бібліотека забезпечує низькорівневий доступ до пристроїв через шину I2C (Inter-Integrated Circuit). Вона дозволяє здійснювати читання і запис до пристроїв, підключених через I2C, таких як датчики, EEPROM і інші периферійні пристрої.;
- Wire.h – Ця бібліотека використовується для роботи з шиною I2C на мікроконтролерах Arduino. Вона забезпечує інтерфейс для взаємодії з пристроями, підключеними до шини I2C, включаючи читання і запис даних;

- MPU6050_6Axis_MotionApps612.h – Ця бібліотека дозволяє використовувати датчик руху MPU6050 (а також 9250[14]), який включає гіроскоп і акселерометр.
- EEPROM.h – Ця бібліотека надає функції для роботи з вбудованою EEPROM на мікроконтролерах Arduino.

3.6.2 Алгоритм програми для вимірювання кутів та керування СПМ

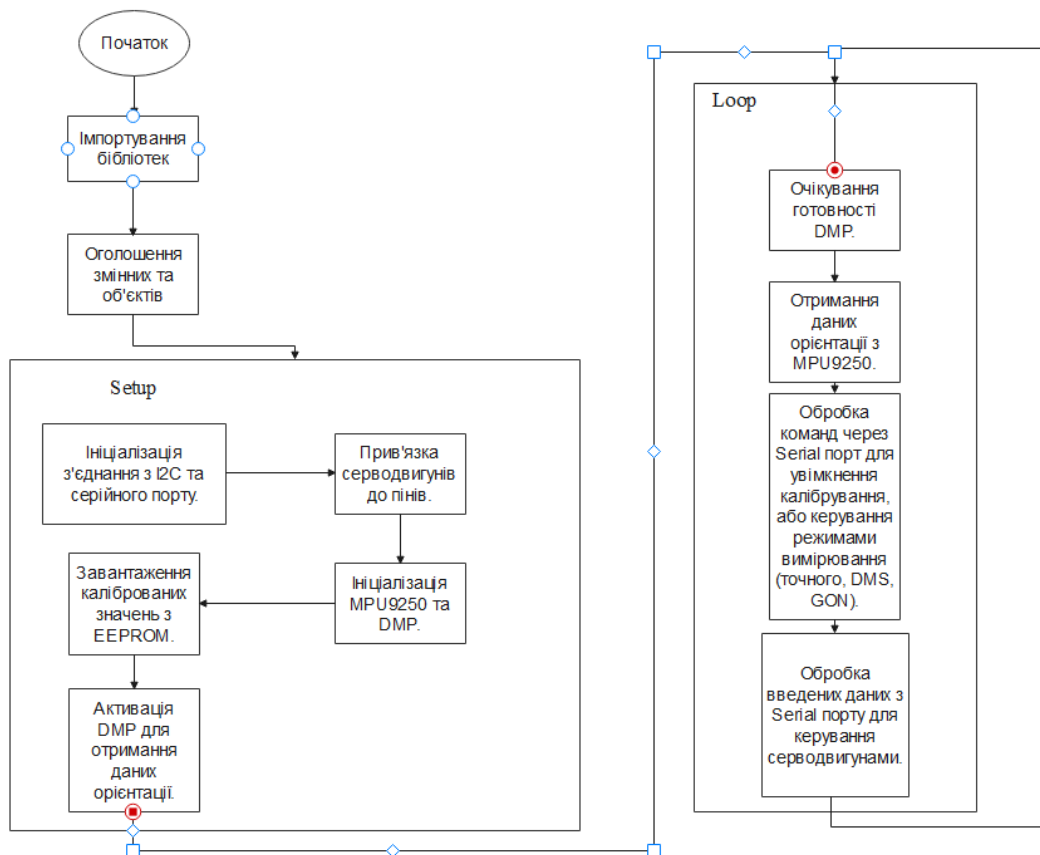


Рис. 3.7. Алгоритм програми для мікроконтролера

