

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб для вимірювання миттєвої швидкості руху з
лічильником відстані на базі мікроконтролера Atmega 328P
Назва теми

КВРКІ. 180102.18.01.02 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ-18-1 Бінь Я. В. Біньковський
Підпис Ініціали, прізвище

Керівник [Підпис] А. О. Нічепорук
Підпис, дата Ініціали, прізвище

Нормоконтролер [Підпис] С.М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної інженерії та інформаційних систем [Підпис] Т.О. Говорушенко
Підпис Ініціали, прізвище

« 16 » 06 2022 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорушенко

.. .. 202_ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Біньковському Ярославу Васильовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P
- Керівник проекту (роботи) Нічепорук А.О., к.т.н. доц

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Огляд відомих програмно-технічних засобів та рішень

Аналіз та вибір елементної бази для лічильника миттєвої швидкості

руху

Побудова пристрою для вимірювання миттєвої швидкості

руху

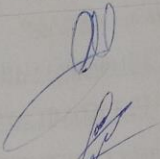

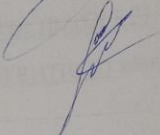

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Схема апаратних з'єднань системи

Блок-схеми роботи пристрою

Реалізація пристрою

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		№	Ф о р м а т
		завдання видав	завдання при		
Нормоконтроль	Лисенко С.М., професор кафедри КПС				
Антиплагіат	Нічепорук А.О., доцент кафедри КПС			1	


7. Дата видачі завдання « 11 » 01 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Прим.
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконано
3	Робота над розділом 1 – Огляд відомих програмно-технічних засобів та рішень	01.03.2022	виконано
4	Робота над розділом 2 – Аналіз та вибір елементної бази для лічильника миттєвої швидкості руху	01.04.2022	виконано
5	Робота над розділом 3 – Побудова пристрою для вимірювання миттєвої швидкості руху	30.04.2022	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2022	виконано
7	Попередній захист ВКР	24.05.2022	виконано
8	Захист ВКР на засіданні ЕК	Червень 2022 року	виконано

Студент

Керівник проекту (роботи)


Підпис

Я. В. Біньковський
Ініціали, прізвище

А. О. Нічепорук
Ініціали, прізвище

швидкості руху.

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P».

Автор роботи: Бінковський Ярослав Васильович.

Керівник роботи: Нічепорук Андрій Олександрович.

Пояснювальна записка: 57 с., 16 рис., 6 табл., 3 дод., 29 джерел.

Графічна частина: 7 презентаційних слайдів.

ЛІЧИЛЬНИК ВІДСТАВНІ, ВИМІРЮВАННЯ МИТТЄВОЇ ШВИДКОСТІ РУХУ, ARDUINO IDE, МІКРОКОНТРОЛЕР АТМЕГА 328P.

Метою роботи є розробка програмно-технічного пристрою для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P.

Завдання розробки бюджетного та компактного програмно-технічного пристрою для вимірювання миттєвої швидкості руху з добовим та загальним лічильником відстані з можливістю встановлення у корпус з під стандартного спідометра з незначним втручанням в конструкцію корпусу спідометра. Потрібно отримати імпульси з датчика через оптопару, вирахувати за формулою і вивести на OLED дисплей з дублюванням швидкості на сервопривід.

Бірко





Підпис студента

16.06.2022

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	3
ВСТУП.....	4
1 ОГЛЯД ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ ТА РІШЕНЬ.....	5
1.1 Огляд предметної області	5
1.2 Типи пристроїв для визначення швидкості руху та принцип функціонування	8
1.3 Аналіз готових приборів, призначених для вимірювання швидкості	12
1.4 Висновки	18
2 АНАЛІЗ ТА ВИБІР ЕЛЕМЕНТНОЇ БАЗИ ДЛЯ ЛІЧИЛЬНИКА МИТТЄВОЇ ШВИДКОСТІ РУХУ.....	20
2.1 Аналіз функціонування пристрою для вимірювання швидкості	20
2.2 Аналіз та вибір мікроконтролера.....	26
2.3 Обґрунтування вибору програмного забезпечення.....	39
2.3 Висновки	49
3 ПОБУДОВА ПРИСТРОЮ ДЛЯ ВИМІРЮВАННЯ МИТТЄВОЇ ШВИДКОСТІ РУХУ	50
3.1 Проектування пристрою для вимірювання миттєвої швидкості руху.....	50
3.2 Програмування пристрою визначення швидкості руху	52
3.3 Блок-схеми роботи пристрою.....	53
3.4 Висновки	59
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	62
ДОДАТОК А.....	65
ДОДАТОК Б.....	65
ДОДАТОК В	66
ДОДАТОК Г.....	67

КвРКІ. 180102.18.01.02 ПЗ					
Зм.	Арк.	Медокум.	Підпис	Дата	
Виконав	Биньковський Я.В.				
Перевір.	Нічепорук А.О.				
Н.контр.	Лисенко С.М.				
Затвер.	Говорущенко Т.О.				
Програмно-технічний засіб для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P			Літера	Арк.ш	Арк.шів
			у		
			ХНУ КІ-18-1		

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЦАП – Цифро-аналоговий перетворювач

АЦП – Аналого-цифровий перетворювач

MCU – Multipoint Control Unit

RAM - Оперативна пам'ять

ПО – програмне забезпечення

EEPROM – енергонезалежна пам'ять

РК дисплей – Рідкокристалічний дисплей

OLED – тип дисплея Organic Light Emitting Diod

I2C або ІІС – двохпровідникова шина даних

SDA (Serial Data) – лінія для передачі даних

SCL (Serial Clock) – передає сигнал з заданою частотою для синхронізації

Спідометр – Лічильник миттєвої швидкості руху

ДАІ – Дорожня автомобільна інспекція

					КвРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Люди постійно прагнуть покращити якість свого життя та оптимізувати свій робочий час, тому щодня з'являються нові технології, які покликані полегшити життя людей. Такі процеси автоматизації не можливі без залучення мікроконтролерів. З огляду на те, що нинішні зразки мікроконтролерів мають досить високі обчислювальні потужності, що дозволяють лише на одній мікросхемі реалізувати повнофункціональній пристрій невеликого розміру, разом із низьким енергоспоживанням – вартість безпосередньо готових пристроїв стає все нижчою.

Тому мікроконтролери можна зустріти всюди в електронних блоках абсолютно різних пристроїв: на материнських платах комп'ютерів, в контролерах DVD-приводів, жорстких і твердотільних накопичувачів, в калькуляторах, на платах управління пральних машин, мікрохвильових печей, телефонів, пілососів, посудомийних машин, всередині домашніх роботів, програмованих реле та ПЛК, у модулях управління верстатами і т.д.

На даний момент є багато морально застарілих автомобілів, які все ще активно використовуються, у яких стандартний прилад виміру швидкості вийшов з ладу, а нових приладів немає в наявності, або покупка останніх є нерентабельною.

Тому виникає завдання розробки бюджетного та компактного програмно-технічного пристрою для вимірювання миттєвої швидкості руху з добовим та загальним лічильником відстані з можливістю встановлення у корпус з під стандартного спідометра з незначним втручанням в конструкцію корпусу спідометра. Потрібно отримати імпульси з датчика через оптопару, вирахувати за формулою і вивести на OLED дисплей з дублюванням швидкості на сервопривід.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

1 ОГЛЯД ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ ТА РІШЕНЬ

1.1 Огляд предметної області

Поява транспортних засобів стала поворотним моментом в історії людства. Самохідні вагони різних схем і моделей були розроблені багатьма винахідниками ще в 16 - 18 століттях. Французький офіцер Ніколя-Жозеф Кюньо має велику заслугу в побудові першого повністю самохідного автомобіля близько 1769 року; створив паровий триколісний велосипед.

Він також виготовив два парові трактори (пожежні машини) для французької армії, один з яких знаходиться у Французькій національній консерваторії мистецтв і ремесел. У 1801 році Річард Тревітік побудував і продемонстрував свій паровий вагон, який, на думку багатьох, був першим представленням парового вагона. Цей локомотив тривалий час не витримував достатнього тиску пари і мало використовувався [1].

Швидкий розвиток автомобілів почався з появою швидких і економічних двигунів внутрішнього згорання. У 1879 році Бенц отримав патент на свій перший двигун, розроблений у 1878 році. Його перший Motorwagen був виготовлений у 1885 році в Мангеймі, Німеччина. Він отримав патент на свій винахід 29 січня 1886 року (під егідою своєї великої компанії Benz & Cie, заснованої в 1883 році). Benz почав продавати автомобіль 3 липня 1886 року, а між 1888 і 1893 роками було продано близько 25 одиниць Benz.

Їх також привезли з чотирма двигунами їхньої конструкції. Француз Еміль Роже, який вже виробляє двигуни Benz за ліцензією, додав автомобіль Benz до свого асортименту автомобілів. Оскільки Франція була більш відкритою для ранніх автомобілів, Роджер виробляв і продавав у Франції більше, ніж Benz продав у Німеччині.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

У серпні 1888 року Берта Бенц, дружина Карела Бенца, здійснила свою першу поїздку на автомобілі, щоб довести технічну придатність винаходу свого чоловіка.

З приїздом автомобілів необхідно було контролювати швидкість автомобіля. Перший спідометр винайшов хорват Йозеф Белушич у 1888 р. Цей винахід був запатентований в Австро-Угорщині як «спідометр». Використання в машинах набули на початку 20 ст. Одна з перших моделей спідометрів була виготовлена Нікола Тесла і запатентована в 1916 році (патент № 1209359, виданий Патентним відомством США).

У серпні 1888 року Берта Бенц, дружина Карела Бенца, здійснила свою першу поїздку на автомобілі, щоб довести технічну придатність винаходу свого чоловіка. З приїздом автомобілів необхідно було контролювати швидкість автомобіля. Перший спідометр винайшов хорват Йозеф Белушич у 1888 р. Цей винахід був запатентований в Австро-Угорщині як «спідометр». Використання в машинах набули на початку 20 ст [2].

Одна з перших моделей спідометрів була виготовлена Нікола Тесла і запатентована в 1916 році (патент № 1209359, виданий Патентним відомством США).

Спідометр – прилад для вимірювання поточної швидкості транспортного засобу (далі спідометр).

За способом вимірювання спідометри бувають:

1. Відцентрові - керуючий важіль, утриманий пружиною, обертається разом з шпинделем і відцентрова сила викидається вбік, тому рух прямо пропорційний швидкості.
2. Хронометричний – поєднання годинникового механізму з одометром.
3. Індукційний – система постійних магнітів, які обертаються разом з шпинделем, створюючи в магнітному полі вихрові струми в мідному або алюмінієвому диску. Таким чином, диск втягується в рух колеса, але його

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

обертання сповільнюється обмежувальною пружиною. Привід підключений до стрілки швидкості.

4. Електромагнітна – швидкість визначається значенням ЕРС, що створюється тахогенератором, підключеним до шпинделя.

5. Вібраційний – механічний резонанс коливань підшипників або рами верстата викликає коливання ступінчастих язиків з частотою, що відповідає швидкості машини.

6. З супутниковою системою позиціонування – швидкість визначається супутниковою системою позиціонування GPS.

7. Електронний - оптичний, магнітний або механічний датчик генерує імпульс струму з кожною швидкістю шпинделя. Імпульси обробляються в електронному вигляді, а значення швидкості відображаються на екрані.

8. Залежно від типу показчика спідометри бувають аналогові та цифрові. Аналогові спідометри поділяються на:

9. Стрічковий – швидкість вказується пропускнуою здатністю, що проходить через секції у фіксованій шкалі. Він використовувався на багатьох американських і деяких японських, європейських і радянських моделях до початку 1975 року.

10. Барабанний – на барабан наносяться розділи, і коли він обертається, він з'являється у вікні, що показує поточну швидкість. Він використовується на багатьох автомобілях до 1940-х років, на деяких американських автомобілях 1960-х років, а також на відносно сучасних моделях "Citroen".

11. Стрілковий — найбільш поширений варіант спідометра, швидкість вказує стрілка, яка обертається навколо осі.

					КвРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1.2 Типи пристроїв для визначення швидкості руху та принцип функціонування

Сьогодні існують автомобілі у яких колеса обертаються з певною швидкістю, і водій хоче знати, за допомогою простого покажчика та циферблата, яка ця швидкість.

Отже, потрібно з'єднати обертові колеса з покажчиком якимось розумним способом. Навіть це досить складно: колеса мчать навколо, але вказівник на деякій відстані просто гортає вперед-назад.

Для того щоб перетворити безперервний обертальний рух у переривчастий, мерехтливий рух вказівника слід використовувати електромагнетизм [3].

Приблизно до 1960-х років практично всі спідометри використовували комбінацію механічної потужності та електромагнетизму. На рисунку 1.1 маленьке колесо (червоне), яке приводиться в рух диском (помаранчевий), прикріпленим до одного з передніх коліс автомобіля (сірий), обертає трос (зелений), який звивався до спідометра (синій).

Вал, що обертає колеса автомобіля, з'єднаний зі спідометром довгим гнучким тросом із скручених проводів. Трос трохи нагадує міні-карданний вал: якщо один кінець троса обертається, то обертається і інший, навіть якщо трос довгий і гнучкий [4].

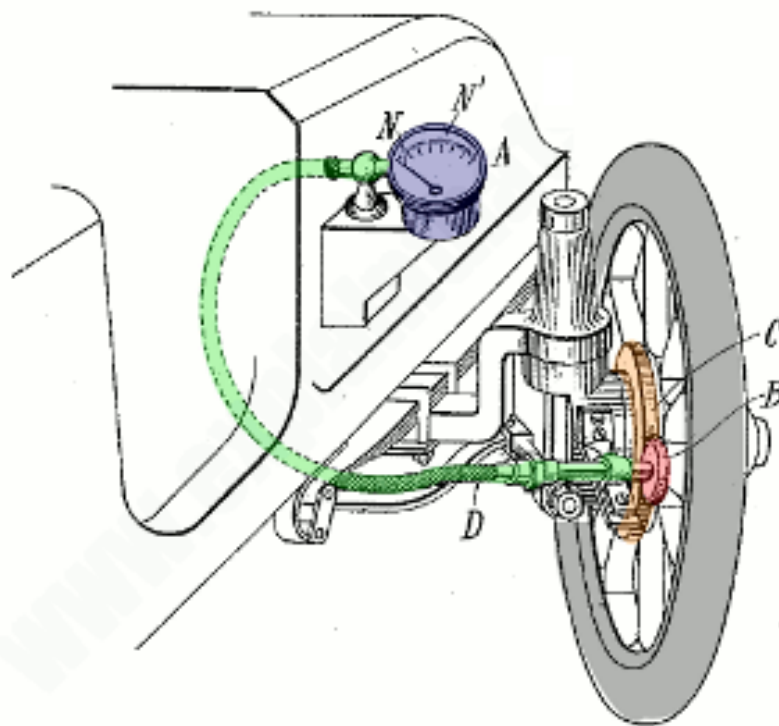
У верхньому кінці трос проходить у задню частину спідометра. Коли він обертається, він обертає магніт всередині корпусу спідометра з тією ж швидкістю. Магніт обертається всередині порожнистої металевої чашки, відомої як чашка швидкості, яка також може вільно обертатися, хоча й утримується тонкою котушкою дроту, відомою як зворотна пружина.

Однак магніт і чашка на якій закріплена стрілка ніяк не з'єднані між собою: їх розділяє повітря. Чашка швидкості прикріплена до покажчика, який рухається вгору і вниз по циферблату спідометра.

PATENTED JULY 26, 1904.

J. W. JONES.
SPEEDOMETER.

APPLICATION FILED MAR. 30, 1903.



Courtesy US Patent & Trademark Office

Рисунок 1.1 – Спідометр Джозефа В. Джонса

Розглянемо принципи функціонування спідометрів.

Коли трос спідометра обертається, він обертає магніт з тією ж швидкістю. Магніт, що обертається, створює флуктуаційне магнітне поле всередині чашки швидкості, і, за законами електромагнетизму, це означає, що електричні струми також протікають всередині чашки.

Фактично чашка швидкості перетворюється на своєрідний генератор електроенергії.

Але, на відміну від правильного генератора (такого типу, який виробляє електрику для вашого будинку на електростанції), струми в чаші швидкості нікуди подітися: їх енергію нема чим віднести.

Зм.	Арк.	№ докум.	Підпис	Дата

КВРКІ 180102.18.01.02 ПЗ

Арк.

9

Тож течії просто марно пливуть у закручених вирах — вони називаються вихровими течіями, або струми Фуко. Оскільки це електричні струми, і вони рухаються в електричному провіднику всередині магнітного поля, інший закон електромагнетизму говорить, що вони створять рух [5].

Струми фактично змушують чашу швидкості обертатися таким чином, що вона намагається наздогнати обертовий магніт. Але зворотна пружина не дає чашці обертатися дуже далеко, тому вона просто трохи повертається, підтягуючи вказівник вгору за циферблатом.

Чим швидше їде автомобіль, тим швидше обертається трос, чим швидше обертається магніт, чим більші вихрові струми він генерує, тим більша сила на чашку швидкості, і тим більше він може підтягнути вказівник вгору циферблата.

Практично всі спідометри, що випускалися до 1980-х років, працювали з використанням вихрових струмів і тросового механізму — так само, як оригінальний запатентований дизайн Шульце. Але є і недоліки.

По-перше, є багато механічних деталей, які зношуються (що робить їх неточними) або раптово виходять з ладу [6]. Якщо трос спідометра порветься, вся штуківина миттєво стає марною — і для ремонту потрібен механік.

Довгі троси спідометра особливо непрактичні, що завжди було проблемою у великих комерційних транспортних засобах, таких як вантажівки та автобуси. Вихреструміві спідометри також не ідеальні для велосипедів, не в останню чергу тому, що насправді немає місця для встановлення великого спідометра на кермі.

Не тільки трос створює проблему: може бути важкочитабельний звичайний циферблат спідометра, якщо водій мчить по автостраді, особливо вночі: він навряд чи захоче відірвати очі від дороги, щоб зрозуміти, де стрілка на циферблаті. Деякі люди вважають за краще бачити свою швидкість як просте число на добре освітленому цифровому дисплеї.

Цифрові спідометри було розроблено в 1993 році. Індикатором в такому спідометрі є рідкокристалічний цифровий або аналоговий дисплей[7]. Основним недоліком таких спідометрів є швидка зміна цифр перед очима водія у випадку цифрового дисплею, і затримка показів швидкості а у випадку аналогового дисплею.

Електронні спідометри працюють зовсім по-іншому. Маленькі магніти, прикріплені до обертового приводного валу автомобіля, проходять повз крихітні магнітні датчики (геркони або датчики Холла), розташовані поруч. Кожного разу, коли магніти проходять через датчики, вони генерують короткий імпульс електричного струму.

Електронна схема підраховує, як швидко надходять електричні імпульси, і перетворює це в швидкість, яка відображається в електронному вигляді на РК або OLED дисплеї [8].

Оскільки схема вимірює кількість обертів колеса, вона також може вести підрахунок того, скільки пройдено відстані, подвоюючи як одометр (метр для вимірювання відстані).

На рисунку 1.2 схематично зображено принцип роботи спідометра:

- 1) магніт, підключений до карданного валу, обертається з високою швидкістю. Кожен раз, коли він робить один повний оберт, він проходить повз датчик Холла;
- 2) датчик Холла (або геркон) в свою чергу реагує магнітне поле та запускає датчик;
- 3) мікроконтролер посилює сигнали від датчика і перетворює їх у миттєву швидкість, а лічильник відстані записує пройдену відстань;
- 4) цифровий OLED або РК дисплей на приладовій панелі виконує функцію і спідометра, і одометра, відображаючи швидкість і відстань поруч [9].

Електронні спідометри також можуть відображати швидкість аналоговопоказчики та циферблати, як і традиційні вихреструміві спідометри: в цьому випадку електронна схема приводить в дію висококерований

електродвигун (так званий кроковий двигун або сервопривід), який обертає вказівник на відповідний кут.

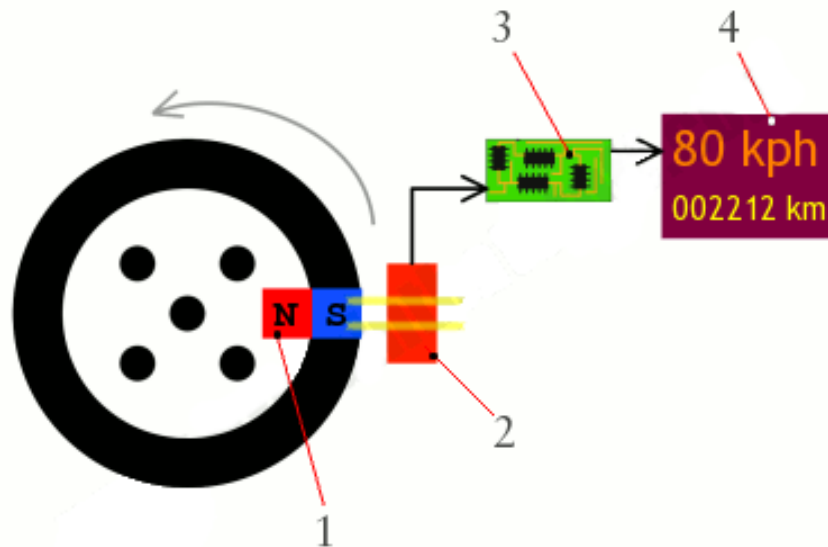


Рисунок 1.2 – Принцип роботи електронного спідометра.

Електронні спідометри більш надійні та компактні, ніж механічні, а датчики руху можуть бути на будь-якій відстані від дисплея, який показує вашу швидкість, що робить їх придатними для будь-якого типу транспортного засобу від велосипеда до 40-тонної вантажівки [10].

1.3 Аналіз готових приборів, призначених для вимірювання швидкості

Для сучасного життя контроль швидкості транспортного засобу став обов'язковим і дуже важливим для життєвих цілей [11].

Таким чином, першою метою є контроль швидкості транспортних засобів у різних районах міст, а також у зонах з обмеженою швидкістю, таких як лікарні, школи, коледжі, парки, зони з більшим населенням, зони обмеження швидкості тощо. швидке щоденне життя людей у світі, як правило, втрачають самоконтроль на своєму автомобілі. Вони їздять на такій великій швидкості, що може бути небезпечно як для себе, так і для оточуючих.

Навіть моніторингу поліції в таких випадках недостатньо [12]. Деякі ситуації навіть можуть бути смертельними.

Тому спідометр є важливим прибором на будь-якому транспортному засобі. Але на старих автомобілях спідометри зазвичай велика похибка в меншу сторону, тобто водій буде їхати з більшою швидкістю, ніж він думає і як наслідок в кращому випадку його зупинять на посту ДАІ, а в гіршому потрапить у автопригоду.

Використовуючи дані фактичних дорожньо-транспортних пригод, вчені з Університету Аделаїди оцінили відносний ризик того, що автомобіль потрапить в аварію, в якій загинули люди або були госпіталізовані, для автомобілів, які рухаються зі швидкістю 60 км/год або вище. Вони виявили, що ризик приблизно подвоюється на кожні 5 км/год вище 60 км/год [13].

Таким чином, автомобіль, який рухався зі швидкістю 65 км/год, мав у два рази більше шансів стати учасником аварії, ніж той, хто рухався зі швидкістю 60 км/год. Для автомобіля, який рухається зі швидкістю 70 км/год, ризик збільшився в чотири рази [14]. Очікується, що на швидкостях нижче 60 км/год ймовірність смертельної аварії буде відповідно зменшена.

Тому виникає потреба в установці нового, заміні старого або модернізації наявного приладу контролю швидкості.

На сьогоднішній день ринок пропонує великий асортимент різних пристроїв з різним типом кріплення та вираховування швидкості, а також великим різноманіттям додаткових функцій та можливостей.

