

Міністерство освіти і науки України
Український державний університет науки і технологій

Навчально-науковий інститут
«Український державний хіміко-технологічний університет»
(назва навчально-наукового інституту)

Факультет комп'ютерних наук та інженерії
(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій та автоматизації
(повна назва кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр
(освітній рівень)

на тему Розробка та реалізація самобалансованого робота на колесах
з дистанційним керуванням

Виконав: студент 4 курсу, групи 4-АВП-22
спеціальності

151 Автоматизація та комп'ютерно -
інтегровані технології
(код і назва спеціальності)

КРЯЖКОВ Д. О.

(прізвище та ініціали)

(підпис)

Керівник ЛОСІХІН Д. А.

(прізвище та ініціали)

(підпис)

Рецензент МИСОВ О. П.

(прізвище та ініціали)

(підпис)

Дніпро – 2026 рік

Міністерство освіти і науки України
Український державний університет науки і технологій
(повне найменування вищого навчального закладу)

Навчально-науковий інститут
«Український державний хіміко-технологічний університет»
(назва навчально-наукового інституту)

Факультет, відділення Комп'ютерних наук та інженерії

Кафедра Комп'ютерно-інтегрованих технологій та робототехніки

Освітньо-кваліфікаційний рівень Бакалавр

Галузь знань 15 "Автоматизація та приладобудування"
(шифр і назва)

Спеціальність 151 "Автоматизація та комп'ютерно-інтегровані технології"
(шифр і назва)

Освітня програма Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри КІТтаА

Левчук І. Л.

“ ” 20 р.

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Кряжков Дмитро Олександрович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка та реалізація самобалансованого робота на колесах з дистанційним керуванням

керівник проекту (роботи) Лосіхін Дмитро Анатолійович, старший викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 02 ” березня 2026 р. № 77-к

2. Строк подання студентом проекту (роботи) 04.06.2026

3. Вихідні дані до проекту (роботи) Звіт з практики, літературні джерела, відповідні сторінки інтернет-сайтів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1 Вступ. 2 Апаратне забезпечення. 3 Математичне забезпечення.

4 Розробка технічної документації та реалізація системи. 5 Програмне забезпечення.

6 Охорона труда та техніка безпеки. 7 Економічні розрахунки. 8 Висновки.

Література. Додаток 1. Додаток 2. Додаток 3

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

Схема електрична принципова підключень компонентів

Схема електрична принципова з'єднань компонентів

Специфікація устаткування на засоби автоматизації

6. Консультанти розділів проекту (роботи)

Розділ	Ініціали, прізвище, та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона	Д. А. Лосіхін, старший викладач	20.03.2026	20.05.2026
праці			
Економічна	Д. А. Лосіхін, старший викладач	20.03.2026	01.06.2026
частина			

7. Дата видачі завдання “ 16 ” березня 20 26 р.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент _____
(підпис)

Д. О. Кряжков
(ініціали, прізвище)

Керівник проекту (роботи) _____
(підпис)

Д. А. Лосіхін
(ініціали, прізвище)

РЕФЕРАТ

Пояснювальна записка дипломного проекту містить: сторінок - 88; таблиць - 22; схем і рисунків - 16.

Відповідно до завдання до кваліфікаційної роботи у спеціальній частині проекту: здійснений вибір технічних засобів; розроблено монтажну плату для розміщення компонентів системи; розроблено таблиці з'єднань, схеми електричні принципиальні з'єднань та підключень компонентів системи; розглянуто математичне забезпечення інформаційних та керуючих каналів системи; розроблено програмне забезпечення системи.

У розділі “Охорона праці і техніка безпеки” виконаний опис заходів по створенню безпечних та здорових умов праці.

В економічній частині виконано розрахунок затрат на проектування та виготовлення прототипу системи, розрахунок точки беззбитковості. Після продажу мінімальної партії 210, шт., кожен наступний продаж приносить 1300 грн чистого прибутку.

Ключові слова: САМОБАЛАНСОВАНИЙ РОБОТ НА КОЛЕСАХ; РЕГУЛЮВАННЯ; ДИСТАНЦІЙНЕ КЕРУВАННЯ; МІКРОКОНТРОЛЕР; ПРОГРАМУВАННЯ C/C++.

				A26.13.APT.00.ПЗ			
				м. Дніпро, УДУНТ, ННІ УДХТУ			
Виконав	КРЯЖКОВ			Розробка та реалізація самобалансованого робота на колесах з дистанційним керуванням	Стадія	Лист	Листів
Перевірив	ЛОСІХІН				ДП	1	88
Рецензент	МИСОВ						
Н.Контр.				Пояснювальна записка	УДУНТ, ННІ УДХТУ, каф.КІТтаР, гр. 4-АВП-22		
Затвердив	ЛЕВЧУК						

ЗМІСТ

1 ВСТУП	4
2 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ	5
2.1 Механічна складова системи	5
2.2 Електричні приводи	6
2.3 Вимірювальні перетворювачі	9
2.4 Комунікаційні модулі	11
2.5 Підсистема живлення	13
2.6 Підсистема керування	15
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	17
3.1 Отримання інформації	17
3.1.1 Інерційні вимірювальні перетворювачі	17
3.1.2 Фільтрування сигналів	18
3.1.3 Електромеханічні вимірювальні перетворювачі	22
3.2 Управління	23
3.2.1 Регулювання кута нахилу платформи, що балансує.	23
3.2.2 Регулювання швидкості руху платформи	25
3.2.3 Управління напрямом руху платформи	26
3.2.4 Дистанційне керування рухом платформи	26
4 РОЗРОБКА ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ ТА РЕАЛІЗАЦІЯ СИСТЕМИ	28
4.1 Розробка та реалізація монтажною плати	28
4.2 Розробка монтажною схеми підключення	29
4.3 Розробка електричної принципової схеми підключення	37
4.4 Реалізація роботизованої балансууючої платформи	37
5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	39

5.1 Регулювання кута нахилу платформи, що балансує.	39
5.2 Регулювання швидкості руху платформи	40
5.3 Управління напрямом руху платформи	41
5.4 Дистанційне керування рухом платформи	42
5.5 Модуль керування балансуючої платформи	43
6 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ	45
6.1 Вимоги безпеки при проектуванні та складанні електронних модулів у домашніх умовах	45
6.2 Вимоги безпеки під час роботи з літій-іонними (li-ion) акумуляторами	48
6.3 Вимоги безпеки при першому увімкненні – подачі живлення на зібрану плату	50
7 ЕКОНОМІЧНІ РОЗРАХУНКИ	54
7.1 Оцінка витрат на проектування	54
7.2 Витрати виготовлення прототипу	56
7.3 Зведений кошторис проекту	59
7.4 Розрахунок точки беззбитковості	59
8 ВИСНОВКИ	63
ЛІТЕРАТУРА	64
ДОДАТОК 1 – Програма тестування інерційного модуля MPU-6050	66
ДОДАТОК 2 – Програма тестування каналу зв'язку через Bluetooth	68
ДОДАТОК 3 – Основна програма управління платформою балансування	71

1 ВСТУП

Роботизовані балансуєчі платформи - це динамічні стійкі мобільні роботи, які використовують алгоритми управління в реальному режимі часу для утримання вертикального положення на нестійкій основі (наприклад, на двох колесах або на одній кулі). Вони забезпечують високу маневреність у вузьких просторах і широко застосовуються у логістиці, дослідженнях та освіті.

В основі алгоритму управління платформою, що балансує, лежить принципи зворотного зв'язку. Датчики – акселерометр та гіроскоп вимірюють кут нахилу та кутову швидкість платформи. Дані надходять до мікроконтролера. За закладеним у алгоритмі закону регулювання обчислюється необхідна швидкість і напрямок обертання валів електродвигунів, компенсуючи відхилення платформи від вертикального становища.

Завданням на дипломну кваліфікаційну роботу є розробка та реалізація роботизованої двоколісної балансуєчої платформи з дистанційним керуванням, що дозволяє віддалено переміщувати та позиціонувати платформу у просторі на площині.

2 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

Апаратне забезпечення балансуючої двоколісної платформи представляє систему компонентів, які забезпечують балансування, переміщення та дистанційне управління. Основні компоненти системи: механічна складова, підсистема живлення, електронні модулі отримання інформації та керування, електричні приводи для реалізації впливів, що управляють, комунікаційні модулі для забезпечення дистанційного керування.

2.1 Механічна складова системи

Електронні, у тому числі комунікаційний модуль, та електричні приводи розміщені на корпусі платформи. Основою корпусу є акрилова пластина. Знизу до корпусу приєднані гвинтами М3 та гайками тримачі електроприводів та відсік для 3-х елементів живлення типу 18650, рис. 2.1, 2.2. Електронні модулі розташовані на розробленій монтажній платі, яка кріпиться гвинтами М3 та шестигранними бронзовими стійками до корпусу зверху. Для захисту монтажної плати від зовнішніх механічних пошкоджень над нею розташовується захисна акрилова пластина, яка кріпиться гвинтами М3 та шестигранними бронзовими стійками до корпусу.

Двигунами є звичайні пластмасові колеса із зовнішнім діаметром 68 мм, товщиною 26 мм із гумовим протектором, рис. 2.3.



Рисунок 2.1 – Тримачі електроприводів



Рисунок 2.2 – Відсік для 3-х елементів 18650



Рисунок 2.3 – Пластмасові колеса з гумовим протектором

2.2 Електричні приводи

Як приводи використано два редукторні електродвигуни постійного струму типу CHR-GM37-520 з датчиками Холла, які використовуються як інкрементний енкодер (тут, у проекті – датчик швидкості), рис. 2.4. Живлення електродвигуна 12 В. Передаточна кількість редуктора 1:30.

Маркування контактів для підключення двигуна та енкодера представлено на рис. 2.5 та у табл. 2.1.



Рисунок 2.4 – Редукторні електродвигуни постійного струму з енкодером

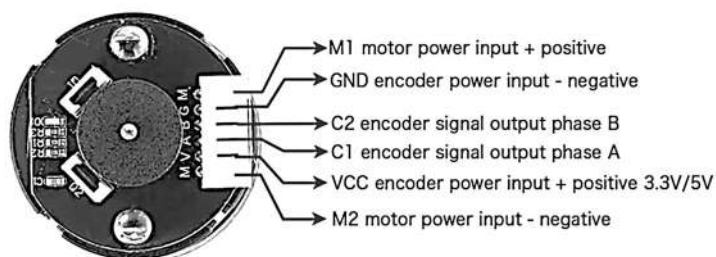


Рисунок 2.5 - Маркування контактів для підключення електродвигуна та енкодера

Таблиця 2.1 – Маркування контактів для підключення електродвигуна та енкодера

Маркування праворуч наліво	Позначення	Підключення
М	M1	Плюс, живлення електродвигуна, 12 В
G	GND	Земля/мінус енкодера
B	C2	Вихідний сигнал енкодера, канал В
A	C1	Вихідний сигнал енкодера, канал А
V	VCC	Плюс, харчування енкодера, 3.3/5 В
М	M2	Мінус, живлення електродвигуна

Для управління потужними ланцюгами живлення електродвигунів керуючими сигналами малопотужного контролера використовуються двоканальні драйвери постійного струму електродвигунів на чіпі TB6612FNG [1]. Основні технічні характеристики драйвера модуля на чіпі TB6612FNG представлені в табл. 2.2. Керуючі сигнали на кожен канал: два контакти для вибору напрямку обертання, один контакт для керування швидкістю обертання валу електродвигуна.

Таблиця 2.2 - Технічні характеристики модуля драйвера електродвигуна на чіпі TB6612FNG.

Напруга живлення	13.5 В
Вихідний струм	1.2 А (піковий 3.2 А)
Каналів управління	2
Функції	Напрямок обертання, короткий розрив, стоп.
Напруга логіки	2.7 ... 5.5 В

Максимальна частота ШІМ	100 кГц
Розмір модуля	20 x 20 мм
Особливості. Схема захисту від переполюсування живлення електродвигунів. Вбудований тепловий захист. Система енергозбереження	

Маркування контактів модуля драйвера електродвигунів постійного струму на чіпі TB6612FNG для підключення живлення, ланцюгів керування та потужних ланцюгів керування електродвигунами представлено на рис. 2.6 та у табл. 2.2.

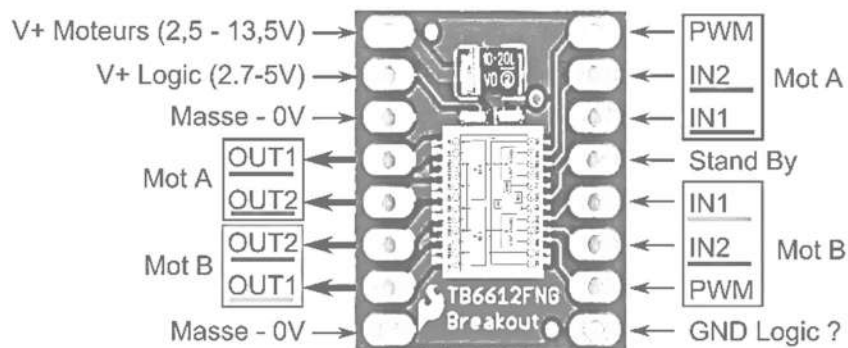


Рисунок 2.6 - Маркування контактів модуля драйвера електродвигунів постійного струму на чіпі TB6612FNG

Таблиця 2.3 – Маркування контактів модуля драйвера електродвигунів постійного струму на чіпі TB6612FNG для підключення ланцюгів живлення, керування та потужних ланцюгів керування

Позначення	Підключення
V+ Moteurs	Плюс, живлення електродвигунів, 2.5...13.5 В
V+ Logic	Плюс, харчування логіки, що управляє 2.7 ... 5 В
Masse – 0V	Земля
Mot A, OUT 1, OUT 2	Двигун А: вихід 1 (табл. 2.1, M1), вихід 2 (табл. 2.1, M2)

Mot B, OUT 1, OUT 2	Двигун В: вихід 1 (табл. 2.1, M1), вихід 2 (табл. 2.1, M2)
Masse – 0V	Земля
GND Logic	Земля
Mot B.PWM	Мотор В (тут, у проекті - Правий), сигнал ШІМ
Mot B.IN2	Мотор В. Вхід логіки керування 2
Mot B.IN1	Мотор В. Вхід логіки, що управляє 1
Stand By	Для активації чи пробудження модуля на вхід Stand By подається логічна 1.
Mot A.IN1	Мотор А. Вхід логіки, що управляє 1
Mot A.IN2	Мотор А. Вхід логіки, що управляє 2
Mot A.PWM	Мотор А (тут, у проекті – Лівий), сигнал ШІМ

2.3 Вимірювальні перетворювачі

Для визначення просторового положення платформи, що балансує, використовується популярний модуль MPU-6050 (GY-521), що поєднує 3-осьовий акселерометр, 3-осьовий гіроскоп і вбудований температурний датчик [2].

Модуль є мікроелектромеханічною системою. Аналоговий сигнал кожного датчика оцифровується 16-бітовим АЦП.

Акселерометр вимірює проекцію прискорення, що здається. У стані спокою він завжди показуватиме прискорення вільного падіння 1g, спрямоване до центру Землі. Недолік акселерометра: "шумить" при вібраціях і різких поштовхах.

Гіроскоп вимірює кутову швидкість обертання довкола власних осей. Щоб отримати поточний кут, мікроконтролеру потрібно постійно підсумовувати ці показники в часі. Недолік гіроскопа: згодом його показання

«дрейфують» (накопичується помилка, і кут починає мимоволі спливати, навіть якщо датчик нерухомий).

Для вирішення проблеми "шуму" і "дрейфу" в реальних проектах дані акселерометра та гіроскопа об'єднують за допомогою математичних фільтрів (компліментарний фільтр або фільтр Калмана). Також можна використовувати вбудований чіп DMP.

Маркування контактів модуля для підключення живлення, вихідних/вхідних інформаційних та сигнальних ліній представлено на рис. 2.7. та у табл. 2.4. Основні технічні характеристики модуля представлені у табл. 2.5.

