

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут Електроенергетики
(інститут)

Електротехнічний факультет
(факультет)

Кафедра електропривода
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра
(бакалавра, спеціаліста, магістра)

студента Гуцько Максима Володимировича
(ПІБ)

академічної групи 141М-17-4
(шифр)

спеціальності 141 Електроенергетика, електротехніка та електромеханіка
(код і назва спеціальності)

спеціалізації¹ _____

за освітньо-професійною програмою Електромеханічні системи автоматизації та електропривод

(офіційна назва)

на тему Розробка автономного цифрового електропривода балансуючого візка

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	Яланський О.А.			
розділів:				
Технологічна частина	Яланський О.А.			
Автоматизований електропривод	Яланський О.А.			
Розробка програми балансуючого візка	Яланський О.А.			
Техніко-економічне обґрунтування	Тимошенко Л.В.			
Рецензент				
Нормоконтролер	Казачковський М.М.			

Дніпро
2018

ЗАТВЕРДЖЕНО:
завідувач кафедри
електропривода
(повна назва)

_____ Казачковський М.М.
(підпис) (прізвище, ініціали)

« _____ » _____ 20__ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра
(бакалавра, спеціаліста, магістра)

студенту Гулько М.В. академічної групи 141М-17-4
(прізвище та ініціали) (шифр)
спеціальності 141 Електроенергетика, електротехніка та електромеханіка

спеціалізації¹ _____

за освітньо-професійною програмою Електромеханічні системи автоматизації та електропривод
(офіційна назва)

на тему Розробка автономного цифрового електропривода балансуєчого візка

затверджену наказом ректора НТУ «Дніпровська політехніка» від _____ № _____

Розділ	Зміст	Термін виконання
Технологічна частина	Опис проекту, вибір його складових, вимоги до системи	3.10.2018- 6.10.2018
Автоматизований електропривод	Опис принципу дії балансування, приведення схем підключення обладнання	8.10.2018- 22.10.2018
Розробка програми балансуєчого візка	Описано середовище розробки програми та наведена програма роботи	25.10.2018- 3.12.2018
Техніко-економічне обґрунтування	Визначення витрат на створення проекту	6.12.2018- 08.12.2018

Завдання видано _____
(підпис керівника)

Яланський О.А.
(прізвище, ініціали)

Дата видачі 15 жовтня 2018

Дата подання до екзаменаційної комісії _____

Прийнято до виконання _____
(підпис студента)

Гулько М.В.
(прізвище, ініціали)

Реферат

Мета магістерської роботи:

Розробка автономного цифрового електропривода балансуєчого візка.

Магістерська робота складається з пояснювальної записки, 16 рисунків, 7 таблиць.

У першому розділі «Технологічна частина» описується принцип роботи балансуєчого візка, його апаратна частина. Зображено зовнішній вигляд балансуєчого візка та його компонентів, висунуті вимоги до системи.

У другому розділі «Автоматизований електропривод» описується принцип дії балансування, приводиться характеристика шини передачі даних між пристроями, приводиться схема підключень усієї апаратної частини системи, та двигунів.

У третьому розділі «Розробка програми роботи балансуєчого візка» описано середовище розробки програми для роботи та приведена програма з коментарями до неї.

П'ятий розділ, останній, «Техніко-економічне обґрунтування», дозволяє розрахувати власні витрати на проект та орієнтовну ціну виробу.

Ключові слова: електропривод, мікроконтролер, гіроскоп – акселерометр, балансуєчий візок, система керування.

					ЕП.МР.18.02.Р.ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Abstract

The design of the master's work:

Development of Autonomous Digital Electric Drive of a Balancing Trolley.

Master's work consists of an explanatory note, 16 figures, 7 tables.

In the first section "Technological part" describes the principle of balancing the cart, its hardware part. The appearance of the balancing trolley and its components is represented by the requirements of the system.

The second section, "Automated Electric Drive" describes the principle of balancing, describes the characteristics of the data transmission bus between the devices, provides a diagram of the connections of the entire hardware system, and engines.

In the third section, "Working out of the program of work of a balancing trolley" describes the environment for the development of the program for work and presented a program with comments to it.

The fifth section, the last, "Technical and Economic importance", allows you to calculate your own costs for the project and the approximate price of the product.

Key words: electric drive, microcontroller, gyroscope - accelerometer, balancing cart, control system.

					<i>ЕП.МР.18.02.Р.ПЗ</i>	Арк.
						4
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Зміст

ВСТУП	6
РОЗДІЛ 1. ТЕХНОЛОГІЧНА ЧАСТИНА	7
1.1 Балансуючий візок	8
1.2 Принцип дії	8
1.3 Апаратна характеристика балансуєчого візка	9
1.4 Вимоги до системи.....	12
РОЗДІЛ 2. АВТОМАТИЗОВАНИЙ ЕЛЕКТРОПРИВОД.....	13
2.1 Інтерфейс І2С.....	14
2.2 Визначення орієнтації візку	16
2.3 Апаратна реалізація	20
2.3.1 Драйвер для керування двигунами.....	20
2.3.2 Підключення логіки балансуєчого візка	21
2.3.3 Підключення силової частини до логіки балансуєчого візка	22
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ РОБОТИ БАЛАНСУЄЧОГО ВІЗКА ..	24
3.1 Опис середовища розробки програми.....	25
3.2 Блок – схема програми роботи балансуєчого візка.....	26
3.3 Програма роботи балансуєчого візка.....	27
РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ.....	31
Вступ.....	32
4.1 Розрахунок витрат пов'язаних з виготовлення апаратури автоматики.....	32
4.2 Розрахунок виробничої собівартості.....	34
Висновок	34
ВИСНОВКИ.....	35
СПИСОК ЛІТЕРАТУРИ.....	36

					<i>ЕП.ДП.17.05.3.ПЗ</i>	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Тема магістерської роботи - Розробка автономного цифрового електропривода балансуєчого візка.

У сучасному світі область застосування робототехніки у різних сферах діяльності людини дуже широка та не перестає рости. Застосування роботів дозволяє значно знизити участь людини в важкій та небезпечній роботі. Наприклад, робота в захисних, хімічних, атомних сферах. Поступово роботи входять у звичайне життя людини.

Тема даної роботи є актуальною, так як розвиток робототехніки відбувається постійно. З моменту своєї появи півстоліття назад, роботи пройшли шлях від примітивних механізмів до складних ефективних пристроїв, багато в чому перевершивши за своїми можливостями людини.

У даній роботі обрана апаратна частина балансуєчого візка, розглянуті принципи роботи балансуєчої системи та розроблена програма роботи електропривода балансуєчого візка.

					<i>ЕП.МР.18.02.ВС.ПЗ</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

1.1 Балансуючий візок

Двоколісний робот (рис 1.1), котрого використовують для визначення куту нахилу корпусу робота та вироблення потрібного на приводи робота керуючої напруги (з метою забезпечити утримання рівноваги та виконання необхідних переміщень) ті чи інші гіроскопічні пристрої.