На рисунку 1.3 зображено GPS-спідометр HUD C1090.

Особливості GPS-спідометра:

1) головний дисплей швидкості руху, розрахунок та відображення часу руху та пробігу, просте налаштування – потрібно лиш підключити живлення через прикурювач автомобіля і можна користуватись [15]. Не вимагає професійного монтажу;

2) сумісний з усіма автомобілями та вантажівками, мотоциклами та велосипедами. Використовує обчислення, керовані GPS. Підключення до VSS (сигналу швидкості транспортного засобу) не потрібно;

3) дисплей MPH або КМ/Г, автоматичне регулювання яскравості за допомогою датчика для денної та нічної їзди. Відображення швидкості руху автомобіля. Зелений індикатор швидкості, який зручний для ока;

4) налаштування сигналізації перевищення швидкості. Відображення загального пробігу після зупинки автомобіля. Дисплей висоти. Сигналізація за перевищення швидкості. Ви можете встановити різні значення швидкості тривоги, сигнал про втому і кожні 1 годину безперервної роботи тривога спрацьовує один раз;

5) статистика пробігу та часу їзди. Автоматичне відображення характеристик водіння, поточного часу водіння та загальної ефективності водіння. Інтелектуальний контроль яскравості дисплея. При сильному освітленні яскравість індикатора швидкості автоматично збільшується, при слабкому освітленні яскравість індикатора швидкості автоматично зменшується [16].



Рисунок 1.3 – GPS-спідометр HUD C1090.

					КвРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Недоліки GPS-спідометра:

- 1) пристрій вичисляє швидкість за допомогою сигналу GPS, що дає затримку та інертність значень;
- 2) якщо авто буде рухатись в тунелі, то даний пристрій не буде працювати через відсутність спутникового сигналу;
- 3) після запуску пристрою потрібний час на пошук спутників;
- 4) власники пристрою повідомляють про неправильну роботу висотоміра та погане регулювання яскравості дисплея;
- 5) Вартість пристрою – 1 499 гривень за комплект.

На рисунку 1.4 зображено проєкційний OBD GPS спідометр Q10.



Рисунок 1.4 – Проєкційний OBD GPS спідометр Q10

Особливості проєкційного OBD GPS спідометра Q10:

- 1) подвійна робоча система OBD + GPS 2 в 1 плюс система навігації на карті Google;
- 2) відображення інформації в режимі OBD: швидкість автомобіля, температура охолоджуючої рідини, напруга, витрата палива, оберти двигуна, пробіг, час у дорозі, напрямок руху, системний час, тиск, температура масла, співвідношення повітря-паливо, тиск турбіни, читання потоку даних, навігація;

					КвРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

3) сигнали тривоги: перевищення швидкості, висока температура води, низька напруга, несправність двигуна, вільне перемикання км/год та миль/год, вільне перемикання °С;

4) відображення інформації в режимі GPS: швидкість, час у дорозі, час супутника, висота, кількість супутників, напрямок руху, пробіг, напруга;

5) сигнали тривоги: перевищення швидкості, низька напруга.

Недоліки проекційна OBD GPS спідометра:

1) у денний час не видно даних, які відбиваються на склі;

2) підходить тільки для тих типів автомобілів, у яких наявний OBD роз'єм;

3) власники пристрою заявляють, що робота в режимі GPS досить неточна, а сигнал нестабільний [17];

4) Вартість пристрою – 3200 гривень.

Особливості Pdtoweb OBD + GPS спідометра у автомобілі, що зображено на рисунку 1.5:

1) OBD + GPS подвійна система, функція GPS може бути показана в режимі OBD;

2) в даний час перший HUD з сигнальним навколишнім світлом (навколишнє світло, як правило, синє, і блимає червоним, коли відбувається тривога);

3) існує три блоки тиску турбіни: PSI, 100KPA і бар. Значення турбіни може бути скориговано та відкалібровано для збільшення блоку витрати палива КМ/л;

4) Частота оновлення до 10 Гц, явище затримки заїкання;

5) Коли швидкість переключється на MPH, температура води автоматично переключиться на Фаренгейт, одна відстань руху автоматично переключиться на Miles, а витрата палива автоматично переключиться на Gallon MPG;

6) Простий в установці і використанні [18].

Недоліки OBD + GPS спідометра:

1) підходить тільки для тих типів автомобілів, у яких наявний obd роз'єм;

2) дуже повільна зміна даних в режимі gps;

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

- 3) власники пристрою заявляють, що в режимі підключення по інтерфейсу obd працює не зі всіма автомобілями;
- 4) двосторонній скотч, який використовується для кріплення до приборної панелі, низької якості;
- 5) Вартість пристрою – 1 800 гривень.



Рисунок 1.5 – Pdtoweb новий OBD + GPS спідометр у автомобіль.

На рисунку 1.6 зображено універсальний цифровий спідометр-одометр для автомобіля [19].

Особливості:

- 1) діапазон виміру: 0-160 км/год;
- 2) пробіг: 0-999999,9 км;
- 3) оповіщення про перевищення швидкості;
- 4) заводська сигналізація про підвищену швидкість: 80 км/год;
- 5) при перевищенні швидкості, у верхньому правому куті відображається оповіщення, без звукової підказки;

б) точність виміру: $\pm 0.1\%$.



Рисунок 1.6 – Універсальний цифровий спідометр-одометр для автомобіля

Недоліки:

- 1) в комплекті немає датчика, потрібно докупляти;
- 2) погана видимість дисплея;
- 3) власники скаржаться на низьку якість корпусу і кріпильних матеріалів;
- 4) Вартість пристрою – 895 гривень.

1.4 Висновки

На даний момент є багато морально застарілих автомобілів, які все ще активно використовуються, у яких стандартний прилад виміру швидкості вийшов з ладу, а нових приладів немає в наявності, або покупка останніх є нерентабельною.

Тому виникає завдання розробки бюджетного та компактного програмно-технічного пристрою для вимірювання миттєвої швидкості руху з добовим та загальним лічильником відстані з можливістю встановлення у корпус з під стандартного спідометра з незначним втручанням в конструкцію корпусу

спідометра. Потрібно отримати імпульси з датчика через оптопару, вирахувати за формулою і вивести на OLED дисплей з дублюванням швидкості на сервопривід.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

2 АНАЛІЗ ТА ВИБІР ЕЛЕМЕНТНОЇ БАЗИ ДЛЯ ЛІЧИЛЬНИКА МИТТЄВОЇ ШВИДКОСТІ РУХУ

2.1 Аналіз функціонування пристрою для вимірювання швидкості

Спідометри в автомобілі – це просте, але корисне обладнання, яке виконує важливу мету. Це дає змогу водієві стежити за швидкістю транспортного засобу та уникати штрафів за перевищення швидкості. Крім цього, він також може записувати пробіг, подоланий автомобілем.

Спідометр працює на основі датчиків і мікросхем, виключаючи механічні з'єднання між валом і спідометром автомобіля. Він складається з таких частин: магніт, датчик швидкості (датчик Холла або геркон), електричне коло, OLED або РК-дисплей.

Магніти прикріплені до зовнішнього валу з магнітними датчиками на протилежній стороні. Кожного разу, коли магніти перетинають датчики, в ланцюзі генерується електричний імпульс. Схема перетворює цей імпульс у швидкість і відображає його на дисплеї. Оскільки кількість обертів колеса вимірюється, він також може відображати показання одометра на тому самому дисплеї [20].

Датчик швидкості – один із безліччі автомобільних датчиків, відповідальних за вироблення сигналів вимірювальної інформації, її передачу, подальше перетворення та обробку електронним блоком управління та деякими іншими пристроями. Якщо в автомобілі немає ЕБУ, потреба в датчиках меншою не стає. Вихід навіть менш важливого з датчиків позначається хоча б на тому, як комфортно буде експлуатувати автомобіль.

Взагалі, автомобільні датчики швидкості діляться лише на два типи:

- 1) електронний;
- 2) механічний.

Почати варто з механічного, адже хронологічно саме він передував досконалішому електронному пристрої. В основі механічного датчика лежить група зчеплених шестерень і невеликий трос. Встановлювали такі датчики прямо

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

на механізмах спідометра приводу недалеко від КПП. Механічні пристрої дуже нехитрі і мають солідний експлуатаційний ресурс, проте не дають точних показань на всьому діапазоні швидкостей і все-таки залежать від низки зовнішніх умов. Чого не скажеш про електронні датчики швидкості [21].

Сучасні датчики ґрунтуються на ефекті Холла. Датчик формує так званий імпульсний сигнал, частота проходження імпульсів в якому має залежність від швидкості обертання валу. Автомобіль рухається швидше – вал обертається швидше – датчик виробляє імпульси більшої частоти – на спідометр виводиться достовірна інформація про зміну швидкості у більшу сторону. Зрозуміло, "звичний" для електроніки сигнал людина зрозуміти не може. З цієї причини в систему вводиться контролер, що підраховує частоту імпульсів, що надходять від датчика швидкості в одиницю часу, а потім переводить цю величину в людино-зрозумілі км/год або миль/год.

Електронні датчики бувають двох видів:

- 1) з контактом від валу;
- 2) без контакту.

Перші датчики просто називають контактними. У них використовують приводні шестірні та гнучкий трос (іноді жорсткий вал невеликої довжини). Трос або вал служать для передачі крутного моменту від автомобільного мосту, валу коробки передач або роздавальної коробки. Кутове обертання перетворюється на електричні імпульси, які передаються далі за системою і перетворюються мікроконтролером на людино-зрозумілі величини. Саме такі датчики знайшли найширше застосування у автомобільній індустрії. Причин дві: вони надійні, їх можна використати замість механічного приводу спідометра без дорогих доопрацювань останнього [22].

Все більш популярні безконтактні датчики ґрунтуються на тому ж ефекті Холла, але технічно реалізуються не так, як контактні. Вони використовують один із допоміжних пристроїв: ротор або диск, що задає. У безконтактних датчиках ефект Холла використовується повною мірою, тим часом як менш складних

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

контактних – той самий ефект, магниторезистивний ефект чи робота оптронів (оптроелектронних пар).

Безконтактні датчики швидкості ґрунтуються на згаданому ефекті Холла, внаслідок чого вони не мають рухомих частин. Суть ефекту Холла в тому, що на плоскому проводку, через який з протилежних сторін пропускається постійний струм, при його знаходженні в магнітному полі виникає напруга на парі інших протилежних сторін [23]. Для роботи датчиків потрібно розмістити на валу агрегату, яким може бути міст, редуктор або коробка передач, імпульсний диск або спеціальний диск. Дані елементи мають намагнічені ділянки. Імпульсний сигнал утворюється завдяки тому, що ротор, віддалений від чутливої частини автомобільного датчика з мікросхемою Холла, починає обертатися. Далі сигнал надходить до контролера.

Контактні датчики швидкості, що використовують ефект Холла, та їх однойменна мікросхема з магнітом постійно нерухомі, а магнітне поле змінюється завдяки обертанню спеціального кільця з прорізами, що називається інакше шторкою. Саме кільце приєднано до валу або гнучкого приводного троса, через яке обертання і передається.

Оптоелектронні контактні датчики швидкості дуже прості, але це єдина їхня перевага. Справа в тому, що вони менш чутливі (до відхилень основного параметра) і більш інерційні (мають велике запізнення у вимірі), ніж описані вище датчики. Працює датчик за рахунок оптопари, що є фототранзистором і світлодіодом, розділені диском з прорізами. Останній закріплений на приводному валу. За рахунок обертання диска та переривання світлового потоку між парою елементів та генерується імпульсний сигнал [24].

У систему визначення швидкості введено спеціальний контролер, який приймає імпульсний сигнал від датчика. Саме контролер передає сигнал електронному блоку управління, який розраховує обсяг палива, необхідний для оптимальної роботи двигуна. Наприклад, якщо швидкість автомобіля зменшується,

зменшується кількість палива, яке подається двигуну. За рахунок цього вдається суттєво економити паливе та експлуатувати двигун у найбільш щадному режимі.