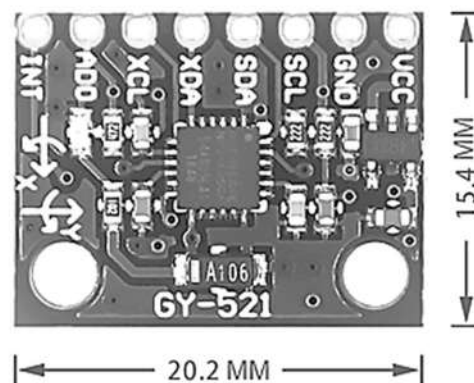


Рисунок 2.7 - Маркування контактів модуля MPU-6050 (GY-521)

Таблиця 2.4 - Маркування контактів модуля MPU-6050 (GY-521)

Позначення	Підключення
VCC	Живлення модуля (3.3В - 5В)
GND	Земля (мінус джерела живлення)
SCL	Serial Clock - тактова лінія шини I2C
SDA	Serial Data - лінія даних шини I2C

XDA	Auxiliary Data – допоміжна лінія SDA для підключення зовнішніх датчиків
XCL	Auxiliary Clock - допоміжна лінія SCL
AD0	Вибір I2C-адреси. Якщо пін не підключено (або притягнуто до GND), адреса датчика 0x68. Якщо подати на нього HIGH (VCC), адреса зміниться на 0x69
INT	Вихід переривання. Сигналізує мікроконтролеру про готовність порції нових даних

Таблиця 2.5 – Основні технічні характеристики модуля MPU-6050

Напруга живлення чіпа MPU-6050	2.375 – 3.46 В
Напруга живлення плати GY-521	3.3 – 5 В
Логічні рівні	3.3 В
Діапазони вимірів:	
- акселерометр (настроюється)	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$
- гіроскоп (настроюється)	$\pm 250, \pm 500, \pm 1000, \pm 2000, \text{ }^\circ/\text{с}$
Інтерфейс зв'язку:	I2C (до 400 кГц)
Особливості. Вбудований апаратний процесор DMP (Digital Motion Processor), який може брати на себе складні математичні обчислення щодо об'єднання даних, розвантажуючи основний контролер	

2.4 Комунікаційні модулі

Для дистанційного керування рухом (вперед, назад, ліворуч, праворуч) балансуючої платформи в проєкті використовується доступний, компактний та енергоефективний Bluetooth-модуль SPP-C (маркування HC-05-SPP-C) [3].

Модуль призначений для організації бездротового послідовного інтерфейсу (UART) за протоколом Bluetooth Classic. Використовується як керований пристрій.

Модуль SPP-C має два основні режими роботи:

- Режим прозорості передачі даних (Data Mode), за умовчанням, коли модуль сполучений з іншим пристроєм (наприклад, зі смартфоном). Всі дані, що надходять на RX, відправляються по Bluetooth, а всі прийняті – йдуть у TX.
- Режим команд (AT Mode): активується, доки модуль не підключено до джерела Bluetooth. У цьому режимі через Serial-порт можна налаштовувати параметри модуля за допомогою спеціальних команд AT.

Маркування контактів модуля для підключення живлення, вихідних/вхідних інформаційних, керуючих та сигнальних ліній представлено на рис. 2.8. та у табл. 2.6. Основні технічні характеристики модуля представлені у табл. 2.7.

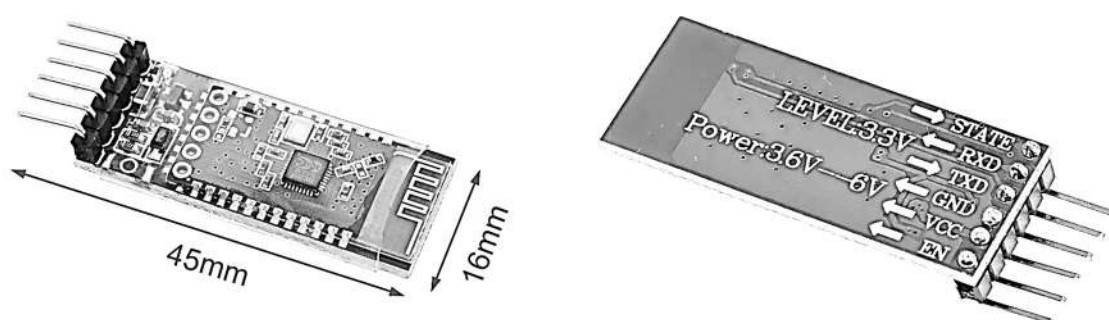


Рисунок 2.8 - Маркування контактів Bluetooth-модуля SPP-C

Таблиця 2.6 – Маркування контактів Bluetooth-модуля SPP-C

Позначення	Підключення
VCC	Живлення модуля 3.3
GND	Земля (мінус джерела живлення)
EN	Вхід. Переведення модуля в режим налаштування AT-команд (високий рівень)
TX	Вихід. Передавач UART
RX	Вхід. Приймач UART

STATE	Вихід. Індикатор стану (видає сигнал, що відображає наявність підключення)
-------	--

Таблиця 2.7 – Основні технічні характеристики Bluetooth-модуля SPP-C

Стандарт Bluetooth	Bluetooth V2.1 + EDR
Профіль	SPP (Serial Port Profile) – профіль послідовного порту для прозорії передачі даних
Радіус дії	Клас 2 (до 10 м)
Робоча частота	ISM-діапазон 2.4 ГГц
Напруга живлення мікросхеми	3.3 У
Напруга живлення плати	3.6 В – 6 В
Логічні рівні (RX/TX)	3.3 У
Енергоспоживання	4.7-8 мА в режимі очікування/підключення

2.5 Підсистема живлення

Джерелом електроенергії живлення системи – платформи, що балансує, обрані Li-Ion акумулятори типу 18650 – 3 шт., підключені послідовно. Сумарна робоча напруга блоку акумуляторів становить від 9.6 до 13 Ст.

Враховуючи параметри живлення електродвигунів – до 12 В, в підсистему живлення включений перетворювач напруги постійного струму, що понижує, Міні DC-DC 4.5–24В на мікросхемі MP2315 виробництва компанії Monolithic Power Systems (MPS) [4]. За замовчуванням плата працює в режимі плавного регулювання через дрібний підстроювальний резистор (ADJ), але якщо перерізати тонку доріжку біля нього і капнути припоєм на один з фіксованих майданчиків, можна отримати чіткі 1.8В, 2.5В, 3.3В, 5В, 9В або 12В з 9В або 12В з 9В або 12В «попливе» від вібрації чи нагрівання.

Знижувальний DC-DC перетворювач призначений для зниження напруги 4.5 – 24В до регульованої або фіксованої вихідної напруги 0.8 – 17В. При довготривалому навантаженні понад 1.5 – 2А модулю потрібне додаткове охолодження (радіатор чи обдування), рис. 2.9.



Рисунок 2.9 – Знижувальний перетворювач напруги постійного струму
"Міні DC-DC 4.5-24В" на мікросхемі MP2315

У проекті вихідна напруга перетворювача підлаштовується на 9 В. Розрахунок сумарного навантаження по всіх компонентах електронної та електричної підсистеми балансує платформи не перевищує 0.7 А, що задовольняє вимогам довготривалого навантаження обраного понижуючого DC-DC перетворювача без додаткового охолодження мікросхем.

Враховуючи параметри живлення 3.3В і 5В окремих модулів, що становлять електронну підсистему балансує платформи, що проектується, для зниження і стабілізації напруги від «Міні DC-DC 4.5–24В» перетворювача по двох каналах живлення на рівнях 3.3В і 5В обраний популярний модуль живлення макетних плат. 2.10.

Максимальний вихідний струм стабільного живлення 700 мА за двома каналами 3.3В та 5В. Розрахунок сумарного навантаження по всіх компонентах електронної підсистеми платформи, що балансує, не перевищує 0.2 А, що задовольняє вимогам довготривалої роботи модуля живлення макетних плат MB102 Arduino.

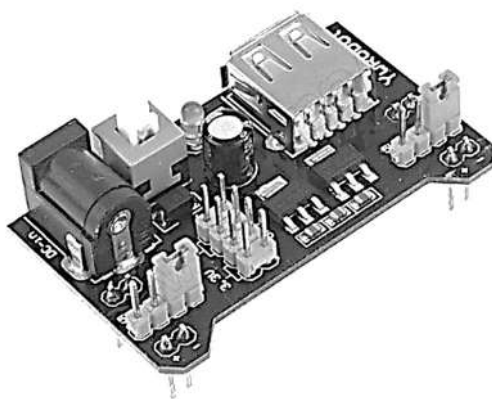


Рисунок 2.10 – Модуль живлення макетних плат MB102 Arduino

2.6 Підсистема керування

Для реалізації балансуєної двоколісної платформи, на підставі поставлених завдань, вибраних технічних засобів автоматизації та аналізу інформаційних та керуючих потоків вводу/виводу, в даному проекті як обчислювальна підсистема обрана популярна і доступна платформа Arduino Nano [5] на базі мікроконтролера ATmega328p [6]. 2.11. Основні технічні характеристики модуля Arduino Nano представлені у табл. 2.8.

Таблиця 2.8 – Основні технічні характеристики Arduino Nano

Напруга живлення логіки	5В
Цифрові входи/виходи	14 (з яких 6 можуть використовуватися як ШІМ- виходи)
Аналогові входи	8 (з яких 2 можуть використовуватись як інтерфейсні I2C)
Flash-пам'ять	32 КБ з яких 2 КБ використовуються завантажувачем
Тактова частота	16 МГц

Таблиця 2.9 - Кількість входів/виходів технічних засобів автоматизації

Технічний засіб автоматизації	Кількість входів/виходів	Тип сигналу
Драйвер електродвигунів TB6612FNG	6	цифровий
Енкодери електродвигунів	4	цифровий
Інерційний датчик MPU-6050	2	цифровий
Bluetooth-модуль SPP-C	2	цифровий
Arduino Nano	14	цифровий
Arduino Nano. Цифрові інтерфейси I2C входи/виходи	2	цифровий

Усього потрібно на платі Arduino Nano 14 цифрових входів/виходів. Усього є на платі Arduino Nano 16 цифрових входів/виходів. Платформа Arduino Nano повністю забезпечує необхідну кількість цифрових входів/виходів для реалізації платформи, що балансує, на вибраних засобах автоматизації, табл. 2.9.

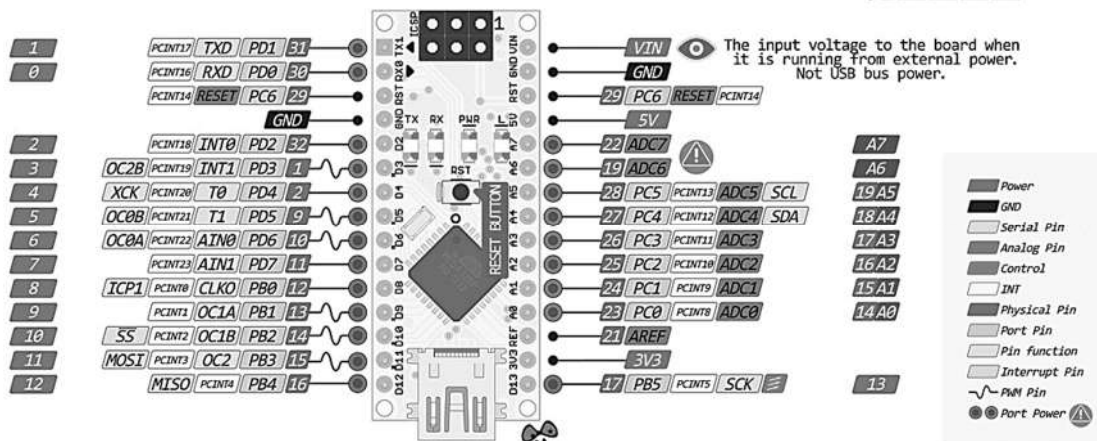


Рисунок 2.11 – Мікроконтролерна платформа Arduino Nano

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Отримання інформації

3.1.1 Інерційні вимірювальні перетворювачі

Для визначення положення платформи, що балансує, найчастіше використовуються інерційні вимірювальні блоки - акселерометр і гіроскоп.

Акселерометр, встановлений на платформу, що балансує, вимірює прискорення a – векторну величину, визначальну наскільки швидко змінюється швидкість платформи Δv в напрямку руху платформи при прямолінійному русі або в напрямку до центру при обертальному русі в одиницю часу Δt :

$$a = \frac{\Delta v}{\Delta t}; \quad (3.1)$$

$$a = \omega R, \quad (3.2)$$

де ω – кутова швидкість, R – радіус обертання.

У даній роботі, для визначення положення платформи, що балансує, обраний недорогий модуль MPU-6050, акселерометр якого будемо використовувати для визначення кута відхилення платформи від горизонтальної осі. Цей акселерометр досить точно вимірює прискорення платформи, що встановилася, або рухається без прискорення. У цьому стані на платформу діє тільки сила тяжіння, і тоді проекції вектора прискорення вільного падіння на нахилену вісь напрямку руху платформи уобчислюємо кут α_{yz} відхилення платформи від вертикальної осі z :

$$\alpha_{yz} = \arctg \frac{\alpha_y}{\alpha_z}, \quad (3.3)$$

де α_y - Проекція вектору прискорення вільного падіння на вісь у платформи (акселерометра); α_z - Проекція вектора прискорення вільного падіння на перпендикулярну осі у вісь z.

На відміну від механічного гіроскопа, що вимірює кут відхилення від заданого напрямку, електронний гіроскоп вимірює кутову швидкість обертання платформи по осях. Знаючи кутову швидкість, можна обчислити кут відхилення платформи від заданого напрямку руху за проміжок часу Δt :

$$\alpha_2 = \alpha_1 + \Delta t \cdot \omega, \quad (3.4)$$

де α_1 - так само розрахований раніше кут; α_2 – знову розрахований кут повороту платформи навколо осі повороту щодо кута повороту α_1 .

Загалом, акселерометр використовується для визначення нахилу щодо вектора прискорення вільного падіння, а гіроскоп – для відстеження поворотів щодо певної осі. У парі ці два датчики дають більш повну картину руху платформи, що балансує.

3.1.2 Фільтрування сигналів

Вимірювальна інформація від датчиків зазвичай схильна до впливу зовнішніх неконтрольованих впливів, через що в сигналах присутня випадкова похибка - шум. Також у вимірювальній інформації присутні похибки, зумовлені метрологічними характеристиками самих датчиків. У контурі управління сумарна похибка вимірювальної інформації впливає на точність управляючих впливів і призводить до помилок управління, а у контурах зворотного зв'язку і до нестійкості систем управління.

Щоб зменшити вплив випадкової похибки на точність вимірювання, застосовують математичні методи згладжування та фільтрації сигналів вимірювальної інформації.

Компліментарний фільтр дозволяє об'єднати дані двох датчиків, зазвичай гіроскопа та акселерометра, з метою уточнення оцінки кута нахилу із систем реального часу. Фільтр зменшує дрейф гіроскопа та шуми акселерометра. Алгоритм фільтрації враховує низьку чутливість до вібрацій гіроскопа та поєднує фільтр низьких частот, що враховує відсутність дрейфу акселерометра та поєднує фільтр високих частот, стійкий до швидких змін сигналу:

$$filteredAngle = \alpha (filteredAngle + gyroRate \cdot dt) + (1.0 - \alpha) \cdot accAngle;$$

де α - коефіцієнт фільтрації (0.95-0.99).

Наприклад, для датчиків модуля MPU-6050, використовуючи бібліотеку <MPU6050.h>, зчитуємо дані з датчиків по 3 осях у відповідні змінні:

```
// Читання «сирих» даних із датчика
int16_t ax , ay , az ;
int16_t gx , gy , gz ;
mpu.getMotion 6(&ax, &ay, & az , & gx , & gy , & gz );
// Переведення даних гіроскопа в градуси за секунду (°/s)
// У бібліотеці чутливість становить ±250 °/s (ділитель 131.0)
float gyroX = gx /131.0;
float gyroY = gy /131.0;
// Розрахунок кутів нахилу по акселерометру за «сирими» значенням
// ax , ay , az . Знак мінус коригує напрямок осей
float accAngleX = atan2 ( ay , sqrt (( long ) ax * ax + ( long ) az * az )) *
RAD_TO_DEG;
```

```

float accAngleY = atan2(-ax, sqrt((long)ay * ay + (long) az * az )) *
RAD_TO_DEG;
// Первинна ініціалізація кутів (без калібрування нуля)
if (! isInitialized ) {
    filteredAngleX = accAngleX ;
    filteredAngleY = accAngleY ;
    isInitialized = true ;
}
// Формула компліментарного фільтра для уточнення оцінки
// кута нахилу платформи по осях X і Y
filteredAngleX = ALPHA * ( filteredAngleX + gyroX * dt) + (1.0 - ALPHA) *
accAngleX ;
filteredAngleY = ALPHA * ( filteredAngleY + gyroY * dt) + (1.0 - ALPHA) *
accAngleY ;

```

Тут, у проекті, нахил у напрямку осі X не враховується, передбачається, що платформа переміщується на горизонтальній площині, тоді залишається і спрощується розрахунок кута нахилу тільки в напрямку осі Y, за формулою (3.3):

```
float accAngleY = atan2(ay, az ) * RAD_TO_DEG; // RAD_TO_DEG = 180/PI
```

Аналогічна задача фільтрації вирішується з використанням фільтра Калмана, що добре зарекомендував себе і перевіреного часом, алгоритм якого, в проекті, об'єднує дані двох датчиків (гіроскопа і акселерометра) модуля MPU6050 з метою уточнення оцінки кута нахилу балансуєної платформи по осі у відносно вертикальної осі z.