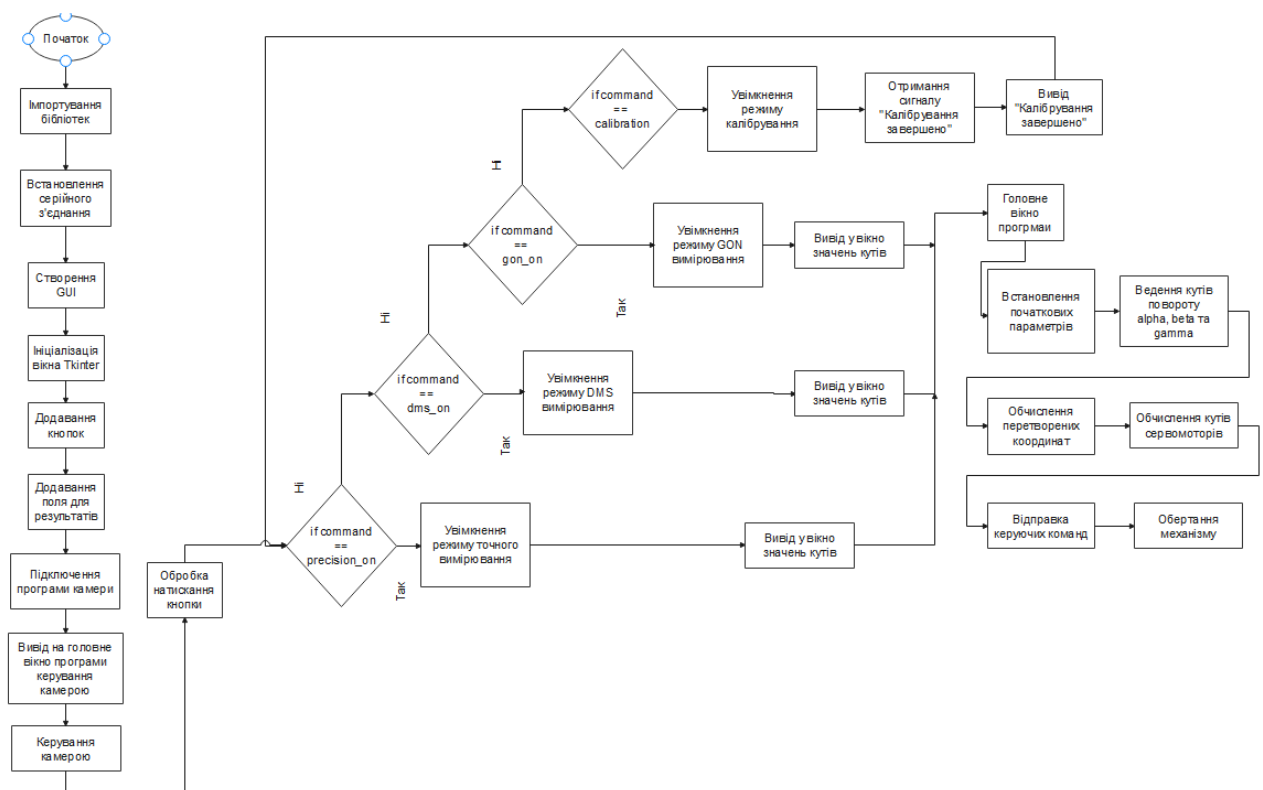


Рис. 3.7. Алгоритм програми для керування СПМ та вимірювання кутів

ВИСНОВОКИ

У ході цієї роботи було досліджено методи вимірювання кутів у будівництві, зокрема, оптичні та електронні теодоліти. Цей огляд літератури дозволив отримати глибоке розуміння принципів їх роботи, переваг та обмежень.

Після аналізу існуючих рішень було виявлено, що традиційні оптичні теодоліти мають високу точність, особливо при вимірюванні на великі відстані, проте вони можуть бути важкими у використанні та вимагати поправок на атмосферні умови.

З іншого боку, електронні теодоліти, зокрема ті, які використовують ІМУ, демонструють нові можливості у вимірюванні кутів. Вони можуть бути компактними, легкими у використанні та можуть надавати широкий спектр додаткових функцій, таких як зберігання даних та автоматизовані вимірювання.