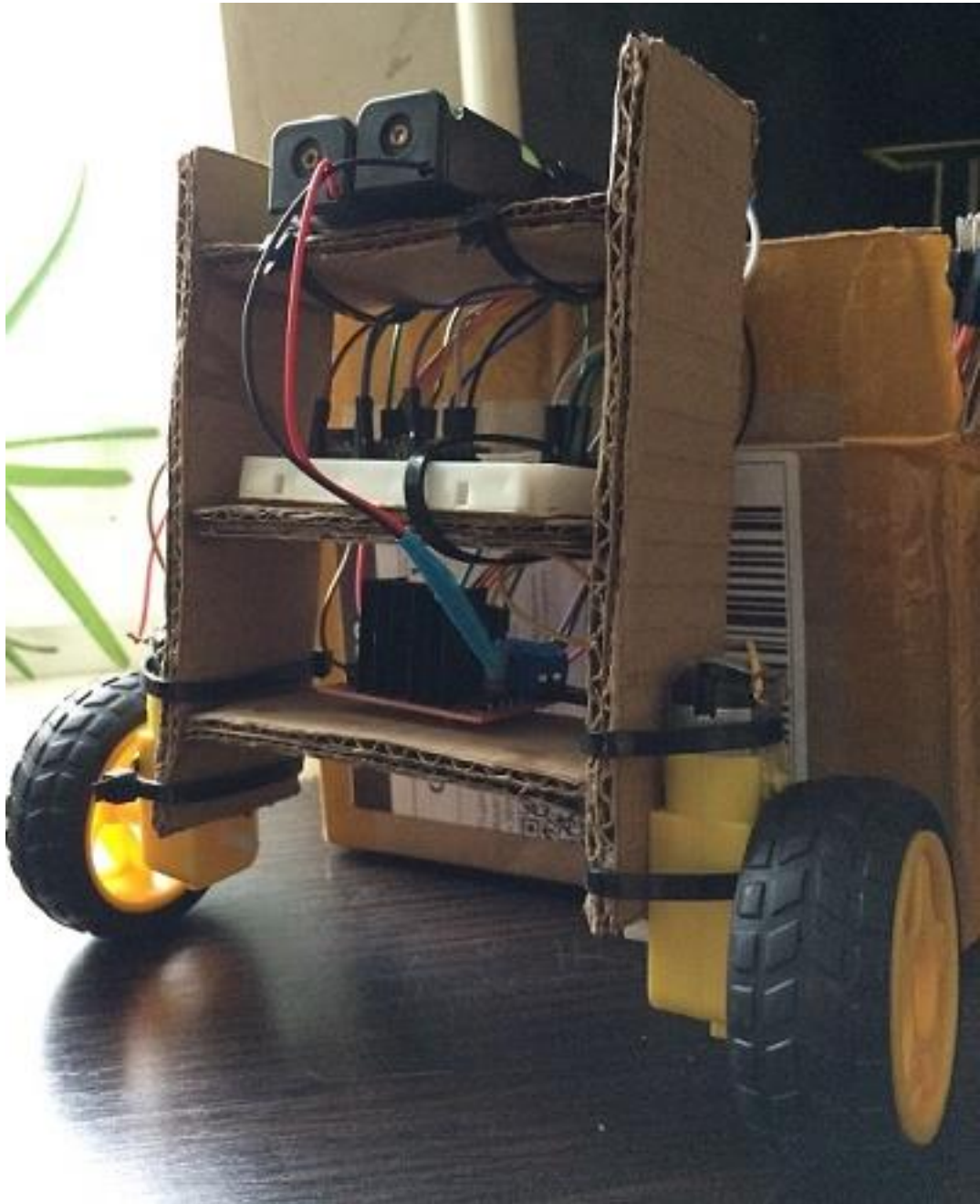


Рис. 1.1 Зовнішній вигляд балануючого візку

1.2 Принцип дії

Після увімкнення системи спочатку відбувається калібрування елемента зворотного зв'язку – гіроскоп – акселерометр. Надалі ця інформація надходить до мікроконтролера, який в свою чергу використовує її, щоб дізнатися про

					ЕП.МР.18.02.1.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

поточну орієнтацію візка. Після обробки цієї інформації, контролер надсилає керуючі імпульси до драйверу двигунів, які є коригувальним елементом, двигун та колесо.

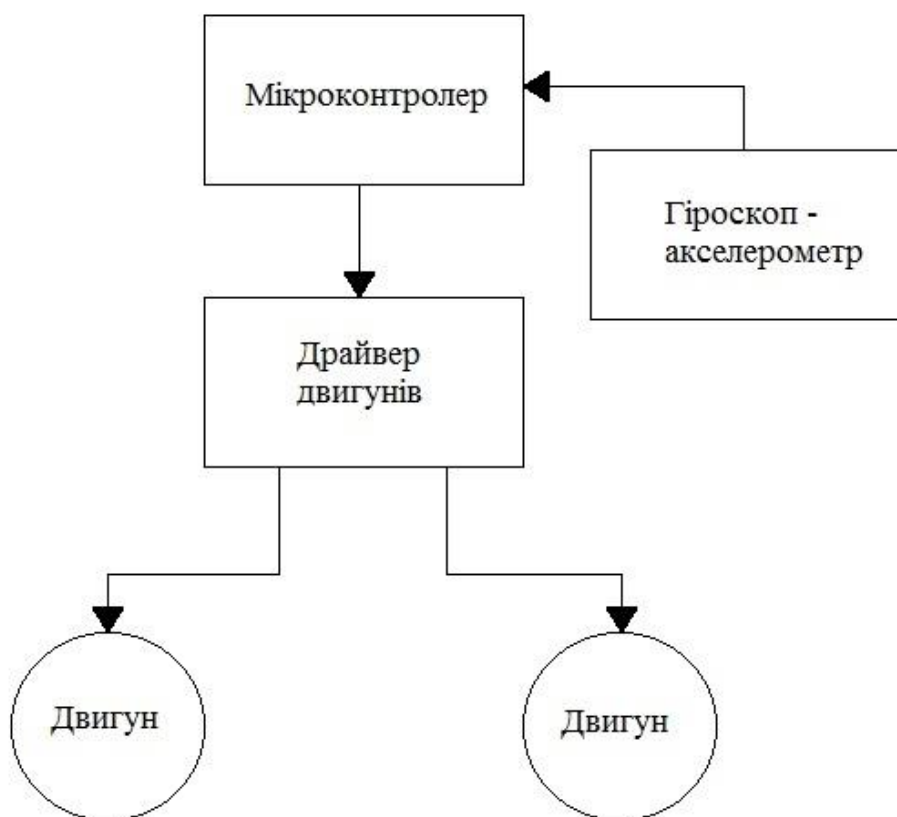


Рис 1.2 Функціональна схема системи керування

1.3 Апаратна характеристика балансуючого візка

1.3.1 Arduino Nano

Плата Arduino Nano зображена на рис. 1.3, має 14 цифрових входів/виходів, 6 входів можна використовувати як ШІМ, 6 аналогових входів, роз'єм USB, кварцовий генератор 16 МГц, а також кнопку перезавантаження.

					ЕП.МР.18.02.1.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

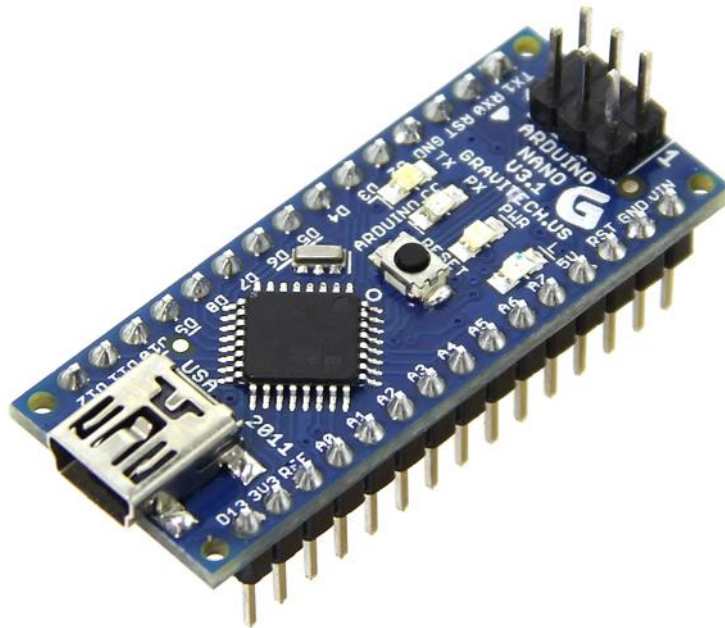


Рис. 1.3 Плата Arduino Nano

Для роботи плати або її програмування необхідно підключити до комп'ютера за допомогою кабелю USB, або зовнішнього живлення від блоку живлення або акумулятора. В якості основного обчислювача Arduino Nano використовує мікроконтролер ATmega328. Характеристики плати наведені у таблиці 1.1.