Рекурсивний алгоритм фільтра працює у два етапи: 1 – прогноз наступного значення та 2 – корекція (отримуємо дані датчиків та виправляємо прогноз).

1 Прогноз

Розраховуємо попередню оцінку.

- новий стан

$$x_{apr} = x_{t-1} \quad (3.5)$$

- помилка прогнозу

$$P_{apr} = P_{t-1} + Q \quad (3.6)$$

де P – невпевненість щодо оцінки; Q – шум процесу.

2 Корекція

Дивимося на показання датчика (z) та обчислюємо підсумкову оцінку

- коефіцієнт Калмана (K):

$$K = \frac{P_{apr}}{P_{apr} + R} \quad (3.7)$$

де R – шум датчика; K – «вага», яка визначає, кому віримо більше – прогнозу чи датчику. Якщо датчик точний (R мало), K близький до 1, тоді віримо датчику. Якщо датчик дуже галасливий (R велике), K близький до 0, тоді віримо прогнозу;

- фінальна оцінка (x):

До прогнозу додаємо різницю між датчиком та прогнозом, помножену на коефіцієнт Калмана.

$$x_t = x_{apr} + K(z - x_{apr}) \quad (3.8)$$

- оновлення помилки:

Впевненість у результаті зростає (помилка P зменшується), оскільки об'єднали два джерела інформації.

$$P_t = (1 - K)P_{apr} \quad (3.9)$$

Наприклад, на підставі попереднього вимірювання датчика акселерометра, обчислений кут нахилу $Angle_{t-1}$ дорівнює 5° . Прогнозуємо $AngleY_{apr}$ рівним 5° . Поточне обчислене значення нахилу $AngleY_t$ становить 7° . Знаємо, що акселерометр дуже чутливий до вібрацій, тому довіряємо йому мало і приймаємо $K = 0.1$. Тому коригуємо нове виміряне значення кута нахилу:

$$AngleY_t = Angle_{apr} + K(AngleY_t - Angle_{apr}) = 5 + 0.1 \cdot (7 - 5) = 5.2$$

3.1.3 Електромеханічні вимірювальні перетворювачі

У проекті необхідно забезпечити вимірювання напрям зсуву в просторі платформи, що балансує, від встановленого стану, а також розраховувати швидкість цього зсуву. Для вирішення цих завдань підходить інкрементний енкодер, встановлений на валу електродвигуна. Інкрементний енкодер перетворює обертальний рух валу електродвигуна на серію електричних імпульсів.

Основними елементами обраного енкодера є два датчики Холла, розташовані на корпусі електродвигуна і магнітний диск, з магнітними і немагнітними секторами, що чергуються, встановлений на валу електродвигуна. Два датчики необхідні для визначення напрямку обертання валу. Ці датчики забезпечують два основних канали зв'язку А і В. Датчики зміщені на корпусі так, що сигнали каналів А і В зсунуті фазою на 90° . Для визначення напрямку обертання валу мікроконтролер порівнює ці сигнали; якщо сигнал А (логічний 1) випереджає сигнал В (логічний 0), вал обертається за годинниковою стрілкою, і навпаки, якщо сигнал В випереджає сигнал А.

Щоб вимірювати швидкість обертання валу електродвигуна, можна фіксувати моменти часу початку відліку імпульсів, каналу А або В, і кінця обраного періоду часу. Знаючи тривалість періоду часу, значення лічильника імпульсів, і роздільну здатність енкодера, розраховуємо частоту обертання валу електродвигуна. Роздільна здатність енкодера визначається кількістю імпульсів за один оберт, і залежить від кількості магнітних секторів, що перемежуються, на магнітному диску.

Наприклад. Роздільна здатність енкодера 4 імпульсу на один повний оборот валу. Період часу лічби імпульсів 10 мс (0.01 с). Кількість імпульсів у період рахунку 20.

У разі рівномірного обертального руху частота або кутова частота (кутова швидкість) обчислюються за формулами:

$$f = \left(\frac{20}{0.01} \right) / 4 = 500, \frac{\text{об}}{\text{с}}, \text{Гц}; \quad (3.10.1)$$

$$\omega = 2\pi f = 6.28 \cdot 500 = 3140, \frac{\text{рад}}{\text{с}}. \quad (3.10.2)$$

3.2 Управління

3.2.1 Регулювання кута нахилу платформи, що балансує.

Для підтримки балансуючої платформи у вертикальному стані використовуємо ПД-закон регулювання. Задане значення – це потрібний кут нахилу робота щодо вертикальної осі z. Вхідне значення – це фактичний нахил робота, що спостерігається. Вихідне значення – напрямок та швидкість обертання колеса робота.

Різниця між заданим значенням регульованої величини $g(t)$ і значенням регульованої величини, що спостерігається, $y(t)$ – це помилка регулювання $e(t)$, рис. 3.1:

$$e(t) = g(t) - y(t). \quad (3.11)$$

Вихідний сигнал ПД-регулятора визначається пропорційно до поточної та кумулятивної помилок, та швидкості зміни помилки, відповідно до пропорційної, інтегральної та диференціальної складових ПД-закону регулювання:

$$u(t) = Kp * e(t) + Ki * t \sum e(t) + Kd * \frac{de(t)}{dt}, \quad (3.12)$$

де Kp, Ki, Kd – коефіцієнти пропорційної, інтегральної та диференціальної складових ПД-закону регулювання.

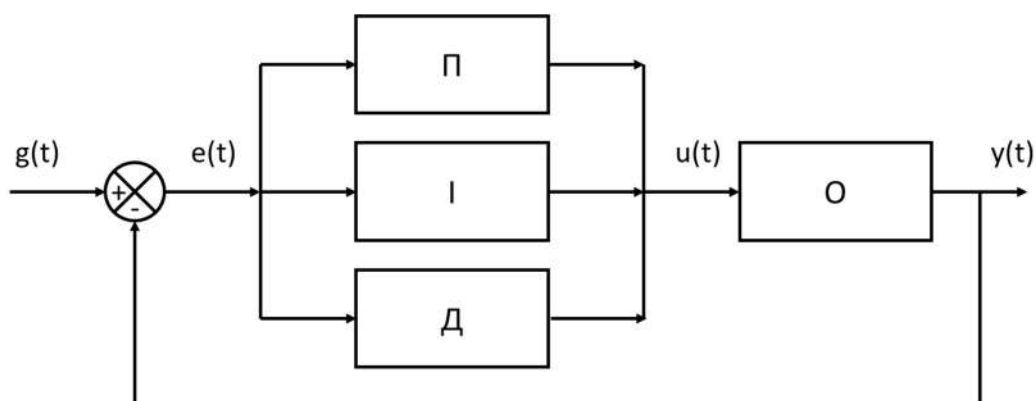


Рисунок 3.1 – Структурна схема ПД регулювання

Похідну $\frac{de(t)}{dt}$ апроксимуємо зворотною кінцевою розділеною різницею і будемо обчислювати, як:

$$e'(t_i) = \frac{de(t)}{dt} = \frac{e(t_i) - e(t_{i-1})}{t_i - t_{i-1}}, \quad (3.13)$$

де t_i і t_{i-1} , поточний та попередній моменти часу; $(t_i - t_{i-1})$ – інтервал між послідовними обчисленнями вихідного сигналу регулятора.

Якщо задане значення $g(t)$ не змінюється, тоді похідну помилку регулювання можна обчислювати, як:

$$e'(t_i) = \frac{g(t_i) - y(t_i) - (g(t_{i-1}) - y(t_{i-1}))}{t_i - t_{i-1}} = \frac{y(t_{i-1}) - y(t_i)}{t_i - t_{i-1}}, \quad (3.14)$$

і рівняння (3.12) можна подати у вигляді:

$$u(t_i) = Kp * e(t_i) + Ki * dt * \sum_{t=-\infty}^{t=i} e(t) + Kd * e'(t_i), \quad (3.15)$$

або в розгорнутому вигляді:

$$u(t_i) = Kp * e(t_i) + Ki * (t_i - t_{i-1}) * \sum_{t=-\infty}^{t=i} e(t) + Kd * \frac{y(t_{i-1}) - y(t_i)}{t_i - t_{i-1}}. \quad (3.16)$$

3.2.2 Регулювання швидкості руху платформи

Якщо платформу піддати впливу зовнішніх сил, наприклад підштовхнути, то платформа прискорюватиметься у бік впливу, тобто у напрямку нахилу платформи. Щоб зберегти вертикальне положення платформи необхідно забезпечити позитивне прискорення у бік впливу зовнішньої сили, щоб кут нахилу платформи змінився на протилежний і повернувся до близького значення в точці рівноваги.

Тут використовуватимемо контур ПІ регулювання швидкості. Задане значення швидкості платформи 0.

3.2.3 Управління напрямом руху платформи

При поступальному русі платформи відхилення від центру напрямку вимагає управління. Тобто задіюється рульове керування. Шляхом регулювання напрямку руху платформи та додавання його до поступального руху можна усунути різницю між відхиленням платформи від центральної лінії руху.

Для реалізації кермового управління звичайно використовувати швидкий пропорційний закон регулювання.

До П закону регулювання додамо диференціальну складову, щоб зменшити чи взагалі виключити перерегулювання. Коли процес стрімко наближається до заданого значення, ця складова створює гальмуючий вплив і запобігає проскакуванню заданого значення, і можливі сильні коливання щодо заданого значення. Диференційна складова реагує зміну помилки регулювання. Помилкою регулювання тут є кут повороту платформи відносно вертикальної осі z .

3.2.4 Дистанційне керування рухом платформи

Управління рухом платформи реалізується формуванням сигналу широтно-імпульсної модуляції у діапазоні 0...255. ШІМ сигнал формується програмою управління та подається з виведення мікроконтролера на драйвер електродвигуна. Драйвер, що пропорційно управляє сигналом, змінює напругу живлення електродвигуна і, відповідно, змінюється швидкість обертання валу електродвигуна.

Для забезпечення прямолінійного руху платформи управляючі сигнали для двох електродвигунів мають однакове значення. При русі з поворотами у вибраному напрямку, платформа рухається різницею швидкостей обертання валів електродвигунів.

Для дистанційного примусового управління рухом платформи, що балансує, необхідно забезпечити бездротовий зв'язок між провідним пристроєм, наприклад, смартфоном або ПК, і веденим, яким є сама балансуєча платформа. Сигнали управління на стороні провідного пристрою формуються спеціалізованим додатком, що взаємодіє з оператором, і передаються бездротовим каналом по протоколу стандартного послідовного інтерфейсу. На стороні веденого пристрою необхідно опитувати відповідний канал зв'язку, зчитувати дані і вносити відповідні зміни параметрів управління, або безпосередньо змінювати величину сигналів, що управляють ШІМ в програмі управління веденого пристрою.

4 РОЗРОБКА ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Розробка та реалізація монтажної плати

Схема зібрана за індивідуальним проектом без необхідності розробляти та витравлювати друковану плату. Вибрано монтажну макетну плату під пайку з текстоліту 8 х 6 зі стандартним кроком отворів 2.54 мм з острівною пайкою. Компоненти розміщено, рис. 4.1, наскрізним монтажем (TND - Through Hole Device) на платі та з'єднуються між собою навісним монтажем відрізками дроту. Для підключення периферійних (щодо монтажної плати) компонентів системи на монтажній платі розміщені штирьові з'єднувачі – класичні гребінки PLS (штирі квадратного перерізу) з кроком 2.54 мм. Штирьові виводи роз'єму, як і інші компоненти вставлені в наскрізні отвори на платі та запаяні зі зворотного боку. До штирьових роз'ємів підключаються або багатоконтактні гнізда BLS або різнокольорові дроти з вже встановленими роз'ємами типу Dupont (BLS). До деяких штирьових роз'ємів підпаюються проводи для більш надійного з'єднання, що виключає брязкіт контактів, зокрема, розпаяні з'єднання для модуля Bluetooth.

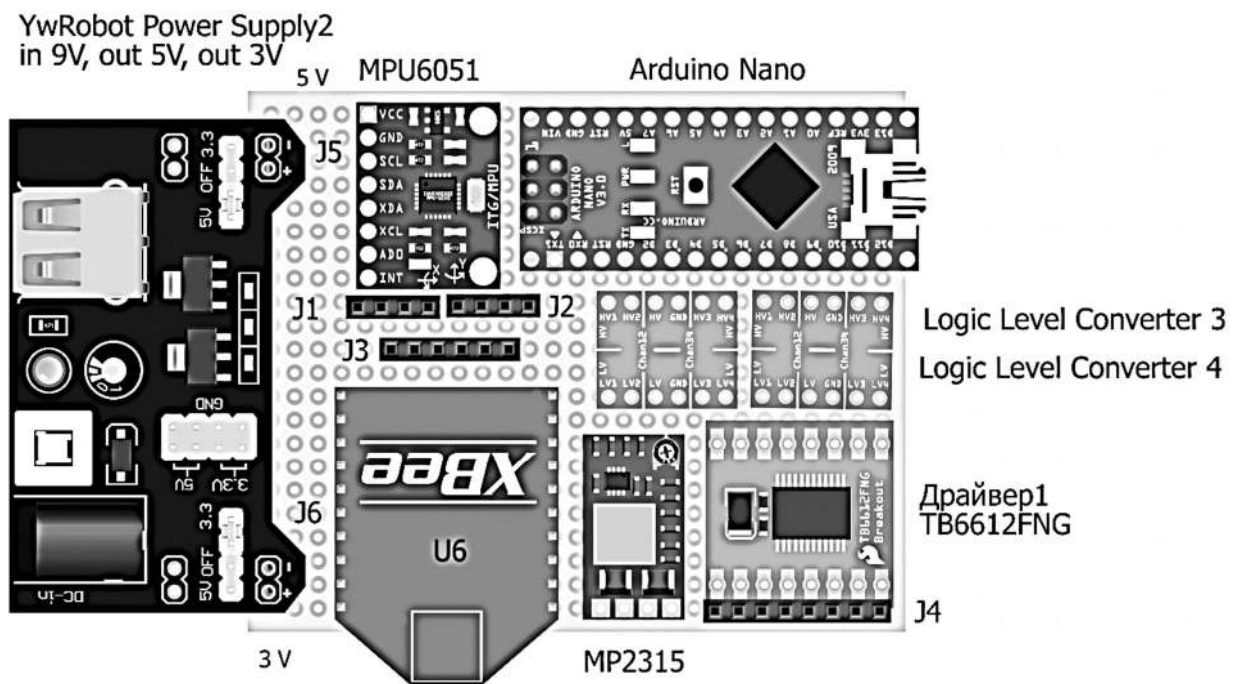


Рисунок 4.1 – Розміщення компонентів на монтажній платі

4.2 Розробка монтажної схеми підключення

Для розробки монтажної схеми підключення компонентів роботизованої балансуєчої платформи використано спеціалізовану платформу Fritzing для проектування та прототипування електронних пристроїв.

Проект виконується за етапами.

Спочатку визначається робочий простір створення монтажної схеми.

На робочий простір перетягуються необхідні компоненти з наявної бібліотеки, які відображаються на вкладці « Parts ».

Створюємо з'єднання між компонентами та підключення периферійних компонентів. У табл. 4.1, ..., 4.9 представлені з'єднання виводів окремих компонентів, розташованих на монтажній платі та підключення периферійних компонентів до штирьових роз'ємів, розташованих на монтажній платі.