Була детально розглянута структура кватерніонів, їхні складові та основні характеристики, такі як норма, модуль та аргумент. Використовуючи ці характеристики, ми розробили математичні формули для визначення кутів у трьох вимірах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Повірка та призначення нівелірів та теодолітів* [Електронний ресурс]. – Режим доступу: <https://www.ifdcsm.com.ua/en/news/267/povirka-ta-pryznachennia-niveliriv-ta-teo>
2. Літнарів Р.М. Геодезична астрономія. Навчальний посібник для студентів спеціальності “Землепорядкування та кадастр”. ЧДІЕіУ, Чернігів, 2000. – 76 с.
3. Методичні вказівки до виконання практичних робіт з дисципліни «Метрологія, стандартизація та сертифікація в землеустрої» для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 193 «Геодезія та землеустрій» денної, заочної та дистанційної форм навчання / Укладачі: Ландо Є.О., Кочан С.М. – Дніпро: ДВНЗ ПДАБА, 2022. – 40с.
4. Топографія. Лабораторний практикум / Уклад.: І.В. Калинич, М.Р. Ничвид, І.І. Калинич. Ужгород: Видавництво УжНУ «Говерла», 2020. 176 с.
5. *Що таке оптичний теодоліт* [Електронний ресурс]. – Режим доступу: <https://geomarket.in.ua/ua/teodolity/opticheskie-teodolity/>
6. *Що таке електронний теодоліт* [Електронний ресурс]. – Режим доступу: <https://geomarket.in.ua/ua/teodolity/elektronnye-teodolity/>
7. Демченко А.Г. Теодоліти та їх особливості // Матеріали Студентської наукової конференції Одеського державного екологічного університету - 2023, 10 - 17 травня. Одеса: ОДЕКУ. 2023. С. 124 -126
8. *Electronic Theodolites Nivel System DT-2, DT-5: manual* [Електронний ресурс]. – Режим читання: [https://www.gosafe.ee/media/inten/general/Tootelehed/Nivel_System_DT-2,_DT-5_\(wer_2\)_-2_-manual_\(EN\).pdf](https://www.gosafe.ee/media/inten/general/Tootelehed/Nivel_System_DT-2,_DT-5_(wer_2)_-2_-manual_(EN).pdf)
9. Теодоліт, як основний кутомірний прилад. Реферат / Уклад.: Н.П. Фель, А. Форманюк. Володимир – Волинський, 2011. 10 с.

					ХНТУ 151.КРБ.24.02 ПЗ	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		70

10. *Теодоліти* [Електронний ресурс]. – Режим доступу:
<https://geosklad.com.ua/menus/view/teodoliti>
11. *Аналітична теорія кватерніонних функцій. Дипломна робота* [Електронний ресурс]. – Режим доступу: <http://eprints.zu.edu.ua/11941/1/Vygivska.pdf>
12. *Використання кватерніону для виконання тривимірних обертань* [Електронний ресурс]. – Режим доступу:
<https://www.cprogramming.com/tutorial/3d/quaternions.html>
13. Деркаченко А.О., Поливода О.В. Математична модель руху елементів каркасних конструкцій // Матеріали Студентської наукової конференції Херсонського національного технічного університету - 2023, 10 - 17 травня. Херсон: ХНТУ. 2024. С. 124 -126
14. *MPU6050 connection failed code 1 initial memory load failed* [Електронний ресурс]. – Режим доступу: <https://github.com/jrowberg/i2cdevlib/issues/617>