Таблиця 1.1

Характеристика пристрою Arduino Nano

Мікроконтролер	ATmega328
Робоча напруга (логічний рівень), В	5
Напруга живлення (рекомендована), В	5-12
Напруга живлення (граничне), В	6-20
Цифрові входи/виходи	14
Аналогові входи	8
Максимальний струм одного виходу, мА	40
Внутрішня пам'ять, кб	32
SRAM, кб	2
EEPROM, кб	1
Тактова частота, МГц	16

1.3.2 GY – 521

Модуль GY – 521 зображений на рис. 1.4, він містить акселерометр, гіроскоп та температурний сенсор. Характеристики модуля наведені у таблиці 1.2.

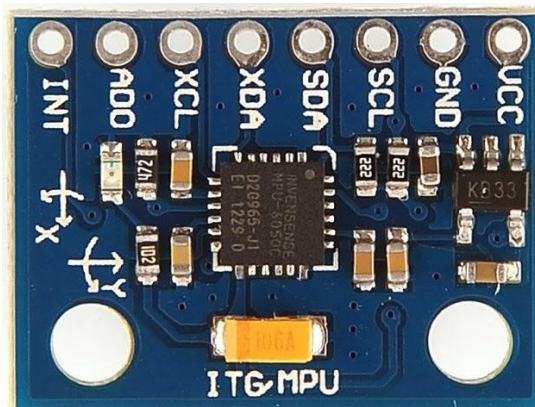


Рис. 1.4 Модуль GY – 521

Таблиця 1.2

Характеристика модулю гіроскоп – акселерометр GY – 521

Мікросхема	MPU – 6050
Напруга живлення, В	3,3 – 5
Діапазон вимірювань гіроскопа, °/с	+/-250, +/-500, +/-1000, +/-2000
Діапазон прискорень акселерометру	+/- 2g, +/- 4g, +/-8, +/-16g

MPU - 6050 - головний елемент модуля GY-531. Так само на платі модуля розташована необхідна обв'язка MPU - 6050, така як підтягуючі резистори інтерфейсу I2C і стабілізатор напруги на 3,3 В з малим падінням напруги (при живленні 3,3 В на виході стабілізатора буде рівно 3В) з фільтруючими конденсаторами. Так само на платі розпаяний SMD світлодіод з обмежуючим опором у якості індикатора напруги живлення. Характеристики мікросхеми MPU – 6050 наведені у таблиці 1.3.

Таблиця 1.3

Характеристика мікросхеми MPU – 6050

Напруга живлення, В	2,375 – 3,46
Споживаючий струм, мА	4
Інтерфейс передачі даних	I2C
Максимальна швидкість I2C, кГц	400
Вхід для інших датчиків	I2C
Внутрішній генератор, МГц	8

Характеристика силової частини наведена у таблиці 1.4.

Таблиця 1.4

Характеристика силової частини

Драйвер двигуна	L298N
Тип двигуна	Двигун постійного струму
Струм двигуна, А	2
Напруга живлення двигуна, В	7,4

1.4 Вимоги до системи

Система повинна відповідати наступним вимогам:

1. Утримання візку у вертикальному положенні.
2. Реагування на зовнішні дії (штовхання рукою).
3. Надання потрібної напруги для створення потрібного моменту двигуну для коригування положення.

					ЕП.МР.18.02.1.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

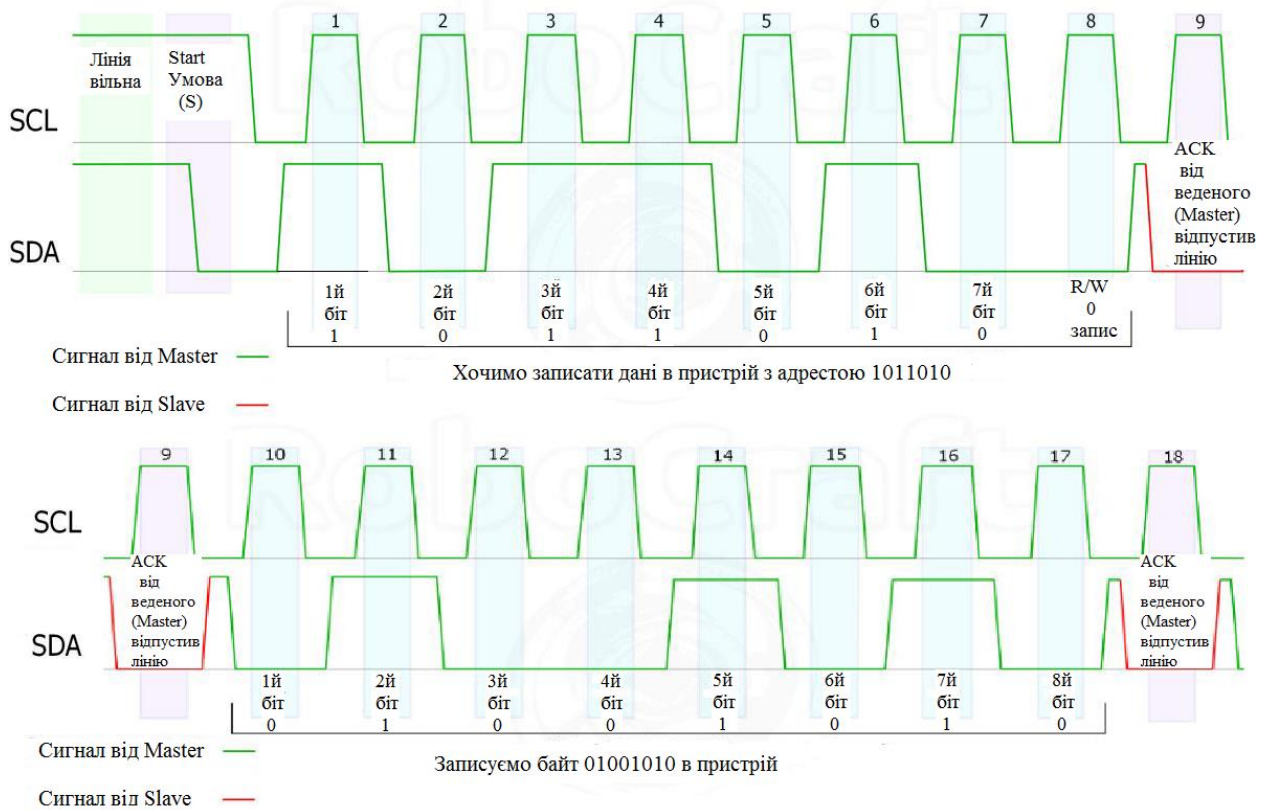


Рис. 2.2 «Спілкування» пристроїв Master та Slave по шині I2C

Цей процес можна описати наступним чином: старт, адрес(читання/запис), підтвердження, дані, стоп. Наглядно цей алгоритм наведено на рис. 2.3.

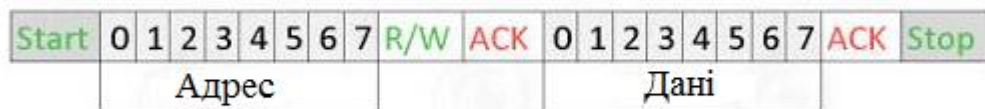


Рис. 2.3 Обмін даними по шині I2C

У випадку, коли дані приймає Master, для більшості пристроїв Slave після останнього байту даних Master передає команду nack, щоб завершити діалог. Цей процес наведений на рис. 2.4.