Таблиця 4.1 – Підключення електродвигунів та енкодерів, див. табл. 2.1 та рис. 2.5

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Мотор лівий	M1 (+)	J4	4
Мотор лівий	M2 (-)	J4	5
Мотор правий	M1 (+)	J4	6
Мотор правий	M2 (-)	J4	7
Енкодер лівий	G	J1	1
Енкодер лівий	B	J1	3
Енкодер лівий	A	J1	2
Енкодер лівий	V	J1	4
Енкодер правий	G	J2	1
Енкодер правий	B	J2	3
Енкодер правий	A	J2	2
Енкодер правий	V	J2	4

Таблиця 4.2 – З'єднання модуля живлення макетних плат MB102

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
J7	GND	Перетворювач напруги MP2315	Pin 3: GND
J7	VCC	Перетворювач напруги MP2315	Pin 4: VOUT

Таблиця 4.3 – Підключення блоку акумуляторів та кнопки живлення, див. мал. 2.9

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Блок акумуляторів	+ (Плюс 13 В)	Перетворювач напруги MP2315	Pin 2: VIN
Блок акумуляторів	- (мінус)	Перетворювач напруги MP2315	Pin 3: GND
Блок акумуляторів	- (мінус)	Кнопка живлення TOUCH KEY	GND
Блок акумуляторів	+ (Плюс 3.7 В)	Кнопка живлення TOUCH KEY	VCC
Кнопка живлення TOUCH KEY	GND	Перетворювач напруги MP2315	Pin 3: GND
Кнопка живлення TOUCH KEY	OUT	Перетворювач напруги MP2315	Pin 1: EN

Таблиця 4.4 - Підключення та з'єднання драйвера TB6612FNG, див. табл. 2.3 та рис. 2.6

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Драйвер TB6612FNG	V+ Moteurs	J4	1
Драйвер TB6612FNG	V+ Logic	J4	2
Драйвер TB6612FNG	V+ Logic	Драйвер TB6612FNG	Stand By

Драйвер TB6612FNG	V+ Logic	Logic Level Converter 2	LV
Драйвер TB6612FNG	Stand By	Logic Level Converter 1	LV
Драйвер TB6612FNG	Masse - 0V	J4	8
Драйвер TB6612FNG	Masse - 0V	Драйвер TB6612FNG	GND Logic
Драйвер TB6612FNG	GND Logic	Logic Level Converter 1	GND
Драйвер TB6612FNG	Mot A, OUT 1	J4	4
Драйвер TB6612FNG	Mot A, OUT 2	J4	5
Драйвер TB6612FNG	Mot B, OUT 1	J4	6
Драйвер TB6612FNG	Mot B, OUT 2	J4	7
Драйвер TB6612FNG	Mot B.PWM	Logic Level Converter 1	LV2
Драйвер TB6612FNG	Mot B.IN2	Logic Level Converter 2	LV3
Драйвер TB6612FNG	Mot B.IN1	Logic Level Converter 2	LV4
Драйвер TB6612FNG	Stand By	Logic Level Converter	LV
Драйвер TB6612FNG	Mot A.IN1	Logic Level Converter 1	LV4

Драйвер TB6612FNG	Mot A.IN2	Logic Level Converter 1	LV1
Драйвер TB6612FNG	Mot A.PWM	Logic Level Converter 1	LV3

Таблиця 4.5 - З'єднання перетворювачів логічних рівнів сигналів

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Logic Level Converter 1	LV1	Драйвер TB6612FNG	Mot A.IN2
Logic Level Converter 1	LV2	Драйвер TB6612FNG	Mot B.PWM
Logic Level Converter 1	LV3	Драйвер TB6612FNG	Mot A.PWM
Logic Level Converter 1	LV4	Драйвер TB6612FNG	Mot A.IN1
Logic Level Converter 2	LV1	J3	3
Logic Level Converter 2	LV2	J3	2
Logic Level Converter 2	LV3	Драйвер TB6612FNG	Mot B.IN2
Logic Level Converter 2	LV4	Драйвер TB6612FNG	Mot B.IN1
Logic Level Converter 1	LV	Драйвер TB6612FNG	Stand By
Logic Level Converter 1	HV	Logic Level Converter 2	HV
Logic Level Converter 2	LV	Драйвер TB6612FNG	V+ Logic

Logic Level Converter 2	LV	J2	4
Logic Level Converter 2	HV	Arduino	5V
Logic Level Converter 1	GND	Драйвер TB6612FNG	GND Logic
Logic Level Converter 1	GND	Logic Level Converter 2	GND
Logic Level Converter 2	GND	J3	6
Logic Level Converter 2	GND	Arduino	GND
Logic Level Converter 1	HV1	Arduino	D8
Logic Level Converter 1	HV2	Arduino	D9
Logic Level Converter 1	HV3	Arduino	D10
Logic Level Converter 1	HV4	Arduino	D12
Logic Level Converter 2	HV1	Arduino	TX1
Logic Level Converter 2	HV2	Arduino	RX0
Logic Level Converter 2	HV3	Arduino	D6
Logic Level Converter 2	HV4	Arduino	D7

Таблиця 4.6 - З'єднання модуля MPU-6050, див. 2.4 та рис. 2.7

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
MPU-6050	VCC	J6	1 (плюс)
MPU-6050	GND	Arduino	GND
MPU-6050	GND	J5	2 (мінус)
MPU-6050	SCL	Arduino	A5
MPU-6050	SDA	Arduino	A4

Таблиця 4.7 – З'єднання модуля Bluetooth SPP-C, див. 2.6 та рис. 2.8

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Bluetooth-модуль	VCC	J3	1
Bluetooth-модуль	GND	J3	6
Bluetooth-модуль	TX	J3	3
Bluetooth-модуль	RX	J3	2

Таблиця 4.8 – З'єднання модуля Arduino Nano

Компонент 1	Позначення контакту	Компонент 2	Позначення контакту
Arduino Nano	TX	Logic Level Converter 2	HV1
Arduino Nano	RX	Logic Level Converter 2	HV2
Arduino Nano	GND	Logic Level Converter 2	GND
Arduino Nano	GND	MPU-6050	GND
Arduino Nano	5V	J5	1 (плюс)
Arduino Nano	5V	Logic Level Converter 2	HV
Arduino Nano	D2	J1	3
Arduino Nano	D3	J2	3
Arduino Nano	D4	J2	2
Arduino Nano	D5	J1	2
Arduino Nano	D6	Logic Level Converter 2	HV3
Arduino Nano	D7	Logic Level Converter 2	HV4
Arduino Nano	D8	Logic Level Converter 1	HV1
Arduino Nano	D9	Logic Level Converter 2	HV2
Arduino Nano	D10	Logic Level Converter 2	HV3
Arduino Nano	D12	Logic Level Converter 2	HV4
Arduino Nano	A4	MPU-6050	SDA

Arduino Nano	A5	MPU-6050	SCL
--------------	----	----------	-----

Таблиця 4.9 – З'єднання штирьових роз'ємів на монтажній платі

Штирьовий роз'єм	Позначення контакту	Компонент 2	Позначення контакту
J1	1	J2	1
J1	1	J6	2 (мінус)
J1	2	Arduino Nano	D5
J1	3	Arduino Nano	D2
J1	4	J3	1
J2	1	J3	6
J2	1	J1	1
J2	2	Arduino Nano	D4
J2	3	Arduino Nano	D3
J2	4	J1	4
J3	1	J1	4
J3	1	Модуль Bluetooth	VCC
J3	1	J6	1 (плюс)
J3	2	Logic Level Converter 2	LV2
J3	3	Logic Level Converter 2	LV1
J3	6	Модуль Bluetooth	GND
J3	6	J2	1
J3	6	Logic Level Converter 2	GND
J4	1	Перетворювач напруги MP2315	Pin 4: VOUT
J4	1	Драйвер TB6612FNG	V+ Moteurs
J4	2	Драйвер TB6612FNG	V+ Logic
J4	3	Перетворювач напруги MP2315	Pin 3: GND

J4	3	J4	8
J5	1	Arduino Nano	5V
J5	2	MPU-6050	GND
J6	1	J3	1
J6	1	MPU-6050	VCC
J7	1	Перетворювач напруги MP2315	VOUT
J7	2	Перетворювач напруги MP2315	GND

Наступний етап розробки – перевірка з'єднань та підключень компонентів. Схеми електричних принципових підключень і з'єднань представлені на листах A26. 13.APT.01.СБ та A26.13.APT.02.СБ відповідно.

Специфікацію обладнання на засоби автоматизації представлено на листах A26. 13.A PT.00.C.

4.3 Розробка електричної принципової схеми підключень

Спеціалізована платформа Fritzing надає можливість автоматичного формування схеми електричної принципової сполук компонентів системи. На вкладці «Схема» перевіряємо та коригуємо з'єднання компонентів.

Електрична принципова схема з'єднань компонентів роботизованої балансуєчої платформи, з урахуванням коригування, представлена на листі A26. 13.A PT.02.СБ.

4.4 Реалізація роботизованої балансуєчої платформи

Результат реалізації платформи, що балансує, а саме складання та монтаж компонентів представлений на рис. 4.2, 4.3.

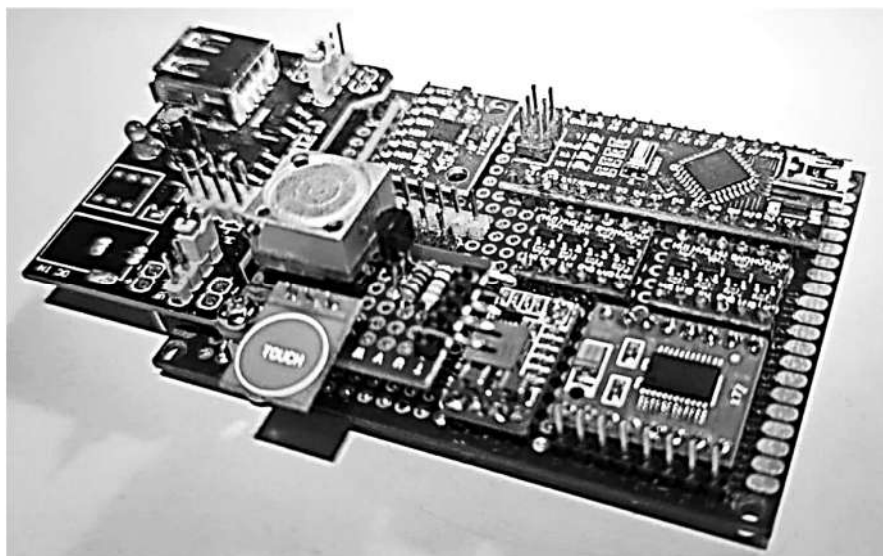


Рисунок 4.2 – Реалізація монтажної плати

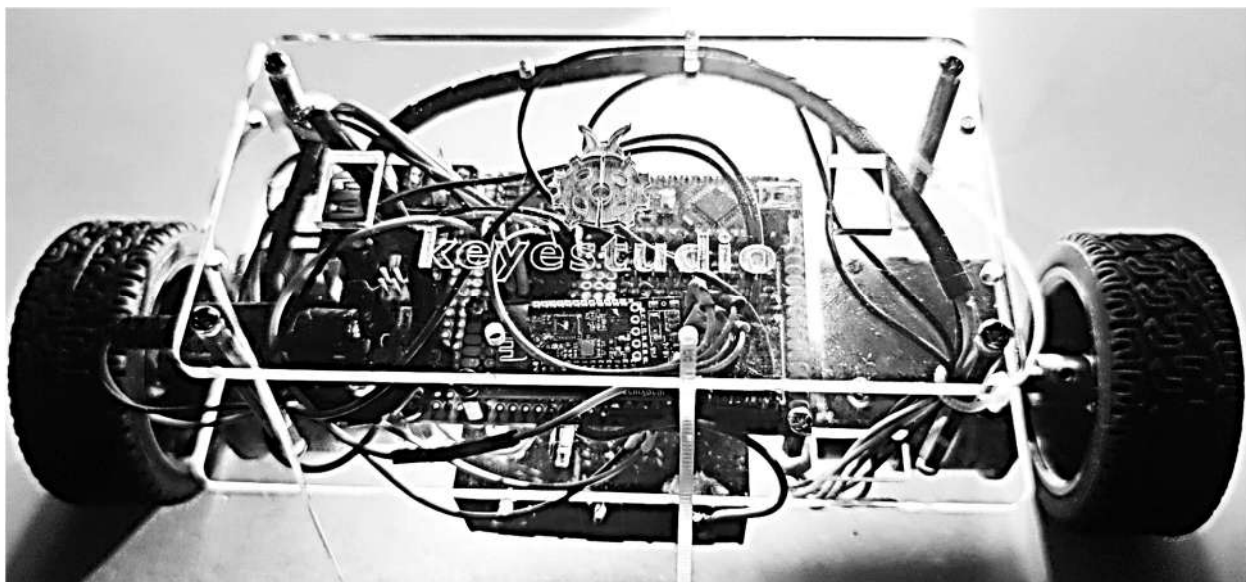


Рисунок 4.3 – Реалізація платформи, що балансує.

5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

5.1 Регулювання кута нахилу платформи, що балансує.

Для роботи з модулем MPU-6050 серед Arduino IDE існує кілька готових рішень:

- Бібліотека Adafruit_MPU6050 автоматично перераховує «сирі» значення АЦП в одиниці виміру прискорення $\frac{m}{c^2}$ та кутової швидкості $\frac{^\circ}{c}$;
- Бібліотека MPU6050 (від Jeff Rowberg) – просунута бібліотека, яка дозволяє задіяти вбудований процесор DMP без завантаження процесора Arduino складною тригонометрією.

Програма (без бібліотек) для тестування інерційного модуля MPU-6050 у середовищі Arduino IDE наведена у Додатку 1. Програма зчитує «сирі» дані безпосередньо з регістрів модуля.

У проекті використано бібліотеку MPU6050 (від Jeff Rowberg), див. Додаток 3. Для отримання сирих даних використовується метод `mpu6050.getMotion6(...)`. Обробка сирих даних для отримання кута відхилення та кутової швидкості відхилення платформи від вертикальної осі z реалізується у функції `angle_calculate ( ... )`. Оброблені дані уточнюються функцією `Kalman_Filter ( ...)`, і далі використовуються вже в контурі керування платформою – в контурі регулювання вертикального положення платформи. Розрахунок значення управляючого впливу `PD_pwm` за ПД-законом регулювання реалізований функції `PD( )`. Керуючий вплив далі подається на драйвер двигуна: обмежений ШІМ-сигнал управління кутовою швидкістю обертання валу електродвигуна функції `anglePWM ( )`. Функція `anglePWM ( )` також формує сигнали напрямку обертання валу електродвигуна, які подаються на драйвер електродвигунів, див. Додаток 3.

5.2 Регулювання швидкості руху платформи

Щоб вимірювати швидкість платформи, використовуємо дані енкодерів і будемо вимірювати кількість імпульсів за певний період часу.

Період часу можна визначати фіксуванням моментів часу між вибірками у кожному основному циклі управління. У початковий момент часу обнулюємо лічильник імпульсів і запускаємо лічильник. У наступному циклі читаємо дані лічильника та лічильного регістра таймера. Таким чином, у кожному циклі одержуємо дані для обчислення швидкості руху платформи.

У проекті організовано явні періоди часу обробки даних; для цього задіяний системний таймер, блок порівняння вихідних сигналів та механізм переривань. У блоці порівняння 8-бітний компаратор порівнює значення в лічильному регістрі системного таймера з регістром порівняння вихідних сигналів OCR2A [6]. Щоразу, коли значення збігаються, компаратор сигналізує про збіг, встановлюючи прапор порівняння вихідних сигналів OCF2A на наступному тактовому циклі таймера. Якщо активувати відповідне переривання, прапор порівняння вихідних сигналів генерує відповідне переривання. Таким чином, у проекті явно встановлено період обробки даних. З урахуванням необхідних обчислень в основному циклі програми управління період обробки даних прийнятий рівним 10 мс. Тоді для розрахунку швидкості руху платформи достатньо зчитувати в кожному або кратному циклі обробки даних вміст лічильника імпульсів енкодера, обнулювати і запускати лічильник знову.

Також важливо визначати напрямок зовнішнього впливу. Визначення напрямку руху описано у п. 3.1.2. Для цього використовуються зовнішні переривання INT0 та INT1 для сигналів каналів А, лівого та правого енкодерів, та дискретні входи D5 та D4 для каналів В відповідно. Використовуючи

зовнішні переривання, визначаємо напрямок руху платформи, і швидкість руху платформи.

Підрахунок імпульсів від енкодерів лівого та правого двигунів здійснюється обробниками переривань `Code_left()` та `Code_right()`.

Визначення напрямку руху платформи здійснюється у функції `countpulse ()`.

Функція `speedpiout ()` реалізує керуючий вплив `PI_pwm` по ПІ-закону регулювання відхилення поточної позиції платформи (`positions`) від заданого положення `setp0`. Швидкість руху платформи пропорційна різниці значень цих змінних.