ДОДАТКИ

```
#include "I2Cdev.h"
#include "Wire.h"
#include "MPU6050_6Axis_MotionApps612.h" // Потрібно включити для DMP,
містить прошивку hex для завантаження в MPU при ініціалізації
#include <EEPROM.h>
#include <Servo.h>

// Адреса MPU для I2C
byte devAddr = 0x68;
MPU6050 mpu(devAddr);

Servo servo1; // Серводвигун 1
Servo servo2; // Серводвигун 2
Servo servo3; // Серводвигун 3

// Змінні керування/статусу MPU від JRowberg
bool dmpReady = false; // встановлюється true, якщо ініціалізація DMP успішна
byte mpuIntStatus;      // містить фактичний статус переривання від MPU
byte devStatus;         // статус повернення після кожної операції з пристроєм (0 =
успіх, !0 = помилка)
int packetSize;         // очікуваний розмір пакету DMP (за замовчуванням 42 байти)
int fifoCount;          // кількість всіх байтів у FIFO
byte fifoBuffer[64];    // буфер збереження FIFO
Quaternion q;           // [w, x, y, z] контейнер для кватерніона

float sensorData[3];    // Оголошення масиву для збереження даних
static byte index = 0;

bool precisionMode = false;
bool dmsMode = false;
bool gonMode = false;

VectorFloat QtoEulerAngle(Quaternion qt) {
    VectorFloat ret;
    double sqw = qt.w * qt.w;
    double sqx = qt.x * qt.x;
    double sqy = qt.y * qt.y;
    double sqz = qt.z * qt.z;

    ret.x = atan2(2.0 * (qt.x * qt.y + qt.z * qt.w), (sqx - sqy - sqz + sqw));
    ret.y = asin(2.0 * (qt.x * qt.z - qt.y * qt.w) / (sqx + sqy + sqz + sqw));
    ret.z = atan2(2.0 * (qt.y * qt.z + qt.x * qt.w), (-sqx - sqy + sqz + sqw));
}
```

```
// Додаємо перетворення радіан у градуси
ret.x = ret.x * 180 / PI;
ret.y = ret.y * 180 / PI;
ret.z = ret.z * 180 / PI;
return ret;
}
```

```
// Записати 2-байтове слово, починаючи з наданої адреси
void eepromWriteWord(int startAddr, int value) {
    EEPROM.update(startAddr, value);
    EEPROM.update(startAddr + 1, value >> 8);
}
```

```
// Повернути 2-байтове слово, прочитане з початкової адреси EEPROM
int eepromReadWord(int startAddr) {
    int value = EEPROM.read(startAddr);
    value = value | (EEPROM.read(startAddr + 1) << 8);
    return value;
}
```

```
void getCalibration() {
    // Отримати збережені калібрувальні значення з EEPROM і оновити MPU
    // Адреса завжди починається з 0 і закінчується на 11, 2 байти на вісь

    // Перевірити, чи є щось збережене (шукаємо всі FF)
    // якщо ні, припускаємо, що калібрування не виконувалось і пропускаємо
    налаштування за замовчуванням
    // Зрештою підкажемо користувачу виконати калібрування! якщо в пам'яті скетчу
    залишиться достатньо місця!
    mpu.setXAccelOffset(eepromReadWord(0));
    mpu.setYAccelOffset(eepromReadWord(2));
    mpu.setZAccelOffset(eepromReadWord(4));
    mpu.setXGyroOffset(eepromReadWord(6));
    mpu.setYGyroOffset(eepromReadWord(8));
    mpu.setZGyroOffset(eepromReadWord(10)); // остання адреса для читання буде 11
    десяткова в EEPROM
}
```

```
void setCalibration() {
    // Виконати автокалібрування DMP, потім отримати ці значення і зберегти їх у
    EEPROM
    // Це слід викликати тільки при виборі опції Автокалібрування
    // для збереження ресурсу EEPROM використовувати update замість write
}
```

```
// Виконати автокалібрування 6 разів  
mpu.CalibrateAccel(6);  
mpu.CalibrateGyro(6);
```

```
// Отримати фінальні значення, які були збережені і перенести їх у EEPROM  
int Data[3];
```

```
// Зміщення акселерометра
```

```
I2Cdev::readWords(devAddr, 0x06, 3, (int *)Data);
```

```
epromWriteWord(0, Data[0]);
```

```
epromWriteWord(2, Data[1]);
```

```
epromWriteWord(4, Data[2]);
```

```
// Зміщення гіроскопа
```

```
I2Cdev::readWords(devAddr, 0x13, 3, (int *)Data);
```

```
epromWriteWord(6, Data[0]);
```

```
epromWriteWord(8, Data[1]);
```

```
epromWriteWord(10, Data[2]); // Останній байт записаний у адресу EEPROM 11  
десятькова
```

```
}
```

```
void degreesToDMS(float degrees) {
```

```
int d = (int)degrees; // Ціла частина градусів
```

```
float m_full = (degrees - d) * 60; // Хвилини з десятковою частиною
```

```
int m = (int)m_full; // Ціла частина хвилин
```

```
float s_full = (m_full - m) * 60; // Секунди з десятковою частиною
```

```
float s = round(s_full * 100) / 100.0; // Округлюємо секунди до 2 десяткових знаків
```

```
Serial.print(d);
```

```
Serial.print("° ");
```

```
Serial.print(m);
```

```
Serial.print("' ");
```

```
Serial.print(s);
```

```
Serial.print("''");
```

```
}
```

```
void degreesToGon(float degrees) {
```

```
float gon = degrees / 0.9;
```

```
Serial.print(gon);
```

```
Serial.print(" gon");
```

```
}
```

```
void resetModes() {
```

```
precisionMode = false;
```

```
dmsMode = false;
```

```
gonMode = false;
```

```

}

void setup() {
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
    Wire.setClock(400000); // 400kHz I2C годинник.
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif

  Serial.begin(9600);

  // Прив'язуємо серводвигуни до відповідних пінів'
  servo1.attach(4); // Пін 4 для серводвигуна 1
  servo2.attach(5); // Пін 5 для серводвигуна 2
  servo3.attach(6); // Пін 6 для серводвигуна 3

  // Ініціалізація MPU9250
  mpu.initialize();
  devStatus = mpu.dmpInitialize();

  // Отримати збережені значення калібрування з EEPROM і відправити їх до MPU
  // Інакше за замовчуванням відобразити необхідність калібрування!
  getCalibration();

  // Переконалися, що все спрацювало - оскільки ми завантажуюмо прошивку при
  запуску DMP,
  // потрібно перевірити, що все дійсно пройшло за планом, devStatus 0 - успіх.
  if (devStatus == 0) {
    mpu.setDMPEntered(true);
    mpuIntStatus = mpu.getIntStatus();
    // встановлюємо прапорець готовності DMP, щоб функція loop() знала, що можна
    використовувати його
    dmpReady = true;

    // отримаємо очікуваний розмір пакету DMP для подальшого порівняння
    packetSize = mpu.dmpGetFIFOPacketSize();
  }
}

// ГОЛОВНИЙ ЦИКЛ ПРОГРАМИ!!
void loop() {
  char input;