Рис. 2.4 Читання даних пристрою Master

2.2 Визначення орієнтації візку

Гіроскоп вимірює та виводить значення миттєвої кутової швидкості з дозволом, заданим у налаштуванні. Показання будуть приходити тільки при повороті, при наявності лінійної швидкості або відсутності впливу, гіроскоп буде видавати нульові значення. Для встановлення орієнтації об'єкта у просторі потрібно миттєве значення кутової швидкості помножити на проміжок часу між опитуваннями датчика гіроскопа. Далі кожне отримане значення повороту потрібно скласти з попереднім. Таким чином, при поверненні датчика у початкове положення значення відхилення дорівнюватиме нулю. При відхиленні від початкового положення, виконавши перераховані вище дії для трьох осей, можна отримати орієнтацію тіла в просторі. Так як гіроскоп не ідеальний і має деяку похибка, ця похибка інтегрується разом з показанням кутової швидкості та при тривалому використанні похибка визначення орієнтації тіла стає значною, тому гіроскоп використовують з акселерометром.

За допомогою акселерометра можна отримати прискорення трьох осей датчика. Перетворюючи дані за допомогою геометрії, можна також отримати орієнтацію об'єкта у просторі. Орієнтація об'єкта може спотворюватися при русі датчика в лінійних напрямках. Дані, одержані від акселерометра завжди досить точні, тому що не залежать ні від часу, ні від характеру впливу однак недолік у тому, що дані мають шум в деякому діапазоні даних, тобто з точністю до десятих часток градуса вимірювати кут неможливо. Виходячи з експериментальних даних, точність до цілих значень градуса досить стабільна, при відсутності лінійних переміщень.

Виходячи з описаних пристроїв та принципів, можна створити систему для визначення орієнтації візку у просторі. Для цього через інтерфейс I2C з'єднаємо плату Arduino Nano в режимі Master та мікросхему MPU-6050 у режимі Slave. Для отримання більшої точності слід сумісно використовувати гіроскоп та акселерометр модуля MPU-6050.

Зробити таке об'єднання дозволяє комплементарний фільтр, роботу якого визначається виразом:

$$a[i] = (1-K) \cdot (a[i-1] + gyr) + K \cdot acc,$$

де $a[i]$ – відфільтрований, результуючий кут нахилу, $a[i-1]$ – відфільтрований, результуючий кут нахилу на попередньому кроці, gyr та acc – значення куту нахилу, отримані від гіроскопа та акселерометра, K – коефіцієнт комплементарного фільтру.

Вибір коефіцієнту K залежить від величини дрейфу нулю гіроскопа, від швидкості накоплення помилок обчислення та від умов використаних пристроєм. Так, сильно велике значення K приведе до того, що на результат роботи фільтру буде сильно впливати зовнішні фактори. Дуже мале значення K може виявитися недостатнім, щоб ліквідувати дрейф нулю гіроскопа. Як

					ЕП.МР.18.02.2.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

правило, коефіцієнт комплементарного фільтру підбирається вручну для кожного інклінометра виходячи з вищевказаних умов.

Фільтр Калмана має можливість злагодити декілька показань датчиків з балансом шумоподавлення та пристосовності фільтру до змін всередині системи. Для цього необхідно знати стан системи, основане на декількох шумних датчиків. Маючи ці показання, система може знати про свій стан, виходячи із даних, отриманих на останньому кроці:

$$Xk = X'k + Kk(yk - h(X'k)),$$

де Xk - нова виправлена оцінка стану системи, $X'k$ - попередня оцінка стану системи, Kk - коефіцієнт посилення, yk - виміри датчику, h - функція виміру перетворення вектору стану системи.

Наступний крок це оцінка, заснована на новій виправленої оцінки з вимірів показань датчику:

$$X'k+1 = f(X'k', uk),$$

де uk - фактичний вимір з датчику, $X'k'$ - отримуємо з рівняння, $X'k$ - новий стан системи.

Фільтр Калману використовують для того, щоб позбавитися від дрейфу нуля гіроскопу. Для цього потрібно знати дрейф нуля на виході, котрий потрібно скорегувати сигналом на вході. Так само необхідно знати кут зміщення системи, тому сигнал з виходу датчику потрібно перевести з радіан у градуси:

$$\begin{aligned} \theta_{k+1} &= \theta_k + \omega_k \delta \\ \omega_{k+1} &= \omega_k + (y_k - bias_k) \delta \\ \omega_{k+1} &= \omega_k \\ \omega_{k+1} &= y_k - bias_k \\ bias_{k+1} &= bias_k \end{aligned}$$

де θ - кут, ω - кутова швидкість, δ - розмір вибірки, $bias$ - кутова швидкість зміщення гіроскопу, y - сигнал з виходу гіроскопу. Отримаємо сигнал з виходу у вигляді суми кутової швидкості зміщення гіроскопу та кутової швидкості обертання:

$$y_k^{gyro} = \omega_k + bias_k.$$

Видно, що є два шляхи для моделювання θ та ω . Проте, якщо обрати невірний набір рівнянь, то фільтр Калмана не буде працювати. Використання у коді рівнянь:

$$\begin{aligned} \omega_{k+1} &= \omega_k + (y_k - bias_k) \delta, \\ \omega_{k+1} &= y_k - bias_k \end{aligned}$$

допоможе уникнути нелінійності у системі, котра може привести до появи розходячої помилки. Необхідно взяти похідну від коефіцієнту посилення по відношенню до погрішності, щоб подати потрібне посилення та таким чином мінімізувати похибку. Отримаємо сигнал з виходу гіроскопу. Оновимо H використовуючи сигнал з виходу та сигнал з входу:

$$\begin{aligned} K &= PH(HPH^T + VRV^T)^{-1} \\ P &= (1 - KY)P \end{aligned}$$

					ЕП.МР.18.02.2.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

$$x = x + K(y - h(x))$$

Збережемо виправлені x для системи. Обновимо F використовуючи нові значення:

$$P = FPF^T + WQW^T$$

$$x = f(x, y)$$

тут K – коефіцієнт посилення Калмана, котрий має розмірність $m \times n$, де m – кількість змінних стану, а n – кількість датчиків. K показує число настроювальних станів системи, виходячи з різниці між вимірами. Q – дисперсія шуму у стані, матриця розміром $m \times m$ показує число помилок на кожний стан системи. R – дисперсія шуму при вимірах стану системи, матриця розміром $n \times n$. P – матриця, котра прирівнюється до матриці Q перед ітерацією. H – матриця для перетворення з старого стану до стану виміру за допомогою функції $h(x)$. Якщо рівняння є нелінійним, то береться його частна похідна відносно вимірюваних змінних. V – частна похідна від функції сигналу з виходу по відношенню до вихідного шуму. Зазвичай це одинична матриця розміром $n \times n$. I – це одинична матриця розміром $m \times m$. F – матриця перетворення у новий стан за рахунком старого стану системи. Це частна похідна функції P по відношенню до кожної змінної стану, якщо рівняння не є лінійним. W – частна похідна стану функції f по відношенню до стану шуму, як правило, це одинична матриця розміром $m \times m$. Важливо застосовувати ці рівняння у точній послідовності. Матриці H та F – це постійні матриці. Сформуємо матрицю станів, необхідних для роботи фільтру:

$$x_k = \begin{bmatrix} \theta_k \\ \omega_k \\ bias_k^{gyro} \end{bmatrix};$$

$$y_k = \begin{bmatrix} y_k^{gyro} \\ y_k^{accel} \end{bmatrix};$$

$$f(x_k, u_k) = \begin{bmatrix} \theta_k + \delta (y_k^{gyro} - bias_k^{gyro}) \\ y_k^{gyro} - bias_k^{gyro} \\ bias_k^{gyro} \end{bmatrix};$$

$$F = \begin{bmatrix} 1 & 0 & -\delta \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix};$$

$$h(x_k) = \begin{bmatrix} x_k + bias_k^{gyro} \\ \theta_k \end{bmatrix};$$

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix};$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$$R = \begin{bmatrix} sd_{gyro}^2 & 0 \\ 0 & sd_{accel}^2 \end{bmatrix};$$

$$Q = \begin{bmatrix} 0.28 \\ 0.2 \\ 0.1 \end{bmatrix} \times [0.28 \quad 0.2 \quad 0.1].$$

Реалізація такого алгоритму у програмі ускладнена великою кількістю обчислень, котрі можна не описувати вручну, а викликати функцією з стандартної бібліотеки «Kalman.h», що значно полегшує задачу.