5.3 Управління напрямком руху платформи

Реалізація пропорційної складової закону регулювання

Якщо є команда на поворот, обчислюємо поточну швидкість платформи ($turnspeed = (pulseright + pulseleft)$). Значення глобальних змінних `pulseright` і `pulseleft` обчислюються в кожному циклі оновлення інформації. Визначаємо абсолютне значення швидкості, що не залежить від напрямку руху. Ініціалізуємо обмеження на швидкість повороту. Обчислюємо коефіцієнт повороту (`rotationratio`), пропорційний зворотній величині швидкості платформи (щоб на великій швидкості платформи не робити крутих поворотів). Вводимо обмеження значення цього коефіцієнта (мінімум 0.5, максимум 5). Цей коефіцієнт повороту використовується як коригування позиціонування при обертанні платформи навколо осі *z*. Якщо команда повороту вліво, змінна (`turnout`) збільшується на поточну величину коефіцієнта повороту, якщо команда поворот вправо – зменшується. Якщо команди на поворот ліворуч і праворуч немає, змінна (`turnout`) дорівнює 0. Керуюча дія пропорційної складової формується пропорційно значенню змінної (`turnout`).

Реалізація диференціальної складової закону регулювання

У функції `angle_calculate ()` за сирими даними гіроскопа обчислюється кутова швидкість осі *z* в градусах. Тому керуючий вплив (`Turn_pwm`) по диференціальній складовій обчислюється пропорційно кутовий швидкості по осі *z* - (`Gyro_z * kd_turn`).

Функція `turnspin ()` обчислює керуючий вплив ПД-регулятора за швидкістю та позиціонування обертання щодо вертикальної осі.

5.4 Дистанційне керування рухом платформи

На провідному пристрої, за посиланням [7] завантажуюмо програму Android на смартфон. Встановлюємо програму. Переходимо до Налаштування -> Підключення (або Bluetooth) і переводимо тумблер в активне положення.

Переводимо ведений пристрій у режим сполучення: після увімкнення живлення модуль Bluetooth сигналізує блиманням світлодіода про відсутність з'єднання з каналом зв'язку.

На провідному пристрої в меню Bluetooth натискаємо кнопку Пошук пристроїв (або телефон автоматично сканує оточення). У списку доступних для підключення пристроїв знаходимо назву веденого пристрою та вибираємо його. Якщо потрібно, підтверджуємо запит на створення пари.

Або, у вікні програми, рис. 5.1, натискаємо значок Bluetooth і автоматично переходимо у вікно пошуку та сполучення Bluetooth пристроїв. Вибираємо керований пристрій (при першому запуску PIN: 1234 або 0000). Якщо керований пристрій підключено, натискаємо значок Ø з'єднання Bluetooth у вікні програми. Вибираємо ведений пристрій, який тепер має підключитися.

Якщо підключення відбулося, світлодіод на веденому пристрої світиться безперервно.

Переходимо до керування. Балансуюча платформа знаходиться у робочому стані. Вибраємо керування кнопками-стрілками, якщо натиснемо кнопку Button. Вибраємо керування нахилом смартфона, якщо натиснемо на кнопку Gravity.

Натискаємо на кнопку Control і тепер можемо керувати рухом платформи, що балансує, обраним раніше способом.

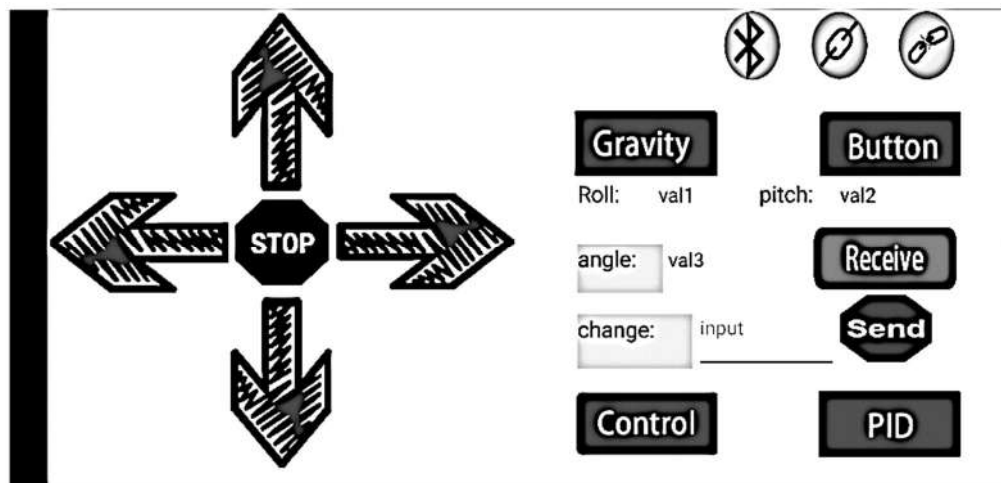


Рисунок 5.1 – Вікно програми для керування платформою, що балансує.

Для тестування каналу зв'язку між провідним і веденим пристроями, у Додатку 2 наведено код управління напрямом руху балансуючої платформи (вперед, назад, стоп).

5.5 Модуль керування балансуючої платформи

Програма управління роботизованою двоколісною балансувальною платформою реалізована в середовищі Arduino IDE (версія 2.3.7) і складається з 1-го модуля «Control_loop_release_6.ino», див. Додаток 3. В основу програми управління покладено програмне забезпечення, розроблене «keyestudio» [12] в

проекті «keyestudio Self-balancing Car Kit». Базову програму доопрацьовано, визначено семантичні помилки, внесено зміни до бібліотечних модулів.

6 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ

При проектуванні, складанні та реалізації електронних модулів у домашніх умовах охорона праці закриває всі ключові ризики: від роботи за комп'ютером до паяння та запуску готового пристрою [8].

6.1 Вимоги безпеки при проектуванні та складанні електронних модулів у домашніх умовах

6.1.1 Загальні вимоги охорони праці

6.1.1.1. До самостійного виконання робіт допускаються особи, які вивчили пристрій устаткування, що використовується, правила роботи з токсичними речовинами (свинець) і заходи першої допомоги.

6.1.1.2. небезпечними та шкідливими факторами при виконанні цього виду робіт є:

- підвищена температура елементів портативного паяльника (до 450 °C);
- наявність свинець містять аерозолів і парів флюсу в зоні дихання;
- механічне травмування очей обрізками висновків радіоелементів;
- Підвищена зорова напруга та статичне навантаження під час роботи за ПК.

6.1.1.3. Через використання свинець містить припою категорично забороняється прийом їжі, зберігання продуктів живлення та питної води на робочому місці.

6.1.1.4. Робоча зона має бути ізольована від доступу дітей та свійських тварин.

6.1.1.5. Працівник зобов'язаний забезпечити наявність первинних засобів пожежогасіння та аптечки першої допомоги (з наявністю протиопікових засобів).

6.1. 2 Вимоги охорони праці перед початком роботи

6.1.2.1. Включити систему місцевої витяжної вентиляції (димоуловлювач) або відкрити вікно для забезпечення притоку свіжого повітря.

6.1. 2.2. Звільнити робочу поверхню від сторонніх предметів, паперів та легкозаймистих матеріалів.

6.1. 2.3. Розкласти інструмент, витратні матеріали (припій, флюс) та об'єкт розробки (плату) у безпечному та зручному для роботи порядку.

6.1. 2.4. Здійснити зовнішній огляд USB-паяльника, кабелю живлення та джерела (ноутбука/Power Bank) на відсутність видимих пошкоджень ізоляції.

6.1. 2.5. Підготувати захисні окуляри та одягти робочий одяг з бавовняної тканини (не синтетику).

6.1. 3 Вимоги охорони праці під час роботи

6.1.3.1. При проектуванні на ПК (ноутбуці):

Дотримуватися оптимальної відстані від очей до екрана монітора (50-70 см).

Встановити регламентовані перерви тривалістю 5 хвилин через кожні 45 хвилин безперервної роботи для зниження зорової та статичної втоми.

6.1.3.2. Під час виконання паяльних робіт:

Підключення USB-паяльника до джерела живлення проводити безпосередньо перед початком процесу паяння.

Розміщувати кабель живлення USB так, щоб унеможливити його випадкове зачеплення, натяг або контакт з нагрітим жалом.

У технологічних паузах розміщувати нагрітий паяльник виключно на спеціалізованій термостійкій підставці.

Забороняється залишати підключений до джерела живлення паяльник без безперервного візуального контролю.

Фіксацію монтажної плати проводити за допомогою механічних затискачів (тримачів типу "третя рука"). Паяння елементів на вазі заборонено.

Видалення надлишків припою з жалу проводити лише з використанням спеціальної целюлозної губки або металевої стружки.

При формуванні та обрізанні висновків радіоелементів бокорізами використовувати захисні окуляри. Притримувати відкушувану частину пальцем, щоб уникнути її відлітання.

6.1.4 Вимоги охорони праці в аварійних ситуаціях

6.1.4.1. У разі виявлення несправності USB-паяльника (іскріння, запах горілої ізоляції, неконтрольований перегрів) негайно відключити його від USB-порту.

6.1.4.2. При термічному опіку: припинити роботу, охолодити уражену ділянку проточною холодною водою протягом 10-15 хвилин, обробити протиопіковим засобом з аптечки. При тяжкому опіку викликати швидку допомогу.

6.1.4.3. При попаданні залишків припою або флюсу у вічі: негайно промити їх великою кількістю чистої води та звернутися до лікаря.

6.1.4.4. При загорянні: відключити ноутбук/джерело живлення від мережі, локалізувати осередок горіння за допомогою щільної тканини, піску або вогнегасника.

6.1.5 Вимоги охорони праці після закінчення робіт

6.1.5.1. Вимкнути USB-паяльник від джерела живлення та дочекатися його повного природного охолодження перед збиранням на зберігання.

6.1.5.2. Навести лад на робочому місці. Збір дрібних металевих відходів, обрізків та залишків припою проводити за допомогою вологої серветки.

6.1.5.3. Закрити флакони з флюсом та хімічними очищувачами.

6.1.5.4. Ретельно вимити руки та обличчя теплою водою з милом для повного видалення можливих мікрочастинок токсичного свинцю.

6.1.5.5. Здійснити наскрізне провітрювання робочої кімнати протягом 10-15 хвилин.

6.2 Вимоги безпеки при роботі з літій-іонними (li-ion) акумуляторами

6.2.1 Загальні вимоги безпеки

6.2.1.1. Літій-іонні акумулятори є джерелами підвищеної пожежо- та вибухонебезпечності. При їх пошкодженні або короткому замиканні можливий лавиноподібний розігрів (термічний розгін) з виділенням токсичних газів і викидом полум'я.

6.2.1.2. Небезпечними факторами при роботі з Li- ion елементами є:

- Коротке замикання полюсів (викликає миттєве нагрівання та вибух).
- Теплова дія (перегрів жалом паяльника вище 60 °C).
- механічне пошкодження (прокол, деформація, падіння).
- Хімічний опік при розгерметизації корпусу та витоку електроліту.

6.2.2 Вимоги безпеки перед початком роботи

6.2.2.1. Здійснити візуальний контроль акумуляторів. Елементи з ознаками здуття, вм'ятин, іржі або пошкодження ізолюючої термозбіжної плівки до роботи не допускаються.

6.2.2.2. Виміряти напругу на елементах мультиметром. Забороняється використовувати і намагатися заряджати акумулятори, розряджені нижче за критичний рівень (менше 2.5 В).

6.2.2.3. Підготувати зону екстреної евакуації пошкодженого акумулятора (наприклад, порожню керамічну або металеву ємність).

6.2.3 Вимоги безпеки під час роботи

6.2.3.1. Забороняється пряме паяння проводів або висновків безпосередньо до корпусу (полюсів) Li- іон акумулятора звичайним паяльником. Локальний перегрів руйнує внутрішній сепаратор елемента.

6.2.3.2. Для підключення акумулятора до плати використовувати:

- акумулятори з вже привареними в заводських умовах нікелевими стрічками (паяння проводити тільки до вільного кінця стрічки на відстані від корпусу);
- Спеціальні пластикові батарейні відсіки (холдери).

6.2.3.3. Ізолювати інструмент. При монтажі плат із встановленим акумулятором використовувати пінцети із ізольованими наконечниками, щоб уникнути випадкового замикання доріжок.

6.2.3.4. Забороняється залишати плату з підключеним Li- іон акумулятором без нагляду під час першого тестового запуску та процесу заряджання.

6.2.3.5. Не допускайте переполюсування (плутанини «плюсу» та «мінуса») при підключенні акумулятора до схеми або модуля заряджання.

6.2.4 Вимоги безпеки в аварійних ситуаціях

6.2.4.1. При сильному нагріванні, шипінні або виділенні диму з акумулятора:

- негайно знеструмити схему (вимкнути USB-паяльник/живлення);

- Використовуючи пінцет або термостійку рукавичку, швидко перемістити акумулятор у металеву/керамічну ємність та винести у безпечне місце (на балкон або вулицю);

- Не вдихати дим, що виділяється (він вкрай токсичний).

6.2.4.2. При загорянні Li- іон акумулятора:

- Категорично забороняється гасити його малими об'ємами води. Літій вступає у бурхливу реакцію з водою з виділенням водню;

- Для гасіння використовувати спеціалізований порошковий вогнегасник (клас D/ABC) або повністю засипати вогнище сухим піском/землею для припинення доступу кисню.

6.2.4.3. При попаданні електроліту на шкіру: негайно промити уражену ділянку великою кількістю проточної води та обробити слабким розчином соди чи борної кислоти.

6.2.5 Вимоги безпеки після закінчення робіт

6.2.5.1. Вийняти акумулятори з макетних плат або холдерів , якщо пристрій залишиться на тривале зберігання без експлуатації.

6.2.5.2. Наклеїти ізолюючий скотч (каптоновий або малярний) на відкриті контакти/полюси акумуляторів, що не використовуються, для виключення їх випадкового замикання при зберіганні.

6.2.5.3. Відпрацьовані або пошкоджені акумулятори скласти в безпечну тару окрему. Забороняється викидати їх із побутовим сміттям, вони підлягають здачі у спеціалізовані пункти утилізації.

6.3 Вимоги безпеки при першому включенні (подачі живлення) на зібрану плату

6.3.1 Загальні вимоги безпеки

6.3.1.1. Перше включення є найнебезпечнішим етапом розробки через ризик прихованих дефектів монтажу (короткі замикання, переполюсування елементів, дефекти паяння).

6.3.1.2. Небезпечними факторами при першому включенні є:

- вибуховий характер руйнування компонентів (електролітичні конденсатори, мікросхеми) з розльотом уламків.
- Виділення токсичного диму при вигоранні доріжок плати та напівпровідників.
- Термічні опіки під час спроби перевірити нагрівання компонентів пальцями.

6.3.2 Вимоги безпеки перед початком роботи (перевірка)

6.3.2.1. Візуальний контроль: очистити плату від залишків флюсу та бризок припою. Перевірити відсутність «соплів» (мостів із припою) між сусідніми доріжками та висновками компонентів під лупою чи мікроскопом.

6.3.2.2. Перевірка полярності: переконатися у правильній орієнтації полярних компонентів (конденсатори, діоди, мікросхеми) та правильності підключення ліній живлення (+/GND).

6.3.2.3. Інструментальний контроль: обов'язково перевести мультиметр у режим продзвонювання та виміряти опір між шинами живлення (плюсом та мінусом) на вході плати. Зумер мультиметра не повинен сигналізувати про коротке замикання (КЗ).

6.3.2.4. Захист обличчя: одягнути захисні окуляри. При подачі живлення на плату із прихованим дефектом електролітичні конденсатори можуть миттєво вибухнути.

6.3.3 Вимоги безпеки під час подачі живлення

6.3.3.1. Обмеження струму: по можливості здійснювати перше включення від лабораторного блоку живлення з мінімальним обмеженням по струму, або через захисний резистор/запобіжник. Перший запуск забороняється безпосередньо від потужних акумуляторів.

6.3.3.2. Дистанція: під час безпосередньої подачі живлення (підключення кабелю або натискання кнопки) не нахиляйтеся над платою. Особа повинна знаходитись на безпечній відстані (не менше 40–50 см) та у захисних окулярах.

6.3.3.3. Контроль струму: відразу після включення зафіксувати показання споживаного струму. Аномально високий струм – ознака прихованого КЗ.

6.3.3.4. Контроль температури : Забороняється перевіряти нагрівання елементів пальцями під час живлення. Для фіксації перегріву використовувати безконтактний метод (пірометр, тепловізор) або короточасний дотик дотику тільки після повного відключення живлення.