```

```

if (!dmpReady) return;
// Отримуємо значення кватерніона з буфера DMP
if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
    mpu.dmpGetQuaternion(&q, fifoBuffer);

    // Розраховуємо кути, перетворюючи кватерніон у Ейлерові кути, це давало більш
стабільні та точні результати порівняно з
    // отриманням Ейлерових кутів безпосередньо з DMP. Думаю, перетворення
кватерніонів враховує гімбальний замок.
    VectorFloat ea = QtoEulerAngle(q);
    if (Serial.available() > 0) {
        String command = Serial.readStringUntil('\n'); // Читання команди з серійного
порту

        if (command == "calibration") {
            setCalibration();
            Serial.println("Калібрування завершено");
        } // Обробка команд ввімкнення режимів вимірювання
        else if (command == "precision_on") {
            resetModes();
            precisionMode = true;
            Serial.println("Режим точного вимірювання увімкнено");
            Serial.print(ea.x);
            Serial.print("° ");
            Serial.print(ea.y);
            Serial.print("° ");
            Serial.println(ea.z);
            Serial.print("°");
            // Почати відображення точних вимірювань негайно
        } else if (command == "dms_on") {
            resetModes();
            dmsMode = true;
            Serial.println("Режим DMS вимірювання увімкнено");
            degreesToDMS(ea.x);
            degreesToDMS(ea.y);
            degreesToDMS(ea.z);
            // Почати відображення вимірювань у форматі DMS негайно
        } else if (command == "gon_on") {
            resetModes();
            gonMode = true;
            Serial.println("Режим GON вимірювання увімкнено");
            degreesToGon(ea.x);
            degreesToGon(ea.y);
            degreesToGon(ea.z);
        }
    }
}