На рисунку 2.1 схематичне зображення роботи датчиків.

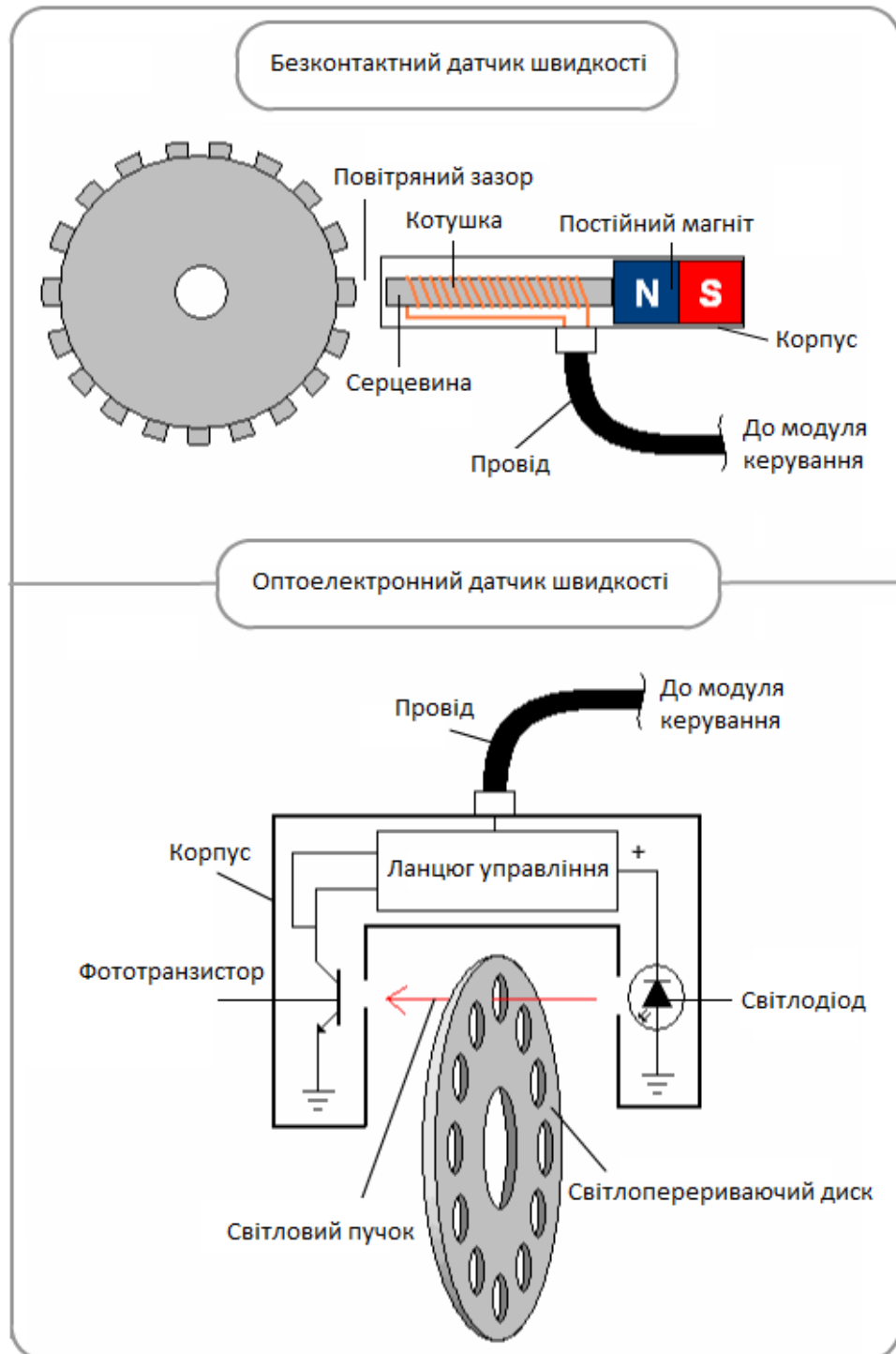


Рисунок 2.1 – Схематичне зображення роботи датчиків

У разі несправності датчика швидкості блок управління не отримуватиме сигнал, що відповідає реальній швидкості автомобіля. Паливо подаватиметься рівномірно незалежно від того, як сильно водій втискає педаль газу. Насправді виходить так, що пальне витрачається з надлишком, а силовий агрегат іноді працює з ривками. Статистика показує, що справний датчик дозволяє заощаджувати близько 2 літрів палива на 100 кілометрів пробігу [25].

Як і практично всі електронні компоненти авто, датчик страждає насамперед від окислення контактів та порушення цілісності проводки. Найчастіше проблему вдається виявити під час візуального огляду, хоча вірності варто скористатися тестером. Кожен із контактів потрібно від'єднати та перевірити. Окислені і просто брудні контакти потрібно очистити і покрити спеціальним мастилом. Інше слабке місце найпоширеніших систем визначення швидкості авто: трос спідометра . Згодом він надривається відразу в кількох місцях, що в результаті позначається на правильності визначення швидкості автомобіля. Старий трос потрібно міняти, а перед установкою про всяк випадок покрити машинним маслом. На рисунку 2.2 зображено датчик швидкості Mazda 3.



Рисунок 2.2 – Датчик швидкості Mazda 3

Найбільш поширені і на вигляд надійні системи визначення швидкості не терплять їзди з різкими підвищеннями швидкості. Через це швидко порушується цілісність пластикового хвостика у датчику. Також починає розшаровуватися і тріскати трос. Як і було сказано вище, трос потрібно змащувати. Щільність кріплення пластикового хвостика потрібно контролювати - вона повинна бути високою, що запобігає подальшому розбобтуванню гнізда кріплення. Якщо є підозри, що встановлений в автомобілі датчик швидкості працює неправильно, не потрібно поспішати з висновками.

Для початку потрібно переконатися в чомусь із наступного:

- 1) є проблеми на неодружених оборотах (машина періодично глухне, в т.ч. і при русі накатом);
- 2) перестав працювати спідометр;
- 3) спідометр працює некоректно;
- 4) серйозно зросла витрата палива;
- 5) двигун втрачає потужність при швидкому розгоні.

Як і було зазначено вище, датчик слід оглянути. На різних автомобілях датчик швидкості розташований по-різному, але, загалом, всі автоконцерни намагаються розташувати його ближче до коробки передач. Часто його мають прямо над головною передачею. Зазвичай датчик кріпиться гайкою. Коли датчик знято, потрібно зробити наступне:

- 1) визначитись з клемами: заземлення, імпульсного сигналу та напруги;
- 2) контакт вольтметра підключити до клеми імпульсного сигналу;
- 3) другий контакт заземлити на корпус автомобіля або будь-яку металеву деталь ДВЗ;
- 4) прокрутити вісь датчика шматком пластику;
- 5) показники частоти та обертання на вольтметрі повинні зростати по ходу збільшення швидкості обертання осі датчика.

Така перевірка є актуальною лише для датчиків, що використовують ефект Холла. Щось подібне можна повернути не демонтуючи датчик. Для цього

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

потрібно вивісити провідне колесо на домкраті, аналогічно підключити вольтметр або інший вимірювальний прилад, після чого повернути вивішене колесо і стежити за показниками напруги і частоти.

2.2 Аналіз та вибір мікроконтролера

Мікроконтролер (або MCU для «блок мікроконтролера») — це невелика інтегральна схема, яка керує електронним пристроєм. Вона схожа на мікропроцесор, але містить пам'ять та інші інтегровані компоненти.

Типовий мікроконтролер включає в себе наступні елементи:

- 1) ЦП – від простого 4-розрядного процесора до повноцінного 64-розрядного процесора;
- 2) RAM – високошвидкісна енергозалежна пам'ять, яка використовується для тимчасового зберігання даних;
- 3) Флеш-пам'ять – тип енергонезалежної пам'яті, що використовується для зберігання даних з електроживленням або без нього;
- 4) Інтерфейс введення-виведення – послідовні порти або інші роз'єми для периферійних пристроїв.

Майже всі електронні пристрої містять мікроконтролери. Багато з них включають кілька MCU, які працюють в тандемі. Наприклад, автомобіль може містити десятки мікроконтролерів для сенсорів, комунікації та функцій інтерфейсу користувача.

Бездротові комунікації - передавачі або приймачі для передачі та прийому бездротових сигналів. Мікроконтролери можна зустріти в багатьох сучасних приладах, таких як телефони, пральні машини, вони відповідають за роботу двигунів і систем гальмування сучасних автомобілів, з їх допомогою створюються системи контролю і системи збору інформації. На основі мікроконтролерів проектують та створюють вимірювальні пристрої, системи керування об'єктами та процесами, вони є основою охоронних, протипожежних систем, домофонів,

сигналізацій тощо. Більшість процесорів, що випускаються у світі це є мікроконтролери.

Мікроконтролери поділяються на різні категорії на основі пам'яті, архітектури, біт та наборів команд. Залежно від конфігурації розрядності мікроконтролер поділяється на три категорії:

1) 8-розрядний мікроконтролер — цей тип мікроконтролера використовується для виконання арифметичних та логічних операцій, таких як додавання, віднімання, ділення множення тощо. Наприклад, Intel 8031 і 8051 є 8-розрядними мікроконтролерами;

2) 16-розрядний мікроконтролер — цей тип мікроконтролера використовується для виконання арифметичних і логічних операцій, де потрібна більш висока точність і продуктивність. Наприклад, Intel 8096 — це 16-розрядний мікроконтролер;

3) 32-розрядний мікроконтролер — цей тип мікроконтролера зазвичай використовується в автоматично керованих пристроях, таких як автоматичні операційні машини, медичні прилади тощо.

Залежно від конфігурації пам'яті мікроконтролер поділяється на дві категорії:

1) мікроконтролер зовнішньої пам'яті — цей тип мікроконтролерів розроблений таким чином, що вони не мають програмної пам'яті на чіпі. Тому його називають мікроконтролером зовнішньої пам'яті. Наприклад: мікроконтролер Intel 8031;

2) мікроконтролер із вбудованою пам'яттю — цей тип мікроконтролера розроблений таким чином, що мікроконтролер має всі програми та пам'ять даних, лічильники та таймери, переривання, порти вводу-виводу вбудовані в чіп. Наприклад: мікроконтролер Intel 8051.

Залежно від конфігурації набору інструкцій мікроконтролер поділяється на дві категорії:

1. CISC означає комп'ютер зі складним набором інструкцій. Це дозволяє користувачеві вставити одну інструкцію як альтернативу багатьом простим інструкціям;

2. RISC означає «Комп'ютери зі скороченим набором інструкцій». Це скорочує час роботи, скорочуючи тактовий цикл на інструкцію.

В основному програмування мікроконтролерів здійснюється на мові C чи на мові Assembler. Крім того, може використовуватися таке спеціальне програмне забезпечення для налагодження і створення програм – MCSStudio, AVRStudio, ArduinoIDE та інші.

Мікроконтролер, який використовується у даному завданні має виконувати наступні завдання:

- 1) зчитування імпульсів з датчика через опторозв'язку;
- 2) обчислення n-ої кількості імпульсів за формулою;
- 3) відображення значень на дисплеї та сервоприводі;
- 4) відображення загального та добового лічильника відстані.

Для реалізації вище перерахованих завдань потрібно програмно описати зовнішні переривання, а саме обнулення таймера для виводу наступного знаку. Для того, щоб забезпечити індикацію потрібно описати роботу портів та створити підпрограму, яка виводитиме значення на індикатор.

Для виконання даної задачі була взята плата Arduino Nano, на базі мікроконтролера ATMEGA328P. Структурна схема мікроконтроллера ATMEGA328P зображено на рисунку 2.3, а розпинівка плати Arduino Nano на рисунку 2.4.