6.3.4 Вимоги безпеки в аварійних ситуаціях

6.3.4.1. При появі диму, іскор, характерного тріску або запаху гару:

- Негайно знеструмити плату (висмикнути кабель живлення, вимкнути тумблер).
- Не торкатися елементів плати протягом 1 – 2 хвилин, оскільки дефектні компоненти можуть залишатися розжареними.
- Ретельно провітрити приміщення.

6.3.4.2. При руйнуванні (вибуху) конденсатора: відключити живлення, дочекатися осідання суспензії, зібрати осколки вологою серветкою, при попаданні частинок у вічі – негайно промити їх водою.

6.3.5 Вимоги безпеки після закінчення робіт

6.3.5.1. Перед проведенням будь-яких доопрацювань, усуненням виявлених помилок паяння або заміною компонентів обов'язково повністю відключати джерело живлення від плати.

6.3.5.2. Після відключення живлення почекати 15-30 секунд для безпечного саморозряду високоємних конденсаторів схеми, перш ніж брати плату руками.

7 ЕКОНОМІЧНІ РОЗРАХУНКИ

7.1 Оцінка витрат на проектування

Витрати проектування включають оплату інтелектуальної праці та амортизацію інструментів.

7.1.1 Розрахунок трудомісткості та фонду оплати праці (ФОП)

Розбиваємо процес розробки на етапи та оцінюємо затрачений час у годинах (t_i), табл. 7.1.

$$Z_i = t_i \cdot R_{h_i} \quad (7.1)$$

де t_i - витрачений час у годинах; R_{h_i} – умовна годинна ставка спеціаліста з проектування електроніки та програмування вбудованих систем (у Східній Європі), рівня Junior (початківець, до 2 років) становить у 2026 році: 670 –1300 грн /год; Z_i - Зарплата i -го етапу проектування.

Таблиця 7.1 - Оплата інтелектуальної праці

№	Час (t_i), година	Ставка (R_{h_i}), грн /година	Сума (Z_i), грн	Етап проектування
1	10	800	8000	Розробка технічного завдання та архітектури
2	25	800	20000	Розробка монтажною плати

3	20	800	16000	Розробка схеми електричної принципової
4	40	1200	48000	Написання програмного забезпечення (прошивки)
5	15	100	1500	Моделювання корпусу
	110		93500	Разом основна зарплата (Z_{osn})

Основна зарплата (Z_{osn}):

$$Z_{osn} = \sum_1^n t_i \cdot R_{h_i} \quad (7.2)$$

$$Z_{osn} = 93500, \text{ грн}$$

Додаткова зарплата (Z_{dor}): Резерв на відпустки/премії. Приймається як 10% від Z_{osn} .

$$Z_{dor} = 0.1 \cdot Z_{osn} \quad (7.3)$$

$$Z_{dor} = 9350, \text{ грн}$$

Страхові внески (S_{soc}): Обов'язкові відрахування 22% від загального ФОП.

$$S_{soc} = (Z_{osn} + Z_{dor}) \cdot 0.22 \quad (7.4)$$

$$S_{soc} = 22627, \text{ грн}$$

7.1.2 Розрахунок накладних витрат та амортизації

Витрати електроенергію, інтернет, знос ПК, паяльної станції та ліцензії ПЗ зазвичай беруться спрощено, як 40 – 60% від Z_{osn} .

$$OVR = Z_{osn} \cdot 0.5 \quad (7.5)$$

$$OVR = 46750, \text{ грн}$$

Разом витрати на проектування, грн .

$$C_{dev} = Z_{osn} + Z_{dop} + S_{soc} + OVR \quad (7.6)$$

$$C_{dev} = 172227, \text{ грн}$$

7.2 Витрати виготовлення прототипу (матеріальне втілення)

Тут відображаються фізичні витрати на складання одного працюючого зразка.

7.2.1 Матеріальні витрати

Враховуємо всі покупні компоненти. Сюди включається доставка.

Таблиця 7.2 - Витрати на покупні компоненти та доставка

№	Кількість (m_i), шт	Ціна за одиницю (c_i), грн	Усього (M_i), грн	Найменування компонента
1	1	70.00	70.00	Драйвер двигуна TB6612FNG Arduino
2	1	100.00	100.00	Гіроскоп акселерометр GY-521, MPU-6050 Arduino
3	1	75.00	75.00	SPP-C Bluetooth модуль з адаптером SPPC для Arduino (аналог HC-05 HC-06)

4	1	23.00	23.00	DC-DC знижувальний перетворювач 4.5-18 В до 0.8-17 В 3А
5	1	7.50	7.50	Ємнісна сенсорна кнопка ТТР223В Arduino (датчик натискання)
6	1	20.00	20.00	РСВ 5х7 двостороння друкована плата
7	1	45.00	45.00	Модуль живлення макетних плат МВ102 Arduino
8	2	15.00	30.00	Перетворювач логічних рівнів ІІС I2C UART SPI 5В 3.3В, 4 канали, двонаправлений
9	1	179.00	179.00	Мікроконтролерна платформа Arduino Nano V3.0 AVR АТmega328Р
10	1	39.00	39.00	Відсік елементів живлення 18650 для трьох акумуляторів
11	2	496.00	992.00	Двигун постійного струму CHR- GM37-520 з датчиком Холла, з утримувачем, з колесом
12	6	2.50	15.00	Стійка різьбова шестигранна , латунь, М3 х 25 мм
13	4	15.00	60.00	Стійка різьбова шестигранна , латунь, М3 х 40 мм
14	10	0.11	1.10	Гайка М3 шестигранна DIN934 6, цинк , К
15	20	0.34	6.80	Гвинт М3х8 DIN967 кл.міц.4,6 цинк М

16	1	105.00	105.00	Доставка «Нова Пошта»
			1768.40	Разом вартість матеріалів (M_{sum})

Разом вартість матеріалів:

$$M_{sum} = \sum_1^n c_i \cdot m_i \quad (7.7)$$

$$M_{sum} = 1768.40, \text{ грн}$$

7.2.2 Розрахунок витрат на складання та налагодження прототипу (Z_{sb})

Оцінюється час, витрачений вручну на монтаж компонентів (паяння) та первинне тестування заліза.

Час на паяння та складання:

$$Z_{sb} = t_{sb} \cdot R_{h_{sb}} \quad (7.8)$$

де t_{sb} - Витрати часу на складання прототипу; $R_{h_{sb}}$ — годинна ставка

$$t_{sb} = 5 \text{ годин}$$

$$R_{h_{sb}} = 700 \text{ грн/година}$$

$$Z_{sb} = 3500, \text{ грн}$$

Внески зі складання : відрахування 22% від Z_{sb} :

$$S_{sb_{soc}} = 0.22 \cdot Z_{sb} \quad (7.9)$$

$$S_{sb_{soc}} = 770, \text{ грн}$$

Разом витрати на прототип:

$$C_{proto} = M_{sum} + Z_{sb} + S_{sb_soc} \quad (7.10)$$

$$C_{proto} = 6038 \text{ , грн}$$

7.3 Зведений кошторис проекту (Повна собівартість)

Для фінального звіту об'єднуємо обидві групи витрат у єдину таблицю:

Таблиця 7.3 - Зведений кошторис проекту

Стаття витрат	Сума, грн	Частка у проекті, %
Витрати проектування	172227	96.6
Витрати створення прототипу	6038	3.4
Повна собівартість розробки	178265	100.0

Висновок. Повна собівартість проектування та створення працюючого прототипу балансує платформи склала 178 265 грн. Ця пропорція витрат типова для хардверних стартапів: 90-95% вартості першого пристрою - це інтелектуальна праця інженера, і лише мала частина йде на компоненти та інші матеріали.

7.4 Розрахунок точки беззбитковості

Розділимо витрати на постійні (один раз на весь проект) та змінні (при виробництві кожної одиниці товару). Також очистимо ринкову вартість від ПДВ, щоби оперувати чистими доходами компанії.

7.4.1 Вихідні дані для комерційного розрахунку

Ринкова ціна з ПДВ 20%:

$$P_{val} = 3000, \text{ грн}$$

Ціна без ПДВ:

$$P = \frac{P_{val}}{1.2} = 2500, \text{ грн}$$

Постійні затрати :

$$FC = C_{dev} = 172227, \text{ грн}$$

Змінні затрати на одиницю продукції – собівартість виробництва однієї серійної платформи, тобто. компоненти плюс збирання. Враховуємо, що вартість компонентів при оптовій закупівлі падає на 30-50%, а збирання автоматизується. Нехай під час закупівлі партій компонентів собівартість одиниці товару становитиме 1200 грн (без ПДВ).

$$VC = 1200, \text{ грн}$$

7.4.2 Розрахунок маржинального доходу на одиницю продукції

Розрахуємо суму, що залишається від продажу однієї платформи (після відрахування змінних витрат) на покриття постійних витрат (проектування) та формування прибутку:

$$MR = P - VC \quad (7.11)$$

$$MR = 1300, \text{ грн}$$

7.4.3 Розрахунок точки беззбитковості

Крапка беззбитковості показує скільки платформ потрібно продати, щоб вийти в нуль - повністю окупити затарати на розробку:

$$BEP_{units} = FC / MR \quad (7.12)$$

$$BEP_{units} = 132, \text{ прим.}$$

І в грошовому вираженні (виторг без ПДВ), грн

$$BEP_{money} = BEP_{units} \cdot P \quad (7.13)$$

$$BEP_{money} = 330000, \text{ грн}$$

З урахуванням ПДВ це складе в грошах:

$$BEP_{money_val} = BEP_{units} \cdot P_{val} \quad (7.14)$$

$$BEP_{money_val} = 396000, \text{ грн}$$

7.4.4 Розрахунок обсягу продажу для отримання планового прибутку

Припустимо, плануємо заробити чистий прибуток ($PP = 100000$, грн)
поверх вкладених сил. Розрахуємо необхідний обсяг продажів:

$$Q_{target} = \frac{FC + P}{MR} \quad (7.15)$$

$$Q_{target} = 210, \text{ шт}$$

Зведений фінансовий план проекту зведемо у табл. 7.4.

Таблиця 7.4 - Зведений фінансовий план проекту

Показник	Значення	Коментар
Обсяг продажів	210	Цільовий обсяг для отримання прибутку
Виручка загальна (з ПДВ)	630000	Гроші отримані від покупців ($Q_{target} * P_{val}$)
ПДВ (20%) до сплати	126000	Податок у бюджет
Виручка чиста (без ПДВ)	504000	Чистий дохід компанії
Постійні витрати (розробка)	172227	Окупність затрат на проектування
Змінні витрати (Виробництво)	252000	Вартість деталей та складання ($Q_{target} * VC$)
Чистий прибуток	79773	Чистий прибуток після всіх відрахувань

Висновки. Щоб повністю окупити створення платформи необхідно продати мінімальну партію: 210, шт. Кожен наступний продаж приносить 1300 грн прибутку (після вирахування податків та вартості деталей).

8 ВИСНОВКИ

На підставі завдання на дипломний проект розроблено та реалізовано роботизовану балансуєчу двоколісну платформу.

Розроблена система є автономною мобільною платформою, яка утримує рівновагу на двох колесах. Дистанційне керування дозволяє оператору переміщувати та позиціонувати платформу в просторі на площині бездротовим каналом Bluetooth. Веденим пристроєм є платформа, що сама балансує. Провідним пристроєм є смартфон із встановленим додатком Android. За протоколом стандартного послідовного інтерфейсу через додаток провідного пристрою передаються команди на рух роботизованої платформи вперед, назад, стоп, поворот ліворуч, поворот праворуч. При цьому платформа утримує рівновагу. Для забезпечення рівноваги платформи, як у стані спокою, так і в режимі пересування, в основній програмі управління передбачено реалізацію 3-х регуляторів:

1. ПД регулятор кута нахилу балансуєчої платформи; дозволяє утримувати платформу у вертикальному положенні;
2. ПІ регулятор швидкості руху платформи; дозволяє протидіяти зовнішнім силовим впливам, що зміщує платформу щодо поточної позиції в просторі.
3. ПД регулятор кермового управління; дозволяє демпфувати кут повороту, виключаючи перерегулювання при вимушеному русі з поворотами.

У проекті обрано комплекс технічних засобів, розроблена технічна документація системи, необхідна для реалізації балансуєчої платформи. На основі базового програмного забезпечення розроблено та протестовано коректне програмне забезпечення, що дозволяє вирішувати поставлені завдання.

ЛІТЕРАТУРА

1. TB6612FNG. Driver IC for Dual DC motor. Toshiba Bi-CD Integrated Circuit Silicon Monolithic.
<https://cdn.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf>
2. MPU-6000 and MPU-6050 / Product Specification / Revision 3.4 / InvenSense Inc. <https://d17t6iyxenbwp1.cloudfront.net/s3fs-public/2026-05/ps-mpu-6000a-00-mpu-6000-and-mpu-6050-datasheet-v3.4.pdf?VersionId=oAn7spWqbpWdJ1TZCrAD6CSMkAwtsCXA>
3. Bluetooth SPP Serial Port Transparent Transmission Module / JDY-31 Bluetooth Backplane User Manual. https://adastra-soft.com/wp-content/uploads/2021/06/JDY-31_manual_2.pdf
4. MP2315 / High Efficiency 3A, 24V, 500kHz / Synchronous Step Down Converter / MPS / Datasheet.
https://www.monolithicpower.com/en/documentview/productdocument/index/version/2/document_type/Datasheet/lang/en/sku/MP2315/document_id/513/
5. Arduino® Nano. User Manual / SKU: A000005.
<https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>
6. ATmega328P / 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash / DATASHEET.
https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
7. Додаток Android для дистанційного керування платформою, що балансує. Balance_car_Keyes.apk / Android Package / Owner: keyestudio925 / Created Mar 19, 2019 / Size 4.2 MB.
https://drive.google.com/open?id=1L69xdfmOvfhg0Wjh_p2FrtRkhs1h0fEu
[Дата звернення: 04.05.2026]

8. Google. (2026). AI Overview for "Охорона праці при проектуванні, збиранні та реалізації електронних пристроїв у домашніх умовах" [Large language model]. <https://www.google.com>. [Дата звернення: 19.05.2026]
9. НПАОП Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Мінсоцполітики України; Наказ, Вимоги від 14.02.2018 № 207. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18>
10. Arduino. (n.d.). Language Reference. Arduino Documentation. Retrieved [03,01,2026], from <https://docs.arduino.cc/language-reference/>
11. Fritzing GmbH. Learning. Fritzing Documentation. Retrieved [04,08,2026], from <https://fritzing.org/learning/>
12. Keystudio. Products / keystudio Self-balancing Car Kit For Arduino Robot Kit. Retrieved [03,15,2026], from <https://www.keystudio.com/products/keystudio-self-balancing-car-kit-for-arduino-robot-kit>

ДОДАТОК 1 – Програма тестування інерційного модуля MPU-6050

```
#include <Wire.h>

const int MPU_addr=0x 68; // I2C адреса MPU6050
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

void setup( ) {
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B ); // Реєстр управління живленням PWR_MGMT_1
Wire.write(0 ); // Будимо MPU6050 (записуємо 0)
Wire.endTransmission(true);
Serial.begin(9600);
}

void loop( ){
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B); // Починаючи з цього регістру лежать дані
акселерометра
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr, 14, true ); // Запитуємо 14 регістрів (по 2
байти на параметр)

// Читаємо дані (склеюємо старший і молодший байти)
AcX = Wire.read ()<< 8 | Wire.read();
AcY = Wire.read ()<< 8 | Wire.read();
AcZ = Wire.read ()<< 8 | Wire.read();
Tmp = Wire.read ()<< 8 | Wire.read(); // Температура
```

```
GyX = Wire.read () << 8 | Wire.read();
GyY = Wire.read () << 8 | Wire.read();
GyZ = Wire.read () << 8 | Wire.read();

// Виводимо у порт
Serial.print("Acc:");
Serial.print(AcX); Serial.print("\t");
Serial.print(AcY); Serial.print("\t");
Serial.println(AcZ);
Serial.print("Gyro:");
Serial.print(GyX); Serial.print("\t");
Serial.print(GyY); Serial.print("\t");
Serial.println(GyZ);

delay (345);
}
```