Змн.	Арк.	№ докум.	Підпис	Дата

2.3 Апаратна реалізація

2.3.1 Драйвер для керування двигунами

Згідно технологічної частини для керування двома двигунами які корегують положення візку було обрано драйвер L298N на базі мікросхеми L298.

Інтегральна мікросхема L298 (рис. 2.5) - це потужний універсальний мостовий драйвер для управління двигунами постійного струму, кроковими двигунами, електромагнітними реле та електромагнітами (соленоїдами). У мікросхемі міститься два Н-моста, виконаних на потужних транзисторах, а також логіка сумісна з TTL.

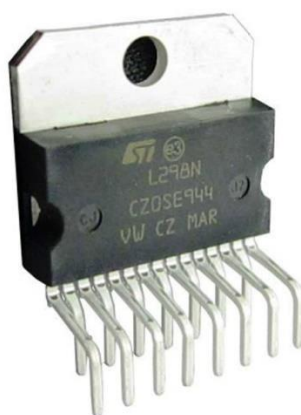


Рис. 2.5 – Зовнішній вигляд мікросхеми L298

Технічні характеристики мікросхеми:

- робоча напруга – до 46, В;
- максимальний постійний струм – 4, А (з радіатором);
- захист від перегріву;
- тримає логічний «0» доки напруга 1,5, В.

Змн.	Арк.	№ докум.	Підпис	Дата

ЕП.МР.18.02.2.ПЗ

Арк.

20

Нижче на рис. 2.6 наведена схема для драйвера двигунів на мікросхемі L298. Управління здійснюється за чотирма проводам (замість шести у L298) завдяки використанню додаткових інверторів в мікросхемі CD4011.

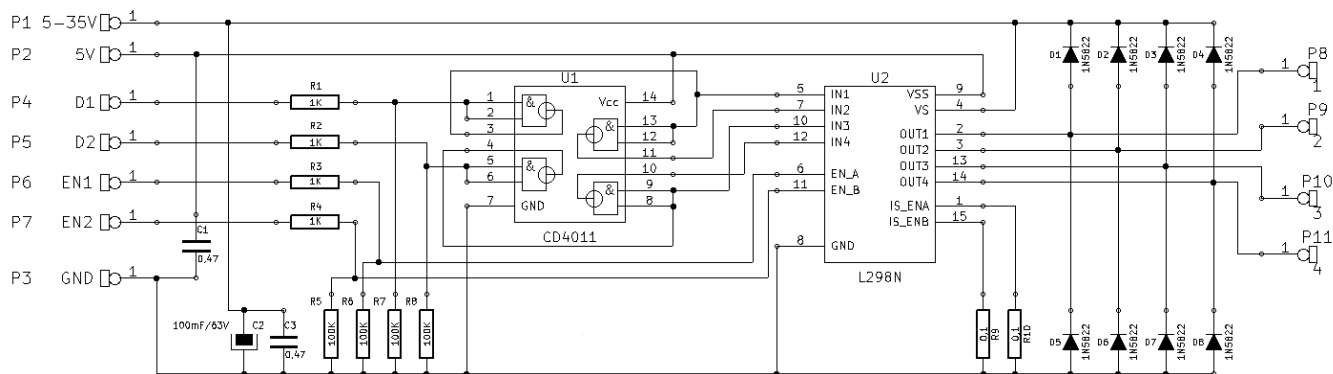


Рис. 2.6 Принципова схема драйверу L298N

Для живлення логіки обох мікросхем потрібна стабілізована напруга 5 В (P2). Для подачі напруги живлення на двигуни використовується окрема лінія P1. Виходи P4, P5 використовуються для установки полярності кожного з каналів, а виводи P6, P7 - дозволяють подачу живлення на каскади (ключі) внутрішнього Н-моста для кожного каналу.

2.3.2 Підключення логіки балансувального візка

Згідно з технологічної частини для логіки візка було обрано Arduino Nano на базі мікроконтролера ATmega328 та модуль GY – 521, схема підключення пристроїв наведена на рис. 2.8.

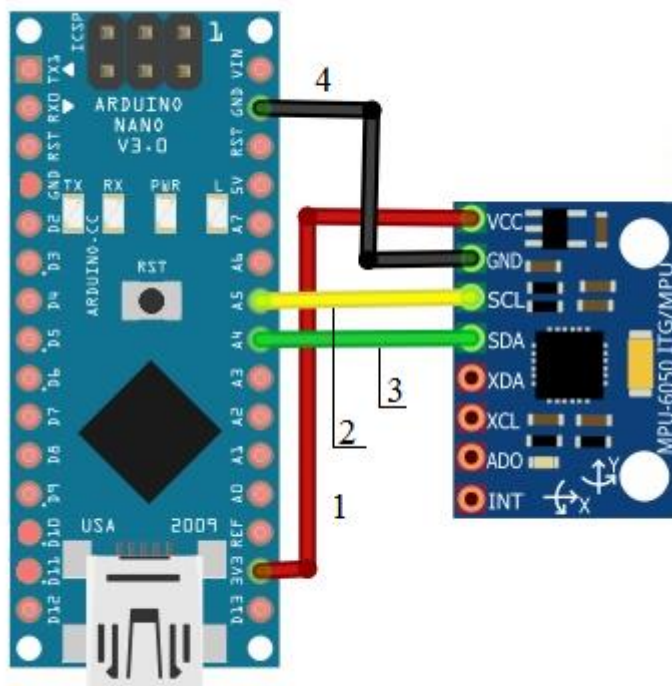


Рис. 2.8 Схема з'єднання плати Arduino Nano з модулем GY – 521

У таблиці 2.1 приведені входи/виходи з'єднання Arduino Nano з модулем GY – 521.

Таблиця 2.1

З'єднання Arduino Nano з модулем GY – 521

№	Arduino Nano	GY – 521
1	Вихід 3.3В	VCC
2	Аналоговий вхід A5	SCL
3	Аналоговий вхід A4	SDA
4	GND	GND

2.3.3 Підключення силової частини до логіки балансуєчого візка

Для керування двигунами було обрано вище описаний драйвер L298N та згідно технологічної частини було обрано два двигуна постійного струму. Схема підключення двигунів до драйверу наведена на рис. 2.9.

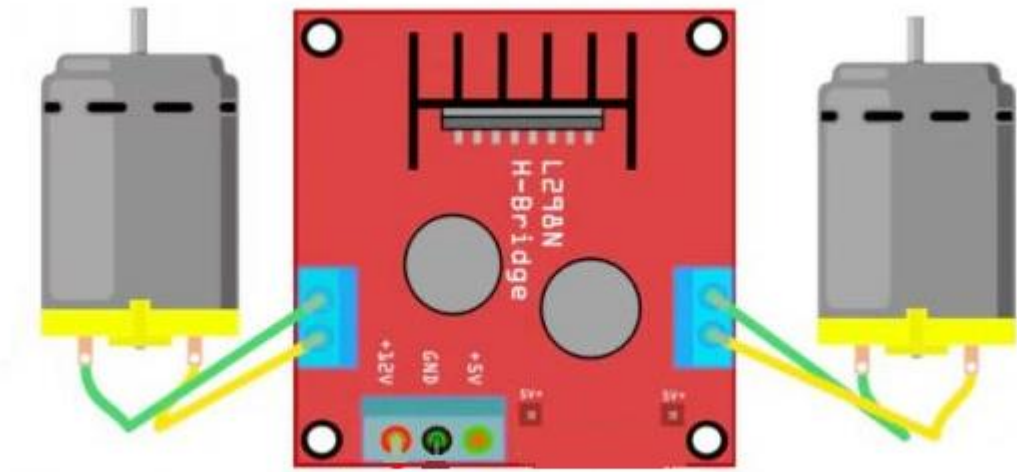


Рис. 2.9 Схема з'єднання двигунів до драйверу L298N

Далі підключимо логічну частину проекту с силовою її частиною, для отримання повної системи керування балансуєчого візка. Схема з'єднання наведена на рис. 2.10.