Загальні характеристики даного мікроконтроллера:

- 1) тактова частота: 0 - 20 МГц;
- 2) обсяг Flash-пам'яті: 32 кб;
- 3) обсяг SRAM-пам'яті: 2 кб;
- 4) обсяг EEPROM-пам'яті: 1 кб;
- 5) напруга живлення: 1,8 - 5,5 В;

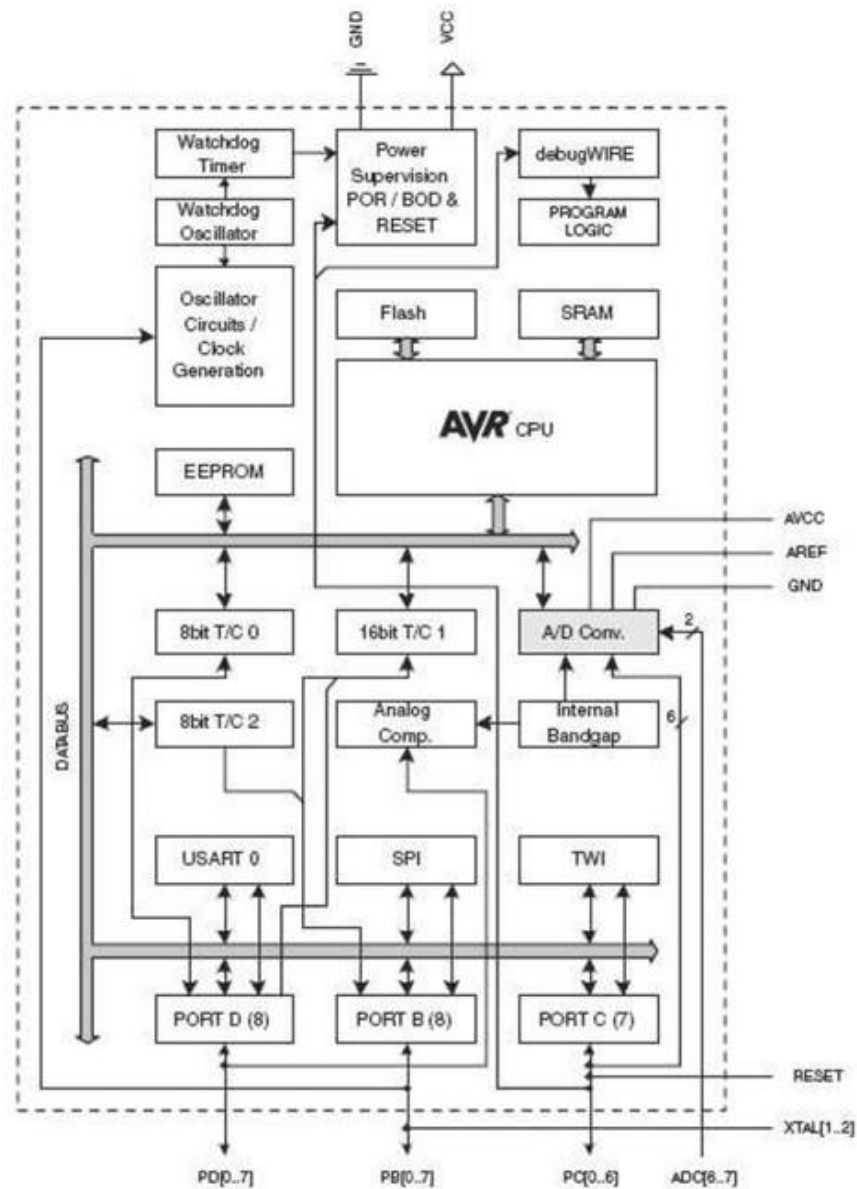


Рисунок 2.3 – Структурна схема мікроконтролера ATMEGA328P

- 6) споживаний струм в режимі роботи: 0,2 мА (1 МГц, 1,8 В);
- 7) споживаний струм в режимі сну: 0,75 мкА (1 МГц, 1,8 В);
- 8) кількість таймерів / лічильників: 2 восьмибітних, 1 шістнадцятибітну;
- 9) загальна кількість портів: 23;
- 10) кількість ШІМ (PWM) виходів: 6;
- 11) кількість каналів АЦП (аналогові входи): 6;
- 12) кількість апаратних USART (Serial): 1;
- 13) кількість апаратних SPI: 1 Master / Slave;

14) кількість апаратних I²C / SPI: 1;

15) роздільна здатність АЦП: 10 біт.

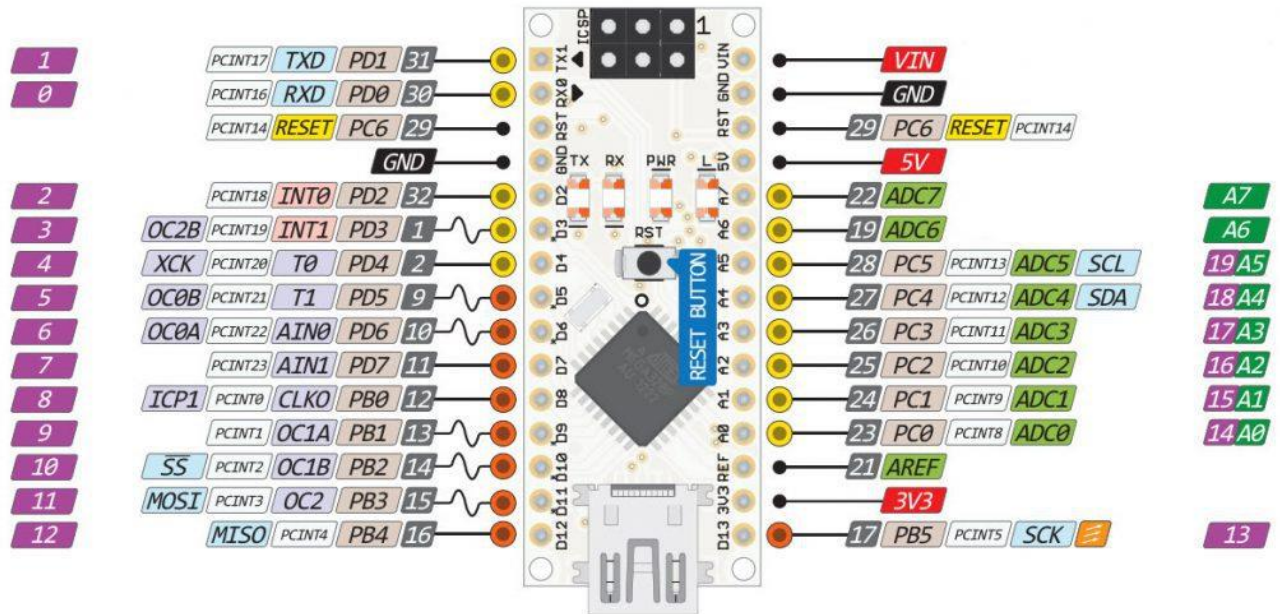


Рисунок 2.4 – Розпіновка плати Arduino Nano

Крім того, даний мікроконтролер забезпечує 1000 циклів стирання / запису та режим одночасного читання/запису. АТМЕГА328Р характеризується додатковим сектором завантажувальних кодів з незалежними бітами блокування та внутрішньосистемного програмування вбудованої програмою завантаження.

Також даний мікроконтролер встановлюється на платі Arduino. Вигляд зверху та знизу плати Arduino Nano зображено на рисунку 2.5.

Плата Arduino буде працювати разом із OLED дисплеєм, сервоприводом та енкодером, а приймати переривання буде з датчика швидкості через оптопару [26].

Сервопривід (сервомеханізм) — це електромагнітний пристрій, який перетворює електрику в точний керований рух за допомогою механізмів негативного зворотного зв'язку. Сервопривід Tower Pro MG90S зображено на рисунку 2.6.

Сервоприводи можуть використовуватися для створення лінійного або кругового руху, залежно від їх типу.



Рисунок 2.5 – Вигляд зверху та знизу плати Arduino Nano

Типовий сервопривід включає двигун постійного струму , редуктор, потенціометр, інтегральну схему (ІС) і вихідний вал. Бажана позиція сервопривода вводиться і надходить у вигляді кодованого сигналу на ІС.



Рисунок 2.6 – Сервопривід Tower Pro MG90S

Зм.	Арк.	№ докум.	Підпис	Дата

Інтегральна мікросхема спрямовує двигун рухатися, рухаючи енергію двигуна через шестерні, які встановлюють швидкість і бажаний напрямок руху, доки сигнал з потенціометра не подає зворотній зв'язок, що бажане положення досягнуто, і ІС зупиняє двигун.

Потенціометр робить можливим контрольований рух шляхом передачі поточного положення, одночасно дозволяючи коригувати зовнішні сили, що діють на контрольні поверхні: після переміщення поверхні потенціометр надає сигнал положення, а ІС сигналізує про необхідний рух двигуна, поки не буде відновлено правильне положення.

Інкрементний поворотний енкодер забезпечує чудовий зворотний зв'язок щодо швидкості та відстані. Він простіший, ніж інший тип поворотного кодера, абсолютний поворотний енкодер, і має менше датчиків, що робить його менш дорогим з меншою кількістю точок відмови. Однак інкрементний кодер може надавати лише інформацію про зміну руху, оскільки він може надавати циклічні вихідні дані лише при обертанні вала з кодером, тому для обчислення руху потрібен еталонний пристрій.

Датчик, який використовується в інкрементному обертовому кодері, може бути механічним або оптичним. Для механічного типу потрібен процес, який називається «відбій», який є способом зробити вихід чистішим, плавнішим і послідовнішим, а не коливанням або «підстрибуванням».

Цей тип поворотного кодера зазвичай використовується в якості цифрового потенціометра в деякому обладнанні та споживчих пристроях, таких як циферблат радіо або автомобільної стереосистеми або датчик руху в кульковій миші. Але через механічну природу датчика швидкість, з якою він може працювати, обмежена [27].

Однак найкращою особливістю механічного типу є його низька вартість, і, незважаючи на наявність лише двох датчиків, його роздільна здатність зовсім не

впливає. Існують інкрементні кодери, які мають до 10 000 або більше відліків на оберт.

Два датчики інкрементного кодера дозволяють йому визначати напрямок обертання, оскільки два датчики розходяться на 90 градусів з точки зору генерованих сигналів (квадратурних виходів), тому результуючі значення кожного датчика разом визначають, чи є рух за годинниковою стрілкою або проти годинникової стрілки.

Комбінація сервоприводів і електродвигунів з багатьма редукторами може бути організована разом для виконання більш складних завдань у різних типах систем, включаючи роботів, транспортні засоби, виробництво та бездротову мережу датчиків і виконавчих механізмів (WSAN). Тип двигуна не є критичним і може бути використано різні їх типи. Найпростіший, який використовують, — на основі постійного магніту і щіток, з огляду на його дешевизну і конструктивну простоту.

За допомогою сервоприводу Tower Pro MG90S буде здійснюватися рух стрілки по шкалі пристрою, а за допомогою OLED дисплея 128×64 буде відображатися великим шрифтом поточна швидкість, а загальна пройдена відстань або добова пройдена відстань буде відображатися маленьким шрифтом.

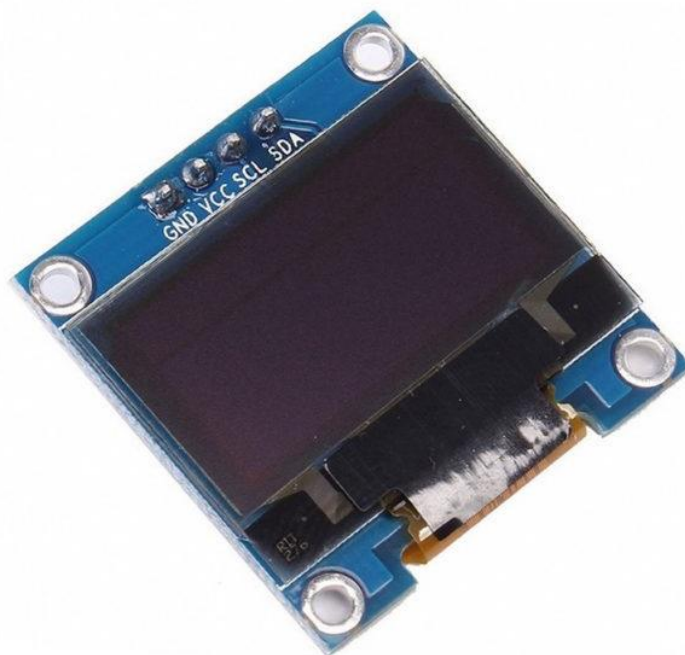


Рисунок 2.7 – OLED дисплей 0.96" I2C 128x64

OLED дисплеї, що зображено на рисунку 2.7, на сьогоднішній день є найдосконалішими типами дисплеїв, і причин тому багато: широкі кути огляду, низьке енергоспоживання, мала товщина дисплеїв та відсутність необхідності у зовнішньому підсвічуванні.