ДОДАТОК 2 – Програма тестування каналу зв'язку через Bluetooth

```
// Виводи керування на входи TB6612
const int right_R1 = 8;
const int right_R2=12;
const int PWM_R=10;
const int left_L1 = 7;
const int left_L2 = 6;
const int PWM_L=9;
char val; // Змінна Bluetooth

void setup( )
{
  Serial.begin(9600);
  // Налаштувати виводи управління на вихід
  pinMode(right_R 1,OUTPUT );
  pinMode(right_R 2,OUTPUT );
  pinMode( PWM_R,OUTPUT);
  pinMode(left_L 1,OUTPUT );
  pinMode(left_L 2,OUTPUT );
  pinMode( PWM_L,OUTPUT);
}

void loop( )
{
  if( Serial.available()) // якщо значення послідовного буфера доступне
  {
    val = Serial.read(); // присвоїти значення, зчитане з послідовного
    порту, змінної val
    Serial.println(val);
    switch (val) // оператор вибору
```

```

{
    front( ); break; // рух вперед
    case 'B': back( ); break; // рух тому
    case 'S': Stop( ); break; // стоп
}
}
}
void front( ) // реалізація руху вперед
{
digitalWrite(right_R 1, HIGH );
digitalWrite(right_R 2,LOW );
digitalWrite(left_L 1, HIGH );
digitalWrite(left_L 2,LOW );
analogWrite( PWM_R,100);
analogWrite( PWM_L,100);
}
void back( ) // реалізація руху назад
{
digitalWrite(right_R 1,LOW );
digitalWrite(right_R 2, HIGH );
digitalWrite(left_L 1,LOW );
digitalWrite(left_L 2, HIGH );
analogWrite( PWM_R,100);
analogWrite( PWM_L,100);
}
void Stop( ) // Стоп машина
{
digitalWrite(right_R 1,LOW );
digitalWrite(right_R 2, HIGH );

```

```
digitalWrite(left_L 1,LOW );  
digitalWrite(left_L 2, HIGH );  
analogWrite( PWM_R,0);  
analogWrite( PWM_L,0);  
}
```

ДОДАТОК 3 – Основна програма управління платформою балансування

```
#include "MsTimer2.h" // Внутрішній таймер
#include <PinChangeInt.h> //задіє всі переривання на платформі
#include <MPU6050.h> //MPU6050 бібліотека
#include <Wire.h> //I2C бібліотека

MPU6050 MPU 6050; //Створення об'єкта mpu6050 класу MPU6050
int16_t ax, ay, az, gx, gy, gz; //визначаємо змінні по 3-м осях
акселерометра, та гіроскопа

//Дискретні виводи Arduino для підключення
//драйвера TB6612
const int right_R1 = 8;
const int right_R2=12;
const int PWM_R=10;
const int left_L1 = 7;
const int left_L2 = 6;
const int PWM_L=9;

////////////////////// Кутові параметри//////////////////////
float angle_X; // Обчислює змінну кута нахилу осі X за допомогою
акселерометра
float angle_Y; // Обчислює змінну кута нахилу осі Y за допомогою
//акселерометра
float angle0 = 4.5; // = 1; // Фактично виміряний кут (в ідеалі 0
градусів)
float Gyro_x , Gyro_y , Gyro_z; // Кутова швидкість для розрахунку
гіроскопа
```

```

////////////////////// Кутові параметри//////////////////////

//////////////////////фільтр Калмана//////////////////////
float Q_angle = 0.001; // Коваріація шуму гіроскопа
float Q_gyro = 0.003 ; // Коваріація шуму дрейфу гіроскопа
float R_angle = 0.5; // Коваріація акселерометра
char C_0 = 1;
float dt=0.005; // Період часу дискретизації фільтра
float K1 = 0.05 ; // Функція, що містить коефіцієнт посилення Калмана,
// використовується для обчислення відхилення оптимальної оцінки
float K_0 ,K_1 ,t_0,t_1;
float angle_err;
float q_bias ; // дрейф гіроскопа

float accelz = 0;
float angle;
float angleY_one;
float angle_speed;

float Pdot [ 4] = {0, 0, 0, 0};
float P [ 2] [2] = {{1, 0}, {0, 1}};
float PCt_0, PCt_1, E;
//////////////////////фільтр Калмана//////////////////////

//////////////////////PID параметри//////////////////////
// залишив І-складник в обробнику переривання, прибрав таймінги для
//I, підправив бібліотеку "MsTimer2.h"
double kp = 40, ki = 0.5, kd = 0.3 ; // Параметри контуру кута нахилу,
d_time = 10 мс

```

```

//-----+-----
double kp_speed=3, ki_speed=0.2, kd_speed=0; //Параметри контуру
швидкості
//-----
double kp_turn = 24, ki_turn = 0, kd_turn = 0.08 ; // Параметри контуру
рульового управління
//-----
double setp0 = 0; //Кут балансу
int PD_pwm; // вихід контуру кута нахилу
float pwm1 = 0, pwm 2 = 0;
float integral = 0;
int d_time = 10; // Період циклу розрахунку управління

////////////////////Вимірювання швидкості переривань////////////////////
#define PinA_left 5 / / Зовнішні переривання
#define PinA_right 4 //Зовнішні переривання
#define PinB_left 2 // Внутрішнє переривання INT0
#define PinB_right 3 //Внутрішнє переривання INT1
volatile long count_right = 0; //Використовується для обчислення
значення імпульсу,
//Розрахований датчиком Холла (тип volatile long гарантує коректність
значення)
volatile long count_left = 0;

//////////////////// розрахунок імпульсу //////////////////////
int rpluse = 0;
int lpluse = 0;
volatile int pulseright, pulseleft;

```

```

////////////////////////// Параметри ПІ //////////////////
float speeds_filterold=0;
float positions=0;
int PI_pwm;
int cc;
int speedout;
float speeds_filter;

////////////////////////// рульове управління PD////////////////////////
int turnmax, turnmin , turnout;
float Turn_pwm = 0;
int zz = 0;
int turncc = 0;

//Bluetooth//
int front = 0; // змінна для повороту вперед
int back = 0; // змінна для повороту назад
int left = 0; // поворот наліво
int right = 0; // поворот праворуч
char val;

void setup( )
{
//set control control's pin to OUTPUT
pinMode(right_R 1,OUTPUT );
pinMode(right_R 2,OUTPUT );
pinMode(left_L 1,OUTPUT );
pinMode(left_L 2,OUTPUT );
pinMode( PWM_R,OUTPUT);

```

```

pinMode( PWM_L,OUTPUT);

//Initial state value
digitalWrite(right_R1,1);
digitalWrite(right_R2,0);
digitalWrite(left_L1,0);
digitalWrite(left_L2,1);
analogWrite( PWM_R,0);
analogWrite( PWM_L,0);

analogWrite( PWM_R,0);
analogWrite( PWM_L,0);

// Вхідний сигнал колеса з кодом швидкості
pinMode( PinA_left, INPUT);
pinMode( PinA_right, INPUT);
pinMode( PinB_left, INPUT);
pinMode( PinB_right, INPUT);

Wire.begin( ); // активація I2C
// Встановлює тайм-аут в 3 мс для запобігання зависанню.
Wire.setWireTimeout(3000, true);
Serial.begin(9600);
delay ( 1500);
mpu6050.initialize();
delay ( 2);

//Протягом 10 мс використовується таймер 2 для встановлення переривання
//таймера

```

//Примітка: використання таймера 2 вплине на ШІМ-вихід контакту //3 та контакту 11.

MsTimer 2: set(d_time, DSzhongduan); //10 мс виконати функцію
//DSzhongduan один раз

MsTimer 2:: start(); //запустити переривання
}

void loop()

{
// для тестування та налагодження

//Serial.print(angle_speed);

//Serial.print(" ");

//Serial.println(Gyro_x);

//Serial.print("angle=");

//Serial.println(angle);

//Delay (100);

//Serial.println(PD_pwm);

//Serial.println(pwm1);

//Serial.println(pwm2);

//Serial.print("pulseright = ");

//Serial.println(pulseright);

//Serial.print(" pulseleft = ");

//Serial.println(pulseleft);

//Serial.println(PI_pwm);

//Serial.println(speeds_filter);

//Serial.println (positions);

if(Serial.available())

{

```

val = Serial.read(); //вважаємо з послідовного порту
switch (val) // Оператор switch
{
case 'F': front=250; break; //якщо val дорівнює F, front=250,
//рухаємося вперед
case 'B': back=-250; break; // рухаємось тому
case 'L': left=1; break; // Повертаємо наліво
case 'R': right=1; break; // Поворот праворуч
case 'S': front= 0,back =0,left=0,right=0;break; // зупинка
case 'Q': Serial.print(angle );break ; //отримання 'D', відправка
//значення змінної кута додаток
    case 'P': kp=kp+0. 5,Serial.print (kp);break;
case 'O': kp=kp-0. 5,Serial.print (kp);break;
case 'I': kd=kd+0. 02,Serial.print (kd);break;
case 'U': kd=kd-0. 02,Serial.print (kd);break;
case 'Y':  kp_speed=kp_speed+0. 05,  Serial.print  (kp_speed);
break;
case 'T':  kp_speed=kp_speed-0. 05,  Serial.print  (kp_speed);
break;
case 'G':  ki_speed=ki_speed+0. 01,  Serial.print  (ki_speed);
break;
case 'H':  ki_speed=ki_speed-0. 01,  Serial.print  (ki_speed);
break;
case 'J': kp_turn=kp_turn+0. 4,Serial.print (kp_turn);break;
case 'K': kp_turn=kp_turn-0. 4,Serial.print (kp_turn);break;
case 'N': kd_turn=kd_turn+0. 01, Serial.print (kd_turn); break;
case 'M': kd_turn=kd_turn-0. 01, Serial.print (kd_turn); break;
}
}

```

```

//Зовнішнє переривання для розрахунку швидкості обертання колеса.
//Зміна рівня PinA_left викликає зовнішнє переривання;
//виконати підфункцію Code_left
attachInterrupt( digitalPinToInterrupt( 2), Code_left, RISING);
//Зміна рівня PinA_right викликає зовнішнє переривання;
//виконати підфункцію Code_right
attachInterrupt( digitalPinToInterrupt( 3), Code_right, RISING);

// для тестування і налагодження
//Serial.print("count_ left = ");
//Serial.println(count_left);
//Serial.print("count_right = ");
//Serial.println(count_right);
}

////////////////////// Розрахунок швидкості по датчиках
Холла//////////////////////
// Лічильник кодів швидкості лівої передачі
void Code_ left( )
{
if( digitalRead( 5) == LOW) count_left--; else count_left++;
}
// Лічильник кодів швидкості правої передачі
void Code_ right( )
{
if( digitalRead( 4) == LOW) count_right++; else count_right--;
}

```

```

////////// обчислення імпульсів //////////
void countpulse( )
{
lpluse = count_left; // Привласнюємо значення , підраховане кодовим //
колесом ;
rpluse = count_right;
// тому: якщо назад (ШИМ, тобто напруга двигуна //негативне), то число
імпульсів негативне
if ((pwm1 < 0 ) && (pwm2 < 0)) // визначаємо напрямок руху;
{
rpluse = rpluse;
lpluse = lpluse;
}
// якщо вперед (ШИМ, тобто напруга двигуна позитивна),
// то число імпульсів позитивне
else if ((pwm 1 > 0) && (pwm2 > 0))
{
rpluse = rpluse;
lpluse = lpluse;
}
else if ((pwm1 < 0 ) && (pwm2 > 0)) // Визначення напрямку повороту
//поворот ліворуч; номер правого імпульсу позитивний;
//Номер лівого імпульсу негативний.
{
rpluse = rpluse;
lpluse = -lpluse;
}
else if ((pwm 1 > 0) && (pwm2 < 0)) // Визначення напрямку повороту
//Поворот праворуч; номер правого імпульсу негативний;

```

```

//Номер лівого імпульсу позитивний.
{
rpluse = -rpluse;
lpluse = lpluse;
}

// Введення переривань кожні 10 мс;
pulseright += rpluse;
pulseleft += lpluse;

count_left = 0; // Очищаємо лічильник кодів
count_right = 0;
}

///// обробка переривань системного таймера
void DSzhongduan( )
{
    sei( ); //Allow overall interrupt
    countpulse( );
    mpu6050.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // отримуємо сирі
дані
    //get angle and Kalman_Filter
    angle_calculate ( ax, ay, az, gx, gy, gz, dt, Q_angle, Q_gyro,
R_angle, C_0, K1);
    PD( ); // Керуюча дія контуру кута нахилу
    anglePWM( );

    cc++;

```

```

if(cc>=4) //10*4=40 мс входить один раз алгоритм ПІ-регулювання
//швидкості
{
speedpiout( );
cc = 0; //Очистити
}

turncc++;
if(turncc>8) // 10 * 8 = 80 мс, введення лічильника ПД-регулювання
рульового // управління
{
turnspin( );
turncc = 0; //Очистити
}
}

////////////////////// Кутові розрахунки//////////////////////
void angle_ calculate( int16_t &ax,int16_t &ay,int16_t &az,int16_t
&gx,int16_t gy,int16_t gz,float dt,float Q_angle,float Q_gyro,float
R_angle,float C_0,float K1)
{
float Angle = -atan2( ay, az) * (180/PI); // Формула розрахунку кута
// Радіального обертання;
//Негативний знак позначає обробку напряду
Gyro_x = -gx/131; // Кутова швидкість осі X, розрахована //гироскопом;
//Негативний знак позначає обробку напряду
Kalman_ Filter ( Angle, Gyro_x); // Фільтр Калмана

// Параметр кута обертання по осі Z

```

```

Gyro_z=-gz/131; //Куттова швидкість по осі Z
//accelz=az/16.4;
}

//////////////////////////////////KalmanFilter//////////////////////////////////
void Kalman_Filter ( double angle_m, double gyro_m)
{
angle + = (gyro_m - q_bias) * dt; //Попередня оцінка
angle_err = angle_m - angle;
//Диференціал коваріації помилки азимуту
Pdot [ 0] = Q_angle - P [0] [1] - P [1] [0];
Pdot [ 1] = - P [1] [1];
Pdot [ 2] = - P [1] [1];
Pdot [ 3] = Q_gyro;
//Апріорна оцінка помилки коваріації диференціального інтегралу
P [ 0] [0] + = Pdot [0] * dt;
P [ 0] [1] + = Pdot [1] * dt;
P [ 1] [0] + = Pdot [2] * dt;
P [ 1] [1] + = Pdot [3] * dt;

//Проміжна змінна матричного множення
PCt_0 = C_0 * P [ 0] [0];
PCt_1 = C_0 * P [ 1] [0];
//Знаменник
E = R_angle + C_0 * PCt_0;
//Поліпшити оцінку
K_0 = PCt_0/E;
K_1 = PCt_1/E;
//Проміжна змінна матричного множення

```

```

t_0 = PCt_0;
t_1 = C_0 * P [ 0] [1];
//Коваріація помилки апостеріорної оцінки
P [ 0] [0] - = K_0 * t_0;
P [ 0] [1] - = K_0 * t_1;
P [ 1] [0] - = K_1 * t_0;
P [ 1] [1] - = K_1 * t_1;

q_bias + = K_1 * angle_err; //Постеріорна оцінка
//Різниця вихідних значень дає оптимальну кутову швидкість
angle_speed = gyro_m - q_bias;
//Оцінка апостеріорного розподілу для отримання оптимального кута
angle + = K_0 * angle_err; //}
}

////////////////////////PID кута нахилу //////////////////////////
void PD( )
{
float error = angle + angle0;
integral + = error * ki; / / * pTime;
integral = constrain ( integral, - 255,255);
PD_pwm = kp * (error) + kd * angle_speed +
integral;
}

//////////////////////// швидкість PI////////////////////////
void speedpiout( )
{
// Значення імпульсу швидкості платформи

```

```

float speeds = (pulseleft + pulseright) * 2.0;
pulseright = pulseleft = 0; // Очищення
speeds_filterold *= 0.7; // Компліментарна фільтрація першого порядку
speeds_filter = speeds_filterold + speeds * 0.3;
speeds_filterold = speeds_filter;
positions += speeds_filter;
positions += front; // Злиття прямого управління
positions += back;
positions = constrain ( positions, -3550,3550); //Антиінтегральне
//насищення
// Управління швидкістю за ПІ-законом
PI_pwm = ki_speed * (setp0 - positions) + kp_speed * (setp0 -
speeds_filter);
}

//////////////////// швидкість ПІ////////////////////

//////////////////// Рульове управління////////////////

void turnspin( )
{
int flag = 0; // не визначили швидкість перед поворотом
float turnspeed = 0;
float rotationratio = 0;

if (left == 1 || right == 1) // повертаємо наліво або праворуч
{
if (flag == 0) // Визначити швидкість перед поворотом;
// Підвищити маневреність платформи.
{
turnspeed = (pulseright + pulseleft) * 2; //поточна швидкість