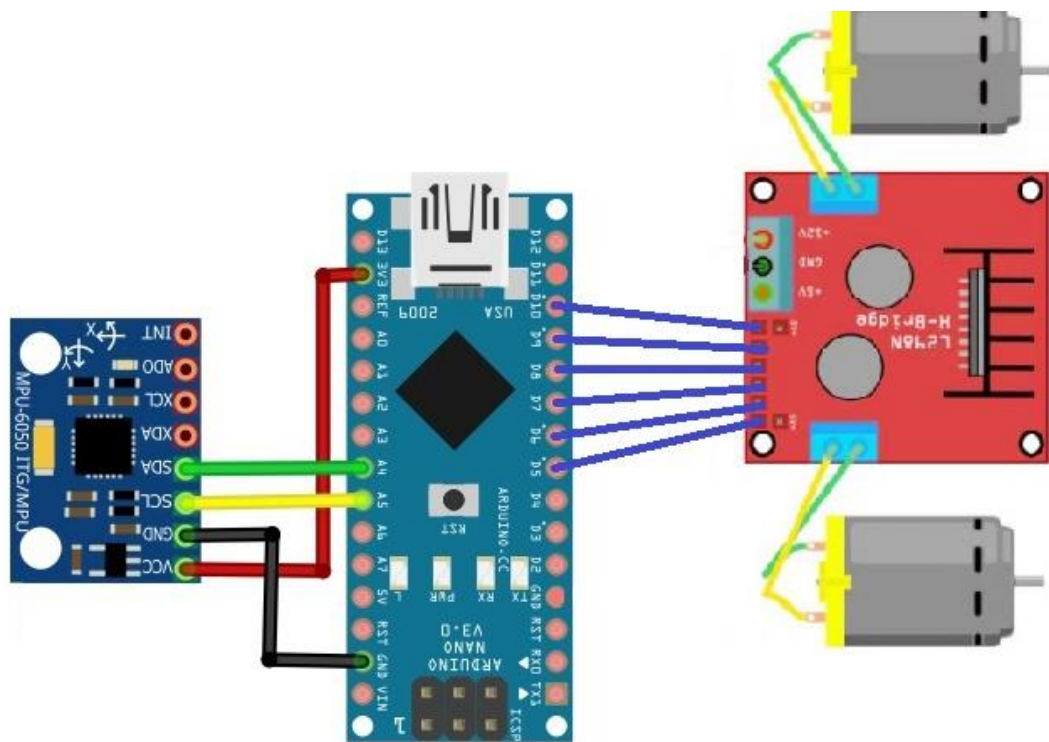


Рис. 2.10 Схема з'єднання драйверу L298N до Arduino Nano з модулем GY – 521

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМИ РОБОТИ БАЛАНСУЮЧОГО
ВІЗКА

					<i>EP.MP.18.02.3.P3</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Гунько М.В.</i>			<i>Розробка програми роботи балансуючого візка</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Перевір.</i>		<i>Яланський О.А.</i>					24	6
<i>Реценз.</i>						<i>НТУ «Дніпровська політехніка» 141М-17-4</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						

3.1 Опис середовища розробки програми

Оскільки для проекту я обрав мікроконтролер ATmega328 на платі Arduino Nano, то програмування ведеться цілком через власне програмне середовище (Arduino IDE), безкоштовно доступну на сайті Arduino. У цьому середовищі є текстовий редактор, менеджер проектів, препроцесор, компілятор і інструменти для завантаження програми в мікроконтролер. Середовище написано на мові Java на основі проекту Processing. Зовнішній вигляд середовища наведено на рис. 3.1.

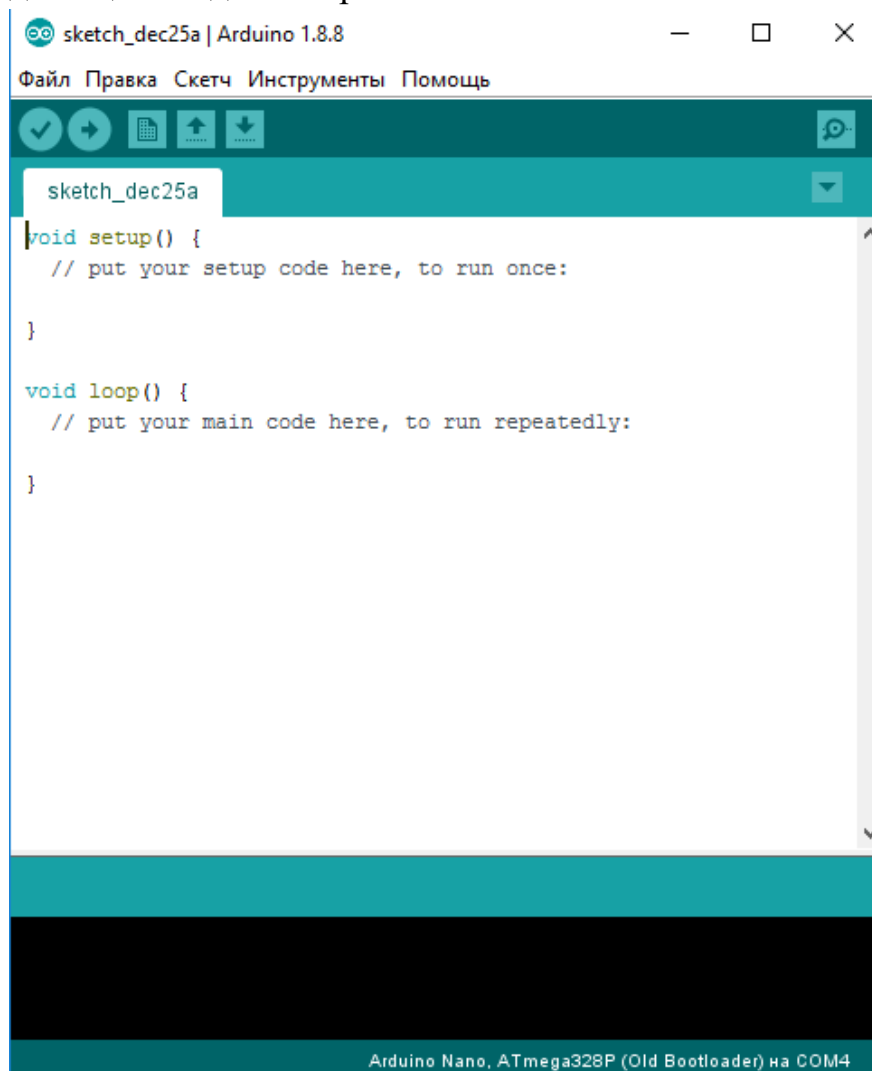


Рис. 3.1 Зовнішній вигляд Arduino IDE

Завантаження програми у мікроконтролер Arduino відбувається через попередньо запрограмований спеціальний завантажувач (всі мікроконтролери від Arduino продаються з цим завантажувачем), що значно полегшує роботу з мікроконтролером, бо не потребує програматора. Завантажувач створений на основі Atmel AVR Application Note AN109. Завантажувач може працювати через інтерфейси RS-232, USB або Ethernet залежно від складу периферії конкретної процесорної плати. У моєму випадку завантаження програми відбувається від USB Mini.

Отже щоб завантажити програму до мікроконтролеру, у Arduino IDE слід виконати наступні дії зображені на рис. 3.2.