Найбільша відмінність між OLED і РК-дисплеями полягає в тому, що самі OLED випромінюють світло: OLED, які складають кожен піксель (субпікселі), світяться у відповідних кольорах, усуваючи потребу в підсвічуванні або кольорових фільтрах, і забезпечуючи більш тонкі конструкції. Це також означає, що, оскільки чорний відображається при простому вимкненні цих OLED, ці дисплеї можуть відображати чисто чорний колір.

Для порівняння, РК-монітори використовують РК-панель як затвори для підсвічування: світло від підсвічування регулюється затворами, регулюючи інтенсивність світла для відображення різних кольорів.

За своєю природою РК-монітори потребують підсвічування та кольорових фільтрів, тому є межі того, наскільки вони можуть бути тонкими. РК-дисплеї також не в змозі повністю заблокувати світло, що виробляється підсвічуванням, що призводить до витоку світла та мутного чорного на темних зображеннях, як-от нічний пейзаж.

OLED дисплей із розширенням 128×64 (0.96 дюйма) складається з двох частин: із самого дисплея з контролером, від якого йде гнучкий шлейф на зворотний бік плати, та плати модуля. Керується такий модуль за допомогою I2C-інтерфейсу.

I2C розшифровується як Inter-Integrated Circuit. Це протокол підключення шинного інтерфейсу, вбудований в пристрої для послідовного зв'язку. Він був розроблений компанією Philips Semiconductor у 1982 році. Останнім часом він широко використовується як протокол для зв'язку на короткі відстані. Він також відомий як двопровідний інтерфейс (TWI).

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Протокол зв'язку I2C використовує лише 2 двонаправлені лінії з відкритим стоком для передачі даних, які називаються SDA і SCL. Обидві ці лінії підтягуються до плюса живлення через резистори.

I2C працює в 2 режимах:

- 1) головний режим;
- 2) підпорядкований режим.

Кожен біт даних, переданий по лінії SDA, синхронізується імпульсом високого та нижчого рівня кожного тактового сигналу на лінії SCL.

Відповідно до протоколів I2C, лінія даних не може змінюватися, коли тактова лінія має високий рівень, вона може змінюватися лише тоді, коли лінія тактової частоти низька. 2 лінії мають відкритий сток, отже, потрібен підтягуючий резистор, щоб лінії були високими, оскільки пристрої на шині I2C активні низьким рівнем. Дані передаються у вигляді пакетів, які містять 9 біт.

На рисунку 1.12 схематично наведено принцип підключення декількох пристроїв по шині I2C [28].

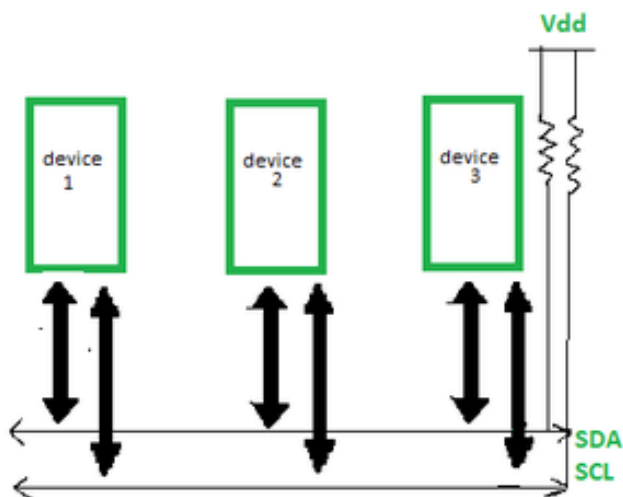


Рисунок 2.8 – Принцип підключення декількох пристроїв з різними напругами живлення до шини обміну

Послідовність цих бітів:

- 1) початкова умова – 1 біт;

Зм.	Арк.	№ докум.	Підпис	Дата

- 2) адреса підпорядкованого пристрою – 8 біт;
- 3) підтвердження – 1 біт.

Умова запуску та зупинки передачі даних зображено на рисунку 1.13.

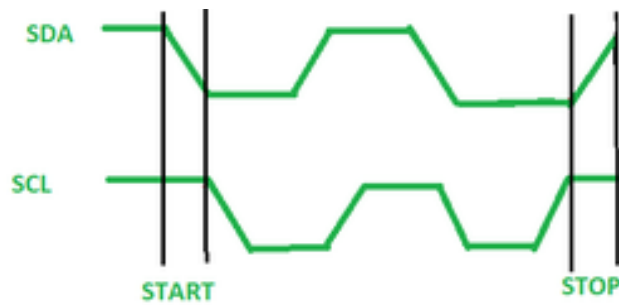


Рис. 2.9 – Умова запуску та зупинки передачі даних.

Інкрементний поворотний енкадер (також відомий як квадратурний енкадер) – це тип електромеханічного пристрою, який перетворює кутовий рух або положення обертового вала в аналоговий або цифровий код, який представляє цей рух або положення. Його можна використовувати для додатків зі зворотним зв'язком швидкості двигуна та положення, які включають контур керування сервоприводом, а також для промислових застосувань від легких до важких. Модуль енкадера EC11 з кнопкою зображено на рисунку 1.14.

Інкрементальний поворотний енкадер забезпечує чудовий зворотний зв'язок щодо швидкості та відстані. Він простіший, ніж інший тип поворотного кодера, абсолютний поворотний енкадер, і має менше датчиків, що робить його менш дорогим з меншою кількістю точок відмови. Однак інкрементний енкадер може надавати лише інформацію про зміну руху, оскільки він може надавати циклічні вихідні дані лише при обертанні вала з енкадером, тому для обчислення руху потрібен еталонний пристрій.

Датчик, що використовується в інкрементному поворотному енкадері, може бути механічним або оптичним. Для механічного типу потрібен процес, який називається «відбій», який є способом зробити вихід чистішим, плавнішим і послідовнішим, а не коливанням або «підстрибуванням».



Рисунок 2.10 – Модуль енкодера EC11 з кнопкою

Цей тип поворотного енкодера зазвичай використовується в якості цифрового потенціометра в деякому обладнанні та споживчих пристроях, таких як циферблат радіо або автомобільної стереосистеми або датчик руху в кульковій миші. Але через механічну природу датчика швидкість, з якою він може працювати, обмежена. Однак найкращою особливістю механічного типу є його низька вартість, і, незважаючи на наявність лише двох датчиків, його роздільна здатність зовсім не впливає. Існують інкрементні енкодери, які мають до 10 000 або більше відліків на оберт.

Два датчики інкрементного енкодера дозволяють йому визначати напрямок обертання, оскільки два датчики розходяться на 90 градусів з точки зору генерованих сигналів (квадратурних виходів), тому результуючі значення кожного датчика разом визначають, чи є рух за годинниковою стрілкою або проти годинникової стрілки.

Для розробки коду і прошивки в мікроконтроллер буде використовуватися додаток для ПК Arduino IDE. Вигляд Arduino IDE показано на рисунку 2.11.

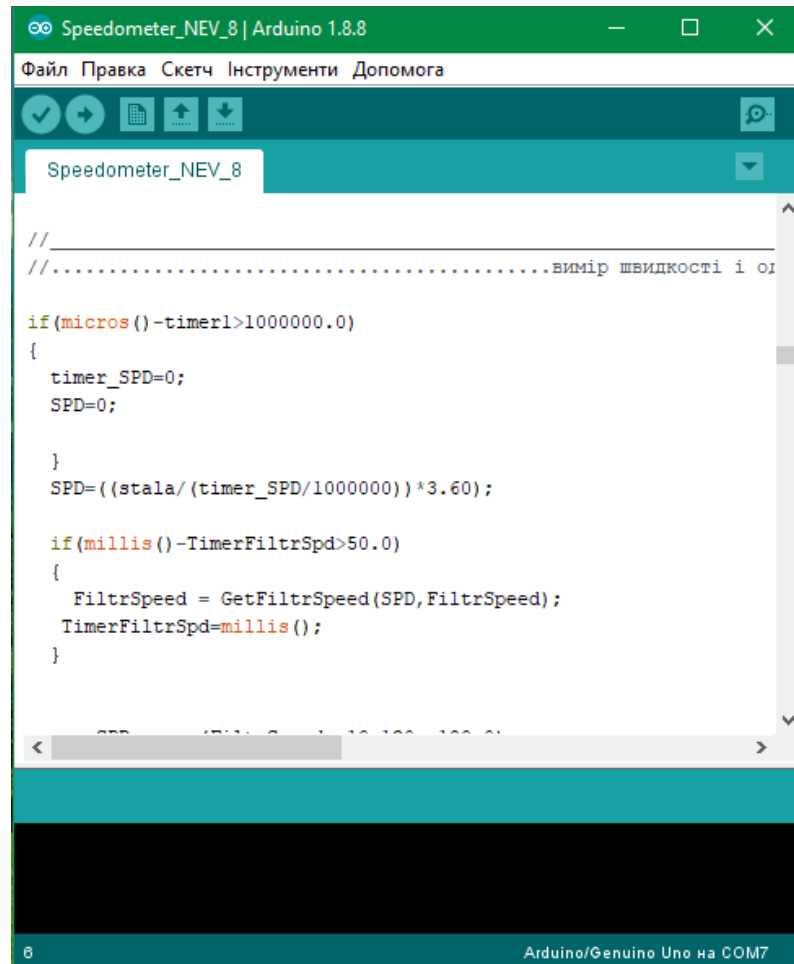


Рисунок 2.11 - Програмний інтерфейс Arduino

Програмне забезпечення Arduino (IDE) — це програмне забезпечення з відкритим вихідним кодом, яке використовується для програмування плат Arduino, і є інтегрованим середовищем розробки, розробленим arduino.cc . Дозволяє писати та завантажувати код на плати Arduino. І він складається з безлічі бібліотек і набору прикладів міні-проектів.

Програмне забезпечення arduino (IDE) сумісне з різними операційними системами (Windows, Linux, Mac OS X) і підтримує мови програмування (C/C++)

Програмне забезпечення Arduino просте у використанні як початківцям, так і досвідченим користувачам. Він використовується для початку роботи з програмуванням в електроніці та робототехнікою, а також для створення інтерактивних прототипів.

Отже, програмне забезпечення Arduino – це інструмент для розробки нових речей. і створювати нові електронні проекти, будь-хто (діти, любителі, інженери, програмісти тощо).

2.3 Обґрунтування вибору програмного забезпечення.

Для роботи з мікроконтролерами типу ATMEGA 328P використовують програмне середовище для розробки Arduino IDE.

Arduino IDE – це програмне забезпечення з відкритим вихідним кодом, яке використовується для написання та завантаження коду на плати Arduino. IDE підходить для різних операційних систем, таких як Windows, Mac OS X і Linux. Він підтримує мови програмування C та C++. IDE тут означає інтегроване середовище розробки.

Програму або код, написаний Arduino IDE, часто називають ескізом. Нам потрібно підключити плату Genuino та Arduino до IDE, щоб записати скетч, написаний у програмному забезпеченні Arduino IDE. Ескіз зберігається з розширенням ".ino". Arduino IDE показано на рисунку 2.3.

На панелі інструментів відображаються піктограми «Створити», «Відкрити», «Зберегти», «Завантажити» та «Перевірити», що показано на рисунку 2.4.

Кнопка «Вивантажити» компілює та виконує наш код, написаний на екрані. Потім завантажує код на прикріплену дошку. Перед завантаженням скетчу потрібно переконатися, що вибрано правильну плату та порти.