```

```

    // Почати відображення вимірювань у форматі гон нехайно
    }
}

// Перевіряємо, чи є дані доступні для читання з Serial порту
if (Serial.available() > 0) {
    input = Serial.read(); // Читаємо один символ з Serial порту

    // Перевіряємо, чи отриманий символ не є символом нового рядка ('\n')
    if (input != '\n') {
        // Якщо символ не є новим рядком, то конвертуємо його в число
        // та додаємо до поточного значення в масиві sensorData[index]
        sensorData[index] = sensorData[index] * 10 + (input - '0');
    } else {
        // Якщо отриманий символ - це новий рядок, це означає, що ми отримали
        // повний набір даних для одного зі значень масиву sensorData

        index++; // Збільшуємо індекс для переходу до наступного значення

        // Якщо ми отримали всі три значення масиву sensorData
        if (index >= 3) {
            // Зчитуємо отримані значення
            float A = sensorData[0];
            float A2 = sensorData[1];
            float A3 = sensorData[2];

            // Налаштовуємо позиції сервоприводів
            servo1.write(A);
            servo2.write(A2);
            servo3.write(A3);

            index = 0; // Скидаємо індекс для наступного набору даних
        }
    }
}
}

```

```
import tkinter as tk
import serial
import threading

# Ініціалізація з'єднання з Arduino через Serial порт
arduino = serial.Serial('COM3', 9600, timeout=1)

# Функція для відправки команди на Arduino
def send_command(command):
    arduino.write(command.encode())

# Функція для калібрування
def set_calibration():
    send_command("calibration")
    check_serial() # Після відправки команди перевіряємо Serial порт

# Функція для перевірки відповіді з Arduino через Serial порт
def check_serial():
    def read_from_port():
        while True:
            if arduino.in_waiting > 0:
                line = arduino.readline().decode('utf-8').rstrip()
                print(f"Received: {line}") # Дебаг: виводимо отримане повідомлення
                if line == "Калібрування завершено":
                    calibration_finished()
                    break

    thread = threading.Thread(target=read_from_port)
    thread.start()

# Функція для увімкнення режиму точного вимірювання
def precision_on():
    send_command("precision_on")

# Функція для увімкнення режиму DMS вимірювання
def dms_on():
    send_command("dms_on")

# Функція для увімкнення режиму GON вимірювання
def gon_on():
    send_command("gon_on")

# Функція для обробки події натискання кнопки "Калібрування"
```

```
def button_calibration_click():
    set_calibration()

def button_precision_click():
    precision_on()

def button_dms_click():
    dms_on()

def button_gon_click():
    dms_on()

# Функція для завершення калібрування
def calibration_finished():
    result_label.config(text="Калібрування завершено!")

# Створення головного вікна Tkinter
root = tk.Tk()
root.title("Управління Arduino") # Заголовок вікна
root.geometry("400x300") # розміри вікна

# Кнопка для Precipsion
button_precision = tk.Button(root, text="Precipsion",
command=button_precision_click)
button_precision.pack()

# Кнопка для DMS
button_dms = tk.Button(root, text="DMS", command=button_dms_click)
button_dms.pack()

# Кнопка для GON
button_gon = tk.Button(root, text="GON", command=button_gon_click)
button_gon.pack()

# Кнопка для Calibration
button_calibration = tk.Button(root, text="Calibration",
command=button_calibration_click)
button_calibration.pack()

# Текстове поле для відображення результату
result_label = tk.Label(root, text="")
result_label.pack()

# Запуск головного циклу обробки подій
```

```
root.mainloop()
```

```
import numpy as np
```

```
def calculate_radius(Rc, z):  
    return np.sqrt(Rc**2 - z**2)
```

```
def calculate_F(R1, R2, x0, y0):  
    return -R1**4 * x0**2 - x0**6 - y0**4 * x0**2 - 2 * x0**4 * y0**2 + \  
        2 * x0**4 * R2**2 - x0**2 * R2**4 + 2 * x0**4 * R1**2 + \  
        2 * y0**2 * R2**2 * x0**2 + 2 * R1**2 * y0**2 * x0**2 + 2 * x0**2 * R1**2 * \  
R2**2
```

```
def calculate_B_positions(x0, y0, R1, R2, F):  
    Bx = (x0**4 + R1**2 * x0**2 - R2**2 * x0**2 + y0**2 * x0**2 - np.sqrt(F) * y0) / (2  
* x0 * (x0**2 + y0**2))  
    By = y0/2 + (R1**2 * y0 - R2**2 * y0 + np.sqrt(F)) / (2 * (x0**2 + y0**2))  
    return Bx, By
```

```
def calculate_angles(Bx, By, R1):  
    angle_x = 180 / np.pi * np.arccos(Bx / R1)  
    angle_y = 180 / np.pi * np.arccos(By / R1)  
    return angle_x, angle_y
```

```
def rotation_matrix_x(alpha):  
    return np.array([  
        [1, 0, 0],  
        [0, np.cos(alpha), -np.sin(alpha)],  
        [0, np.sin(alpha), np.cos(alpha)]  
    ])
```

```
def rotation_matrix_y(beta):  
    return np.array([  
        [np.cos(beta), 0, np.sin(beta)],  
        [0, 1, 0],  
        [np.sin(beta), 0, np.cos(beta)]  
    ])
```

```
def rotation_matrix_z(gamma):  
    return np.array([  
        [np.cos(gamma), -np.sin(gamma), 0],  
        [np.sin(gamma), np.cos(gamma), 0],  
        [0, 0, 1]
```