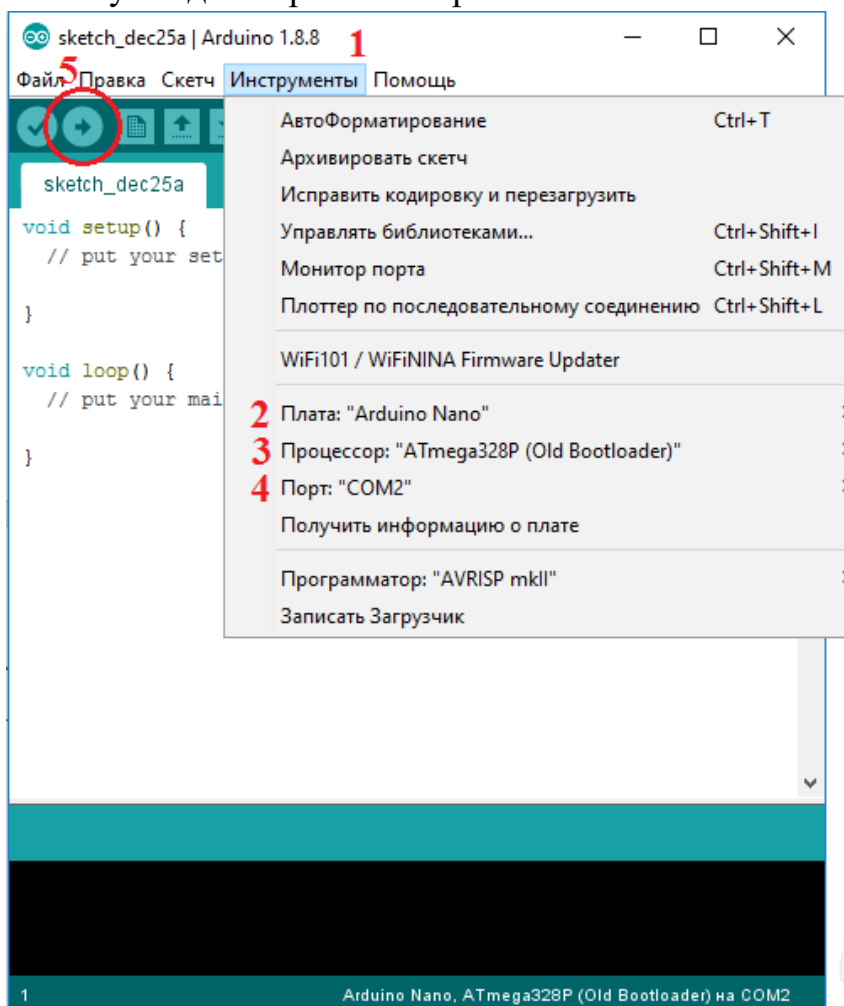


Рис. 3.2 Конфігурація середовища для завантаження програми:

1. Відкрити Інструменти;
2. Обрати серед плат свою плату;
3. Обрати на базі якого мікроконтролера зроблена плата;
4. Обрати порт до якого підключена плата;
5. Натиснути кнопку «Завантажити».

Проте крім рідного середовища розробки, можна встановити плагіни для роботи с Arduino на сторонні середовища, такі як Eclipse та Microsoft Visual Studio.

3.2 Блок – схема програми роботи балансуючого візка

Програма для мікроконтролера була написана на мові «C++», у середовищі Arduino 1.8.8. Структура блок – схеми наведена на рис. 3.3.



Рис. 3.3 Блок – схема роботи програми балансуючого візка

3.3 Програма роботи балансуючого візка

Обов'язковою для написання програми є використання таких функцій як: «setup, loop». Функція «setup ()» викликається тоді, коли стартує програма. Використовується для визначення режимів роботи виходів, запуску використовуваних бібліотек. Функція «setup» запускається тільки один раз, після кожної подачі живлення або скидання плати Arduino Nano. Функція «loop ()» використовується після функції «setup ()», яка ініціалізує і встановлює початкові значення, функція «loop» крутиться в циклі, дозволяючи

програмі здійснювати обчислення і реагувати на них. Використовується для активного управління Arduino.

```
//Підключення бібліотек
#include <Wire.h> // Для обміну даних між пристроями
#include "Kalman.h" // Для розрахунку стану системи
#include "PID_v1.h" // Для завдання на відпрацювання
заданного куту

Kalman kalmanY;
// Встановлення адреси модуля MPU6050
uint8_t IMUAddress = 0x68;

// Оголошення змінних
int16_t accX;
int16_t accY;
int16_t accZ;

int16_t gyroY;

double accYangle;
double gyroYangle = 180;
double compAngleY = 180;
double kalAngleY;

double in, out, setpoint;
PID pid(&in, &out, &setpoint, 30, 410 , 0.8, DIRECT);

uint32_t timer;
// Ініціюємо підключення до шини I2C в якості ведучого
(master) пристрою
void setup() {
  Serial.begin(115200);
  Wire.begin();
  i2cWrite(0x6B,0x00); // Disable sleep mode
  if(i2cRead(0x75,1)[0] != 0x68) { // Read "WHO_AM_I"
register
    Serial.print(F("MPU-6050 with address 0x"));
    Serial.print(IMUAddress,HEX);
    Serial.println(F(" is not connected"));
    while(1);
  }
  kalmanY.setAngle(180);

  pid.SetMode(AUTOMATIC);
  pid.SetOutputLimits(-255,255);
  pid.SetSampleTime(10);
```

```

    setpoint = 181.8;

    timer = micros();
}

double ang;
int pwm;

void loop() {
    uint8_t* data = i2cRead(0x3B,14);
    accX = ((data[0] << 8) | data[1]);
    accY = ((data[2] << 8) | data[3]);
    accZ = ((data[4] << 8) | data[5]);
    gyroY = ((data[10] << 8) | data[11]);

    // Для отримання положення тіла за допомогою
    акселерометра. Цей кут буде вимірюватися в радіанах.
    accYangle = (atan2(accX,accZ)+PI)*RAD_TO_DEG;

    double gyroYrate = -((double)gyroY/131.0);
    gyroYangle += gyroYrate*((double)(micros()-
timer)/1000000);
    // Розрахунок стану системи за допомогою бібліотеки
    "Kalman.h"
    kalAngleY = kalmanY.getAngle(accYangle, gyroYrate,
(double)(micros()-timer)/1000000);
    timer = micros();
    // Завдання керуючих імпульсів до двигунів
    in = kalAngleY;
    pid.Compute();
    if (out < 0) {
        pwm = -1 * out;
        analogWrite(11, 0);
        analogWrite(10, pwm);
    } else {
        pwm = out;
        analogWrite(10, 0);
        analogWrite(11, pwm);
    }

    delay(1); // Затримка 1мкс після перетворення показань
акселерометра, так як він працює з частотою 1МГц
}

void i2cWrite(uint8_t registerAddress, uint8_t data){
    Wire.beginTransmission(IMUAddress);