Також знадобиться USB-кабель для з'єднання плати та комп'ютера. Після виконання всіх вищенаведених дій потрібно натиснути кнопку «Завантажити» на панелі інструментів.

Останні версії плат Arduino можуть автоматично скидатися перед початком запису. На старих платах доводиться натискати на кнопку Reset, яка знаходиться на них. Після успішного запису можна помітити блимання світлодіодів Tx та Rx. Якщо завантаження не вдалося, у вікні помилки з'явиться повідомлення [29].

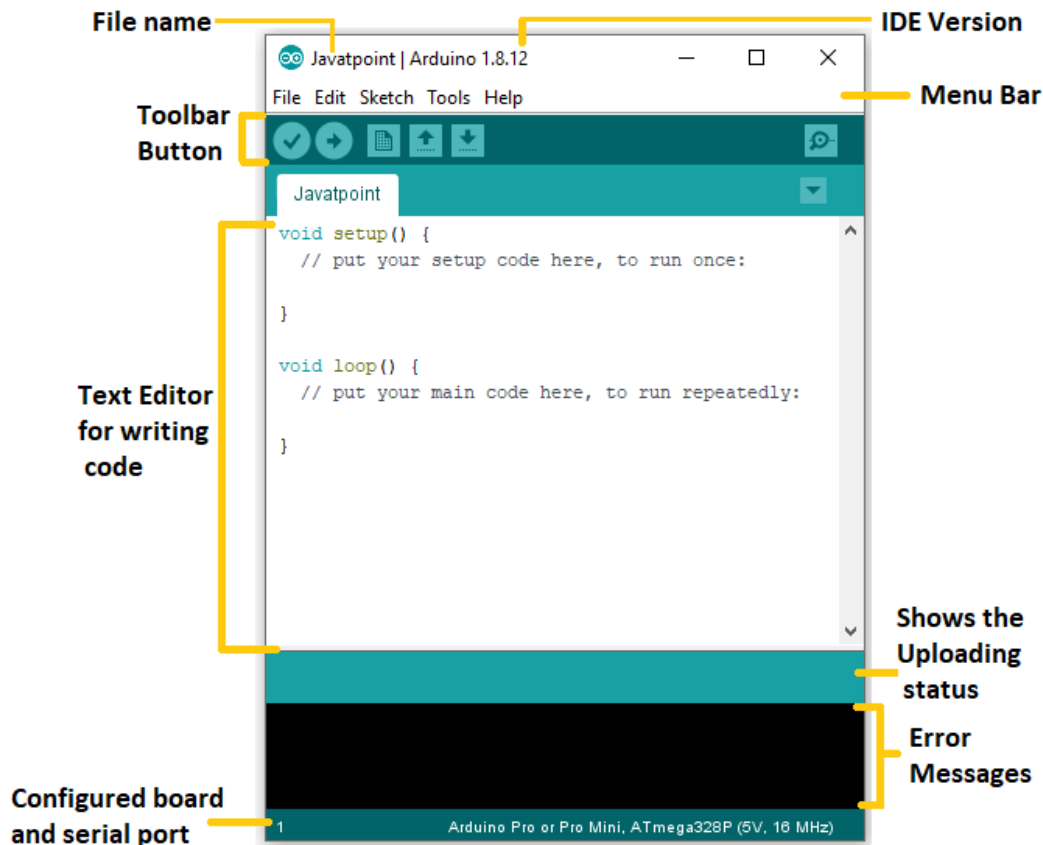


Рисунок 2.12 – Функціональні можливості Arduino IDE

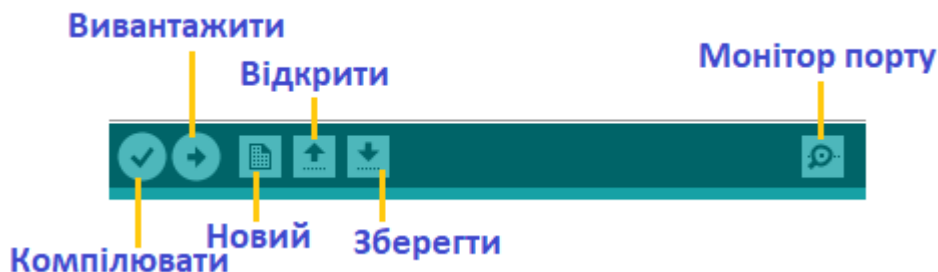


Рисунок 2.13 – Панель інструментів

Для завантаження скетчу за допомогою завантажувача Arduino не потрібне додаткове обладнання. Завантажувач визначається як невелика програма, що завантажується у мікроконтролер, встановлений на платі. Світлодіод блиматиме на PIN 13.

Кнопка «Відкрити» використовується для відкриття вже створеного файлу. Вибраний файл відкривається у поточному вікні.

Кнопка «Зберегти» використовується для збереження поточного ескізу або коду.

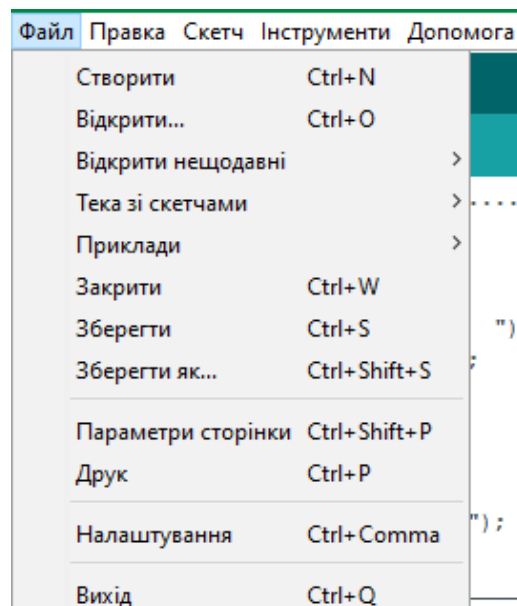
Кнопка «Новий» використовується для створення нового ескізу або відкриття нового вікна.

Кнопка «Компілювати» використовується для перевірки помилок під час компіляції скетчу або написаного коду.

Кнопка послідовного монітора розташована у правому куті панелі інструментів. Відкриє серійний монітор.

Коли запускається послідовний монітор, плата скидається у Windows, Linux та Mac OS X. Якщо є потреба обробити керуючі символи у скетчі, потрібно використовувати зовнішню термінальну програму. Термінальну програму слід підключити до СОМ-порту, призначеного для підключення плати до комп'ютера.

Якщо натиснути кнопку «Файл» у рядку меню з'явиться список, що розкривається, що показано на рисунку 2.14.



Рисунк 2.14 Відкрита вкладка «Файл»

Кнопка «Створити» відкриває нове вікно. Данна операція не видаляє ескіз, який існує. Кнопка «Відкрити» відкриває скетч, який можна переглянути з папок та драйверів комп'ютера.

Кнопка «Відкрити нещодавні» покаже нещодавно закриті скетчі.

Зберігає поточні ескізи, створені програмним забезпеченням Arduino IDE. Відкриває вибраний скетч або код нового редактора екземпляра. Наведено різні приклади невеликих проектів для кращого розуміння IDE та плати. IDE надає приклади із власної практики.

Кнопка «Закрити» закриває вікно, в якому було натиснуто кнопку.

Кнопка «Зберегти» використовується для збереження поточного скетчу. Данна операція також зберігає зміни, внесені до поточного скетчу. Якщо ім'я файлу не вказано, то з'явиться вікно «Зберегти як...», в якому можна вказати ім'я файлу.

Кнопка «Параметри сторінки» дозволяє настроїти поля сторінки, орієнтацію та розмір друку. Зовнішній вигляд «Параметри сторінки» відображено на рисунку 2.15.

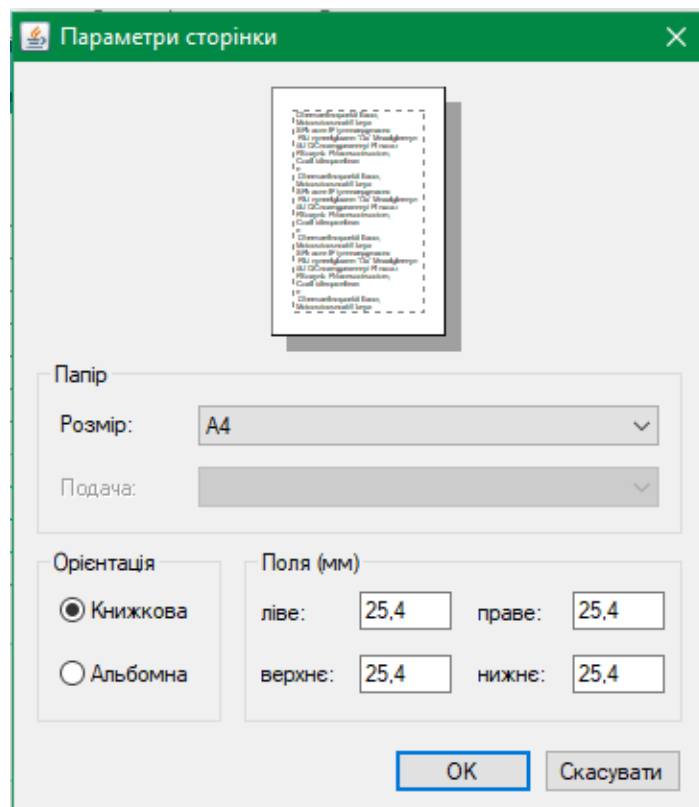


Рисунок 2.15 – Вікно «Параметри сторінки»

Підготовка поточного ескізу до друку відповідно до параметрів, зазначених у «Параметри сторінки».

Вкладка «Налаштування» дозволяє настроїти додаткові параметри Arduino IDE.

Кнопка «Вийти» використовується для закриття всіх вікон IDE. Той самий закритий скетч знову відкриється, коли повторно відкрити Arduino IDE.

Після натиснення кнопки «Змінити» у рядку меню, з'явиться випадаючий список, що показано на рисунку 2.16.

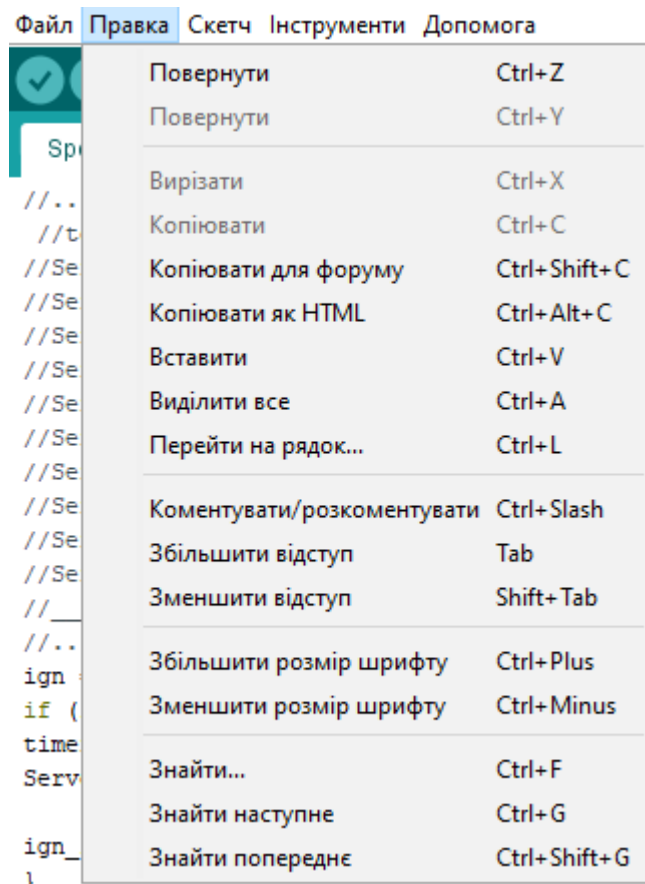


Рисунок 2.16 – Відкрита вкладка «Правка»

Кнопка "Назад" використовується для скасування останнього редагування ескізу, зробленого під час редагування.