])

```
def transform_coordinates(alpha, beta, gamma, coords):  
    M = np.dot(np.dot(rotation_matrix_x(alpha), rotation_matrix_y(alpha)),  
rotation_matrix_z(gamma))  
    return np.dot(M, coords)
```

```
# Початкові дані
```

```
Rc = 5.0
```

```
R1 = 3.0
```

```
x0, y0, z0 = 1.0, 2.0, 3.0
```

```
x2, y2, z2 = 1.5, 2.5, 3.5
```

```
x3, y3, z3 = 2.0, 3.0, 4.0
```

```
# Обчислення радіусів
```

```
R2 = calculate_radius(Rc, z0)
```

```
R3 = calculate_radius(Rc, z2)
```

```
R4 = calculate_radius(Rc, z3)
```

```
# Обчислення значення F
```

```
F = calculate_F(R1, R2, x0, y0)
```

```
# Обчислення положення точок B
```

```
Bx, By = calculate_B_positions(x0, y0, R1, R2)
```

```
B2x, B2y = calculate_B_positions(x2, y2, R1, R3)
```

```
B3x, B3y = calculate_B_positions(x3, y3, R1, R4)
```

```
# Обчислення кутів
```

```
A_x, A_y = calculate_angles(Bx, By, R1)
```

```
A2_x, A2_y = calculate_angles(B2x, B2y, R1)
```

```
A3_x, A3_y = calculate_angles(B3x, B3y, R1)
```

```
print(f"A_x: {A_x}, A_y: {A_y}")
```

```
print(f"A2_x: {A2_x}, A2_y: {A2_y}")
```

```
print(f"A3_x: {A3_x}, A3_y: {A3_y}")
```

```
# Обчислення матриць повороту
```

```
alpha, beta, gamma = np.radians(30), np.radians(45), np.radians(60)
```

```
coords = np.array([x0, y0, z0])
```

```
transformed_coords = transform_coordinates(alpha, beta, gamma, coords)
```

```
print(f"Transformed coordinates: {transformed_coords}")
```