```

					ЕП.МР.18.02.3.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Wire.write(registerAddress);
Wire.write(data);
Wire.endTransmission();
}

uint8_t* i2cRead(uint8_t registerAddress, uint8_t nbytes)
{
    uint8_t data[nbytes];
    Wire.beginTransmission(IMUAddress);
    Wire.write(registerAddress);
    Wire.endTransmission(false);
    Wire.requestFrom(IMUAddress, nbytes);
    for(uint8_t i = 0; i < nbytes; i++)
        data[i] = Wire.read();
    return data;
}

```

					ЕП.МР.18.02.3.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

РОЗДІЛ 4
ТЕХНІКО - ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

					<i>ЕП.МР.18.02.4.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Гулько М.В.</i>			<i>Техніко-економічне обґрунтування</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Перевір.</i>		<i>Тимошенко</i>					31	3
<i>Реценз.</i>						<i>НТУ «Дніпровська політехніка» 141М-17-4</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						
<i>Затверд.</i>								

Вступ

У наш час технологічний прогрес настільки швидко зростає, що роботи замінюють людину у сферах типової праці. Тому ринок праці вимагає все більше кваліфікованих випускників спеціальності 141 «Електромеханічні системи та електропривод», для спостереження цих самих роботів, вміння їх полагодити, налаштувати та запрограмувати. Навчання за даним напрямком включає в себе вивчення електродвигунів та систем керування, автоматизації.

Метою проекту є створення автономного електроприводу балансуєчого візка (або робот – балансир) на базі мікроконтролера, який надалі буде використовуватися як демонстраційний стенд для дисципліни – автоматизація загальнопромислових установок і технологічних комплексів «АЗПУ і ТК».

Лабораторний стенд має такий ряд переваг, як:

- легкість зборки та розбирання (що полегшує розуміння, як це працює на апаратній частині);
- сучасна база елементів;
- можливість вдосконалення проекту;
- можливість перепрограмування.

4.1 Розрахунок витрат пов'язаних з виготовленням апаратури автоматики

Оскільки на стадії проектування та розробки системи керування питання про місце впровадження апаратури в виробництво не вирішено, для розрахунку можливо використати укрупнений метод розрахунку. Цей засіб заснований на визначенні заводської собівартості за питомою вагою в ній певних елементів витрат, оскільки для проєктованого об'єкту і аналогу структура окремих елементів витрат в певних межах зберігається.

Виробничу собівартість можна визначити за формулою:

$Спр = Зму + Зу * (3 + (\alpha + \beta)/100)$, грн, де, $Зму$ - вартість основних матеріалів і комплектуючих виробів в проєктованій апаратурі, грн;

$Зу$ - основна заробітна плата робочих, зайнятих виготовленням запроєктованої апаратури, грн;

α - цехові видатки, %

β - загальнозаводські видатки, %

Вартість основних матеріалів визначається на підставі норм їх видатку і оптових цін. Із загальної вартості матеріалів необхідно вирахувати вартість зворотних відходів і додати вартість транспортно-заготівельних видатків.

Для розрахунку вартості основних матеріалів і комплектуючих треба підрахувати основні витрати. Результати зводяться до таблиць 4.1-4.2.

					ЕП.МР.18.02.4.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 Вартість основних матеріалів

№	Найменування матеріалу	Одиниця виміру	Вага; об'єм; розмір.	Норма витрат на один виріб	Оптова ціна за одиницю	Вартість на один виріб	обґрунтування витрат
1	Олово LC-60	грам	100	≈50-60 грам	120, грн	120, грн	RKS
2	Всього					120, грн	

Таблиця 4.2 Вартість комплектуючих виробів апаратури

№	Найменування, тип, номінал	Кількість, од.	Сума, грн	Обґрунтування витрат та ціни.
1	Мікроконтролер Arduino Nano	1	54,06	Aliexpress
2	Модуль GY-521	1	21,62	Aliexpress
3	Драйвер L298N	1	39,09	Aliexpress
4	Дроти	120	60,99	Aliexpress
5	Акумулятор Li-ion MH12210	2	151,10	Aliexpress
6	Двигун с колесом та редуктором	2	89,83	Aliexpress
7	Коробка під акумулятор	2	40	Aliexpress
8	Макетна плата	1	27,45	Aliexpress
9	Всього		484,14	

Згідно до табл. 4.1 – 4.2 вартість елементів на реалізацію повного проекту складатиме: $120 + 484,14 = 604,14$, грн.

До витрат на комплектуючі вироби потрібно додати транспортно-заготівельні видатки. Оскільки всі елементи можливо купити в одному онлайн магазині – мінімальні видатки складатимуть лише доставку поштою – 35, грн.

$$Зму = 604,14 + 35 = 639,14, \text{ грн}$$

Основна заробітна плата робочих Z_u визначається за укрупненим методом виходячи з ознак зіставляємості:

$Z_u = Z_{му} * (U_z / U_m)$, де U_z , U_m – питомі ваги основної заробітної плати і вартості матеріалів і комплектуючих в виробничій собівартості аналогу відповідно, %. [n]

В таблиці немає даних, які повністю підходять під проект. Приймаємо найбільш приблизне – Прилади радіоелектронні. $U_z = 13.5 \%$, $U_m = 54 \%$.

$$Z_u = 639,14 * (0.135 / 0.54) = 159.79, \text{ грн}$$

Через відсутність фактичних значень цехових та загальнозаводських видатків, середні значення можливо отримати з табл.2 [n].

					ЕП.МР.18.02.4.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

В таблиці немає даних, які повністю підходять під проект. Приймаємо найбільш приблизне – Електронні вимірювальні прилади. $\alpha = 127 \%$, $\beta = 73 \%$, $\gamma = 7.2 \%$.

4.2 Розрахунок виробничої собівартості

$$\text{Спр} = \text{Зму} + \text{Зу} * (1 + (\alpha + \beta)/100) = 639,14 + 159,79 * (1 + (127 + 73)/100) = 1118,5, \text{ грн.}$$

Повна собівартість запроєктованої апаратури, визначається як:

$$\text{Сп} = \text{Спр} * (1 + \gamma/100) = 1118,51 * (1 + 7.2/100) = 1199,04, \text{ грн.}$$

Оптова відпускна ціна запроєктованої апаратури визначається за формулою:

$\text{Цл} = \text{Сп} + \text{П} + \text{ПДВ}$, грн, де ПДВ – податок на додану вартість (20 % від $\text{Сп} + \text{П}$), грн;

П – прибуток, що включається в ціну апаратур:

$\text{П} = \text{Сп} * (\text{R}/100)$, грн, де R – очікувана рентабельність виробництва апаратури, % (в межах 30%).

$$\text{П} = 1199,04 * (20/100) = 239,8, \text{ грн.}$$

$$\text{ПДВ} = (1199,04 + 239,8) * 0.2 = 287,77, \text{ грн.}$$

$$\text{Цл} = 1199,04 + 239,8 + 287,77 = 1727, \text{ грн}$$

Висновок

При розрахунку економічної складової магістерської роботи було встановлено, що витрати на створення лабораторного стенду становлять 639 гривень, а орієнтовна ціна виробу - 1727 гривень. Лабораторний стенд буде використовуватись у навчальному процесі при проведенні лабораторних занять з дисципліни «Автоматизація загальнопромислових установок і технологічних комплексів», при підготовки магістрів за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка».

					ЕП.МР.18.02.4.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У роботі були описані методи отримання орієнтації тіла в просторі, з використанням гіроскоп - акселерометра, а так само управління двигунами постійного струму з використанням цих даних. Було встановлено що розробити прилад для визначення орієнтації тіла в просторі може бути використано Arduino Nano, як пристрій, що виконує необхідні розрахунки і управління приводами, і модуль MPU-6050 в якості недорогого і зручного у використанні датчика кутового положення.

Результатом роботи став макет ручного балансуєчого візка, що дозволяє здійснювати стабілізацію його у просторі.

					ЕП.МР.18.02.В.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ЛІТЕРАТУРИ

1. Улли Соммер «Программирование микроконтроллерных плат Arduino/Freeduino», Издательство БХВ–Петербург 2012г.
2. Стенли Б. Липпман, Жози Лажойе «Язык программирования C++. Вводный курс.», Издательство «Вильямс» 2007г
3. Статті о Arduino Nano. URL: <https://arduinomaster.ru/platy-arduino/plata-arduino-nano/>
4. L298N Datasheet
5. MPU – 6050 Datasheet
6. Статті о гіроскоп – акселерометр MPU – 6050. URL: <http://robotclass.ru/articles/complementary-filter/>
7. Методические указания к выполнению экономической части дипломной работы для студентов направления подготовки 6.050702 «Электромеханика» / Составители: Л.В. Тимошенко, И.В. Шереметьева. – Днепрпетровск: НГУ, 2015. – 15 с.
8. Филлипс Ч., Харбор Р. Системы управления с обратной связью. – М.. Лаборатория Базовых Знаний, 2001 – 616 с.

					<i>ЕП.МР.18.02.СЛ.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36