```

```

//вираз через pulse
flag=1; // Визначили швидкість перед поворотом
}
if (turnspeed < 0 ) // Абсолютне значення поточної швидкості
{
turnspeed = -turnspeed;
}
if(left==1||right==1) //якщо натиснута ліва чи права клавіша
//управлення (Bluetooth)
{
turnmax = 3; //максимальне значення повороту
turnmin = -3; //Мінімальне значення повороту
}
rotationratio = 5 / turnspeed; //встановлюємо значення за швидкістю
if (rotationratio < 0.5)
{
rotationratio = 0.5;
}
if (rotationratio > 5)
{
rotationratio = 5;
}
}
else
{
rotationratio = 0.5;
flag = 0; // не визначили швидкість перед поворотом
    turnspeed = 0; // немає повороту
}

```

```

if (left ==1)//збільшуємо за параметром напрямку
{
turnout + = rotationratio;
}
else if (right == 1 ) //збільшуємо за параметром напрямку
{
turnout -= rotationratio;
}
else turnout = 0;
// обмеження швидкості
if (turnout > turnmax) turnout = turnmax; // максимальне значення //
амплітуди
if (turnout <turnmin ) turnout = turnmin; // мінімальне значення //
амплітуди
// Алгоритм ПД-регулювання обертання керує швидкістю // накопичення
кута повороту та позиціонуванням обертання по осі Z
Turn_pwm = -turnout * kp_turn - Gyro_z * kd_turn ; }
//////////////////////////PWM керування //////////////////////////
void anglePWM( )
{
pwm2 = -PD_pwm + PI_pwm + Turn_pwm;
pwm1 = -PD_pwm + PI_pwm -Turn_pwm;

if(pwm1>255)
{
pwm1 = 255;
}
if(pwm1<-255)
{

```

```

pwm1=-255;
}
if (pwm2>255)
{
pwm2 = 255;
}
if (pwm2<-255)
{
pwm2=-255;
}
if (angle>45 || angle<-45)
{
pwm1 = pwm2 = 0;
}

if (pwm 2>= 0)
{
digitalWrite(left_L 1,LOW );
digitalWrite(left_L 2, HIGH );
analogWrite( PWM_L, pwm2);
}
else
{
digitalWrite(left_L 1, HIGH );
digitalWrite(left_L 2,LOW );
analogWrite( PWM_L,-pwm2);
}

if (pwm1>=0)

```

```
{  
digitalWrite(right_R 1,LOW );  
digitalWrite(right_R 2, HIGH );  
analogWrite( PWM_R,pwm1);  
}  
else  
{  
digitalWrite(right_R 1, HIGH );  
digitalWrite(right_R 2,LOW );  
analogWrite( PWM_R,-pwm1);  
}  
}  
  
// Кінець модуля
```

Позиція	Найменування і технічна характеристика устаткування і матеріалів завод-виготівник (для імпортного устаткування - країна, фірма)	Тип, марка устаткування. Позначення документа і N опитового листа	Одиниця вимірювання		Код заводу- виготівника	Код устаткування матеріалу	Ціна оддиниці тис. грн.	Кількість	Маса оддиниці устатку- вання, кг
			наймену- вання	код					
1	2	3	4	5	6	7	8	9	10
	Прилади і засоби автоматизації								
1	Триосьовий гіроскоп + триосьовий акселерометр на чіпі MPU-6050. Напруги живлення: 2,375...3,46 вольт. Споживаний струм: до 4 мА. Інтерфейс передачі даних: I2C. Максимальна швидкість I2C: 400 кГц. Китай.	GY-521	шт	796				1	
2	Bluetooth модуль з адаптером SPPC. Контролер: BEKEN BK3231. Інтерфейс RS232 Чутливість приймача: -84 dBm. Робоча частота: 2,4 ГГц. Напруга живлення: 3,6...6 В. Можливі швидкості передачі даних: 1200, ..., 115200 бод. Китай.	SPP-C	шт	796				1	
3	Датчик напрямку швидкості обертання. Подвійний магнітний кодер Холла. Коефіцієнт роздільної здатності - 16 імпульсів за оберт. Внутрішній діаметр магнітного кільця 0.5 мм. Штекер порту 2.54-4p Китай.	775-P16	шт	796				2	
4	Двоканальний драйвер двигуна постійного струму. Напруга моторів (VMOT): 4,5 13,5 В. Напруга логіки (VCC): 2,7...5,5 В. Вихідний безперервний струм: 1 А на кожен канал. Максимальна частота ШІМ: 100 кГц. Китай.	TB6612FNG	шт	796				1	
5	Редукторний двигун постійного струму. DC 12V 320 грт-і30. Китай.	CHR-GM37-520	шт	796				2	
6	Ємнісна сенсорна кнопка. Напруга живлення: 2.5-5 В. Час відгуку в активному режимі: 60 мс. Час відгуку в режимі зниженого енергоспоживання: 220 мс. Чутливість: 0 - 50 нФ. Китай.	TTP223B	шт	796				1	

				A26.13.APT.00.C			
Виконав	КРЯЖКОВ			Розробка та реалізація самобалансованого робота на колесах з дистанційним керуванням Специфікація устаткування на засоби автоматизації	Стадія	Лист	Листів
Перевірив	ЛОСІХІН				ДП	1	2
Рецензент	МИСОВ				УДУНТ, ННІ УДХТУ, каф.КІТмаР, гр.4-АВП-22		
Н.Контр.							
Затвердив	ЛЕВЧУК						

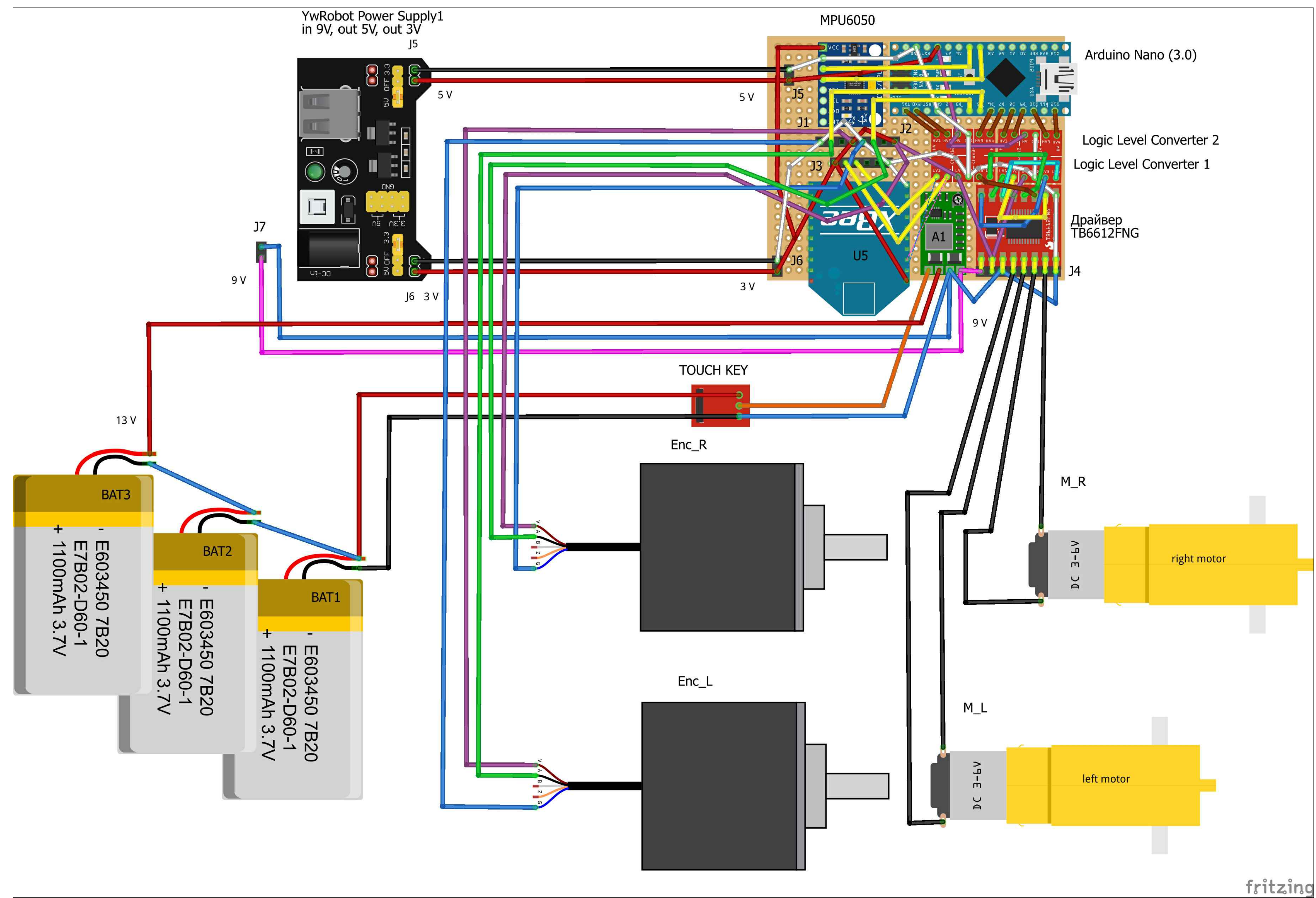
Позиція	Найменування і технічна характеристика устаткування і матеріалів завод-виготівник (для імпортного устаткування - країна, фірма)	Тип, марка устаткування. Позначення документа і N опитового листа	Одиниця вимірювання		Код заводу-виготівника	Код устаткування матеріалу	Ціна одиниці тис. грн.	Кількість	Маса одиниці устаткування, кг
			найменування	код					
1	2	3	4	5	6	7	8	9	10
7	Універсальна двостороння друкована плата. Матеріал: склопластик (склотекстоліт). Розмір плати 5x7 см. Товщина 1.6 мм. Діаметр отвору 1 мм. Крок між отворами 2.54 мм. Китай.	-	шт	796				1	
	<u>Засоби обчислювальної техніки</u>								
8	Мікроконтролерна платформа. Мікроконтролер: ATmega328P. Робоча напруга: 5 В. Тактова частота: 16 МГц. Китай.	Arduino Nano	шт	796				1	
	<u>Електроапаратура</u>								
9	Модуль живлення макетних плат. Вихідна напруга: 3.3 або 5 з обох сторін плати. Живлення постійним струмом напругою 6.5-12 В. Максимальний струм: 700 мА. Китай.	MB102	шт	796				1	
10	Двохнаправлений 4-канальний перетворювач логічних рівнів. Вхідна напруга 5 В. Вихідна напруга 3.3 В. Схема (https://arduino.ua/docs/Logic_Level_Bidirectional.pdf) Китай.	-	шт	796				2	
11	Зверхмініатюрний регульований імпульсний понижуючий перетворювач постійної напруги DC-DC step-down. Вхідна напруга: DC 4.5-18 В. Вихідна напруга: Можливість вибору фіксованої напруги (1,8 До 2,5 3,3 В, 5 В 9 До 12 В) або регульований діапазон (0,8-17). Вихідний струм: максимум 3А. Струм холостого ходу: 0,85 мА. Китай.	MP2315	шт	796				1	

A26.13.APT.00.C

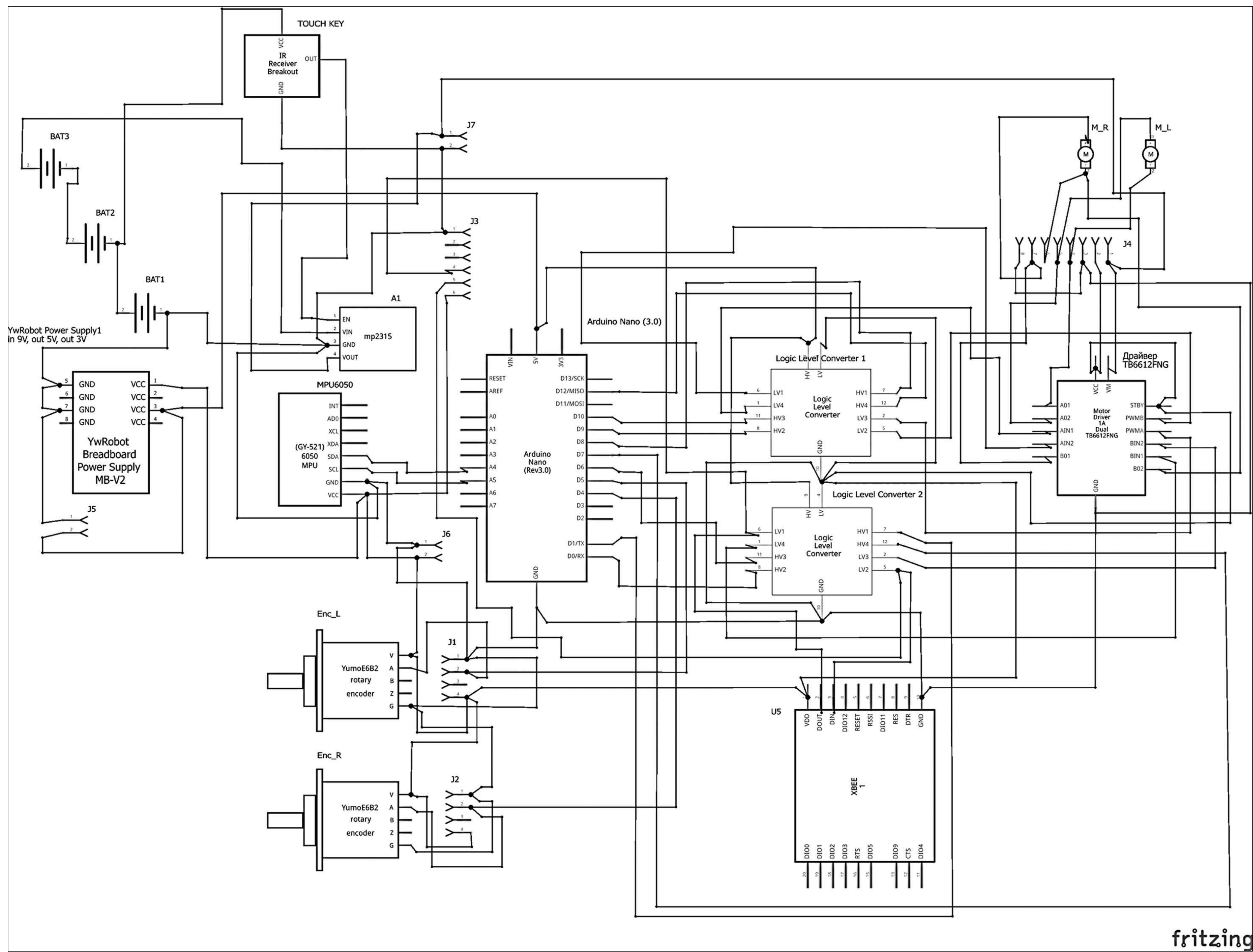
Лист
2

ЗМІСТ ТОМУ

№	Найменування	Позначення	Кількість листів
1	Титульний лист		1
2	Завдання на дипломний		1
	проект		
3	Відомість дипломного	A26.13.APT.00.ВД	1
	проекту		
4	Пояснювальна записка	A26.13.APT.00.ПЗ	88
5	Специфікація устаткування	A26.13.APT.00.C	2
	на засоби автоматизації		



A26.13.APT.01.CБ				
м. Дніпро, УДУНТ, ННІ УДХТУ				
Виконав	КРЯЖКОВ	Розробка та реалізація		Лист
Перевірив	ЛОСІХІН	самобалансуючого робота на колесах		1
Рецензент	МИСОВ	з дистанційним керуванням		1
Н. Контр.		Схема електрична принципова		УДУНТ, ННІ УДХТУ,
Затвердив	ЛЕВЧУК	підключень компонентів		каф. КІТтаР, гр. 4-АВП-22



					A26.13.APT.02.C5		
					м. Дніпро, УДУНТ, ННІ УДХТУ		
Виконав	КРЯЖКОВ				Розробка та реалізація самобалансуючого робота на колесах з дистанційним керуванням	Стадія	Лист
Перевірив	ЛОСІХІН					ДП	1
Рецензент	МИСОВ				Схема електрична принципова з'єднань компонентів	Листів	
Н. Контр.						УДУНТ, ННІ УДХТУ, каф. КІТтаР, гр. 4-АВІТ-22	
Затвердив	ЛЕВЧУК						1