Кнопка «Повторити» використовується для повторного редагування останнього ескізу під час редагування.

Дозволяє нам видалити виділений текст із написаного коду. Потім текст копіюється в буфер обміну. Також можна вставити цей текст у будь-якому місці даного скетчу.

Кнопка «Копіювати» копіює виділений текст. Потім текст поміщається в буфер обміну.

Кнопка «Повторити» використовується для повторення останньої зміни, внесеної до скетчу під час редагування.

Це дозволяє нам видалити виділений текст із написаного коду. Далі текст поміщається в буфер обміну. Також можна вставити цей текст будь-де в скетчі.

Кнопка «Копіювати для форуму» використовується для копіювання виділеного тексту в буфер обміну, який також підходить для розміщення на форумі.

Кнопка «Копіювати для форуму» використовується для копіювання виділеного тексту як HTML до буфера обміну. Бажано для вбудовування у веб-сторінки.

Кнопка «Вставити» використовується для вставлення виділеного тексту буфера обміну у вказане положення курсору.

Кнопка «Вибрати все» виділяє весь текст ескізу.

Кнопка «Перейти до лінії...» переміщує курсор на вказаний номер рядка.

Кнопка «Коментувати/розкоментувати» використовується для розміщення або видалення позначки коментаря (//) на початку вказаного рядка.

Кнопка «Збільшити відступ» використовується для додавання пробілу на початку вказаного рядка. Інтервал переміщує текст праворуч.

Кнопка «Зменшити відступ» використовується для віднімання або видалення пробілу на початку вказаного рядка. Інтервал переміщує текст ліворуч.

Кнопка «Збільшити розмір шрифту» збільшує розмір шрифту написаного тексту.

Кнопка «Зменшити розмір шрифту» зменшує розмір шрифту написаного тексту.

Кнопка «Знайти...» використовується для пошуку зазначеного тексту. Ми також можемо замінити текст. Він виділяє текст на ескізі, що вказано на рисунку 2.17.

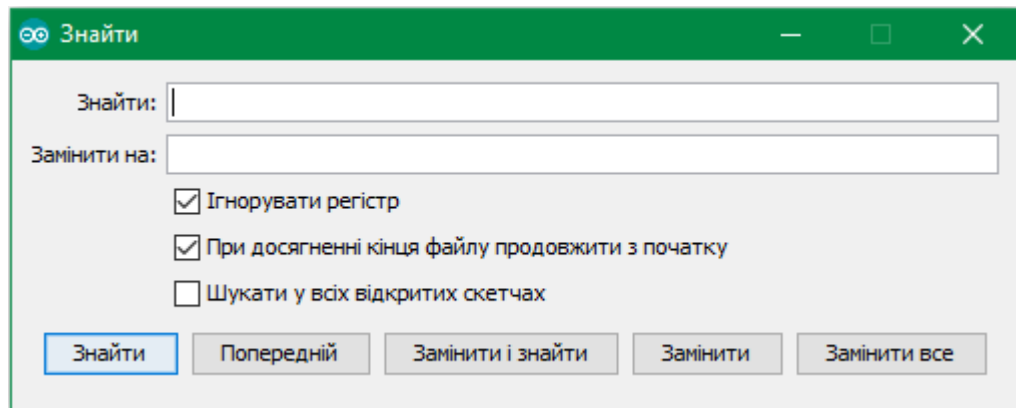


Рисунок 2.17 – Вікно «Знайти»

Кнопка «Знайти далі» виділяє наступне слово, яке вказано у вікні «Знайти...». Якщо такого слова немає, виділений текст не відобразиться.

Кнопка «Знайти попередній» виділяє попереднє слово, яке вказано у вікні «Знайти...». Якщо такого слова немає, виділений текст не відобразиться.

При натисканні кнопки «Скетч» на панелі меню, з'являється випадаючий список, який зображено на рисунку 2.18.

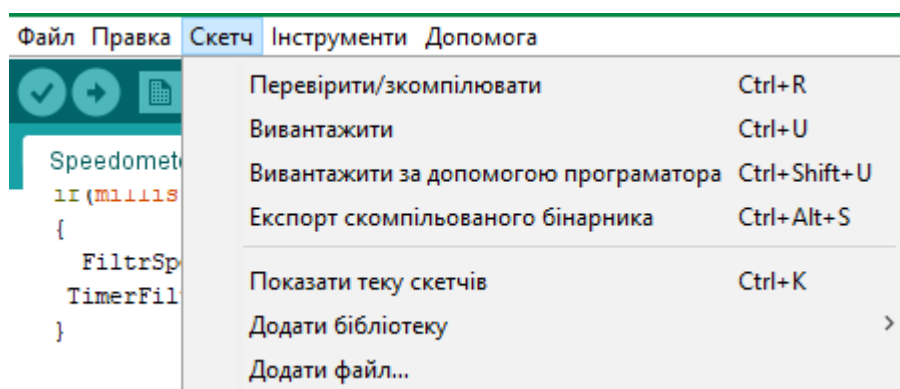


Рисунок 2.18 – Випадаючий список «Скетч» панелі меню

У даному списку є наступні варіанти:

- 1) «Перевірити/Компілювати» - перевірка коду на наявність помилок під час його компіляції;
- 2) «Завантажити» - проведення налаштування коду через порт на вказану плату;
- 3) «Завантажити за допомогою програматора» - зміна завантажувача, який встановлений на плату. За допомогою цього параметра стає можливим використання всієї флеш-пам'яті. Для реалізації цього потрібно відновити завантажувач за допомогою параметра «Інструменти»/«Записати завантажувач», для того щоб завантажити його через послідовний порт USB;
- 4) «Експортувати скомпільований двійковий файл» - використовується для збереження шістнадцяткового файлу (.hex) та дозволяє збереження в архіві. За допомогою інших інструментів файл формату .hex також можна надіслати на плату;
- 5) «Показати папку скетчу» - відкриття папки з поточним написаним або скетчовим кодом;
- 6) «Включити бібліотеку» - увімкнення різних бібліотек Arduino. Бібліотеки підключаються на початку коду за допомогою символу «#». Також є можливість імпорту бібліотеки з файлу .zip;
- 7) «Додати файл» - додавання створеного файлу у нову вкладку наявного файлу. Для прикладу на рисунку 2.19 зображено доданий файл «Blink» до файлу «Speedometer_NEV_8».



Рисунок 2.19 – Файл «Blink», доданий до файлу «Speedometer_NEV_8»

При натисканні кнопки «Інструменти» на панелі меню, з'являється випадаючий список, який зображено на рисунку 2.20.

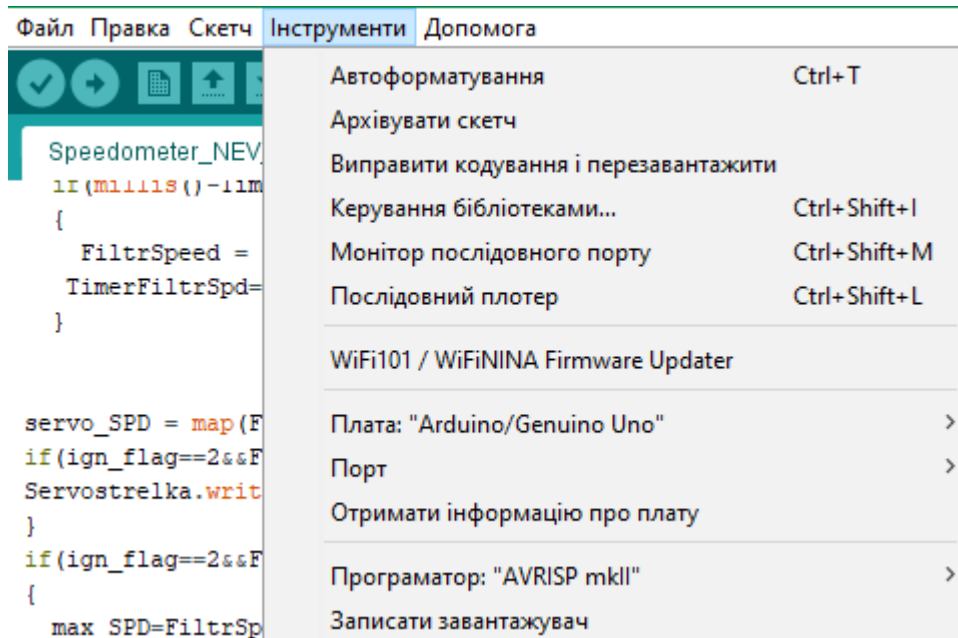


Рисунок 2.20 - Випадаючий список «Інструменти» панелі меню

Після натискання на кнопку «Інструменти» стануть доступними такі варіанти:

- 1) «Автоформат» - форматування написаного коду. Прикладом може бути вирівнювання відкритих та закритих фігурних дужок в кодї;
- 2) «Архівувати скетч» - архівування копії поточного скетчу або коду у файл формату .zip. Каталог архіву залишається таким самим, як і скетч;
- 3) «Виправити кодування і перезавантажити» - виправлення невідповідностей між різними картами символів операційної системи та кодування карт символів редактора;
- 4) «Керування бібліотеками» - виведення оновленого списку усіх попередньо встановлених бібліотек. Також цю опцію можна використовувати для встановлення нової бібліотеки у Arduino IDE;
- 5) «Монітор послідовного порту» - дає змогу обмінюватись даними з підключеною платою;
- 6) «Послідовний плотер» - відображення послідовних даних на графіку;
- 7) «Програма оновлення мікропрограми WiFi101/WiFiNINA» - перевірка та оновлення підключеної до плати мікропрограми WiFi;

- 8) «Плата» - вибір плат з списку. Вибрана плата має бути принаймні схожою на плату, яка підключена до комп'ютера;
- 9) «Процесор» - відображення процесору відповідно до обраної плати. Оновлення відбувається щоразу при виборі плати;
- 10) «Порт» - відображення віртуальних та реальних послідовних пристроїв, які присутні на машині;
- 11) «Отримати інформацію про плату» - вивід інформації про обрану плату. Спершу потрібно вибрати відповідний порт;
- 12) «Програматор» - вибір апаратного програматора під час програмування плати. Це потрібно, якщо не використовувати вбудований послідовний USB порт. Також це може знадобитися при записі завантажувача;
- 13) «Записати завантажувач» - корисна опція, якщо мікроконтролер був придбаний без завантажувача. Перед його записом слід переконатися у правильності вибору плати.

2.3 Висновки

В даному розділі було проведено аналіз та вибір елементної бази для лічильника миттєвої швидкості руху, а саме аналіз функціонування пристрою, було вибрано тип мікроконтролера на якому буде побудований пристрій. Також було ознайомлено з програмним забезпеченням, а саме Arduino IDE та принципом роботи датчика швидкості на основі ефекту Холла.

Для роботи з мікроконтролерами типу ATMEGA 328P використовують програмне середовище для розробки Arduino IDE.

Arduino IDE – це програмне забезпечення з відкритим вихідним кодом, яке використовується для написання та завантаження коду на плати Arduino. IDE підходить для різних операційних систем, таких як Windows, Mac OS X і Linux. Він підтримує мови програмування C та C++. IDE тут означає інтегроване середовище розробки.

3 ПОБУДОВА ПРИСТРОЮ ДЛЯ ВИМІРЮВАННЯ МИТТЄВОЇ ШВИДКОСТІ РУХУ

3.1 Проектування пристрою для вимірювання миттєвої швидкості руху

Основною задачею при виборі елементів бази був вибір доступних та недорогих компонентів, для оперативної заміни у випадку поломки пристрою. Так було вибрано за мікроконтролер ATMEGA 328P, через його надійність та доступність, для управління було вибрано модуль енкодера EC11, через його популярність та наявність об'язки, для виводу цифрової інформації було вибрано OLED дисплей 0.96" I2C 128x64, через його компактність наявність оптимізованих бібліотек, а для виводу інформації про швидкість на шкалу було вибрано сервопривід Tower Pro MG90S, через його поширеність та малошумність.

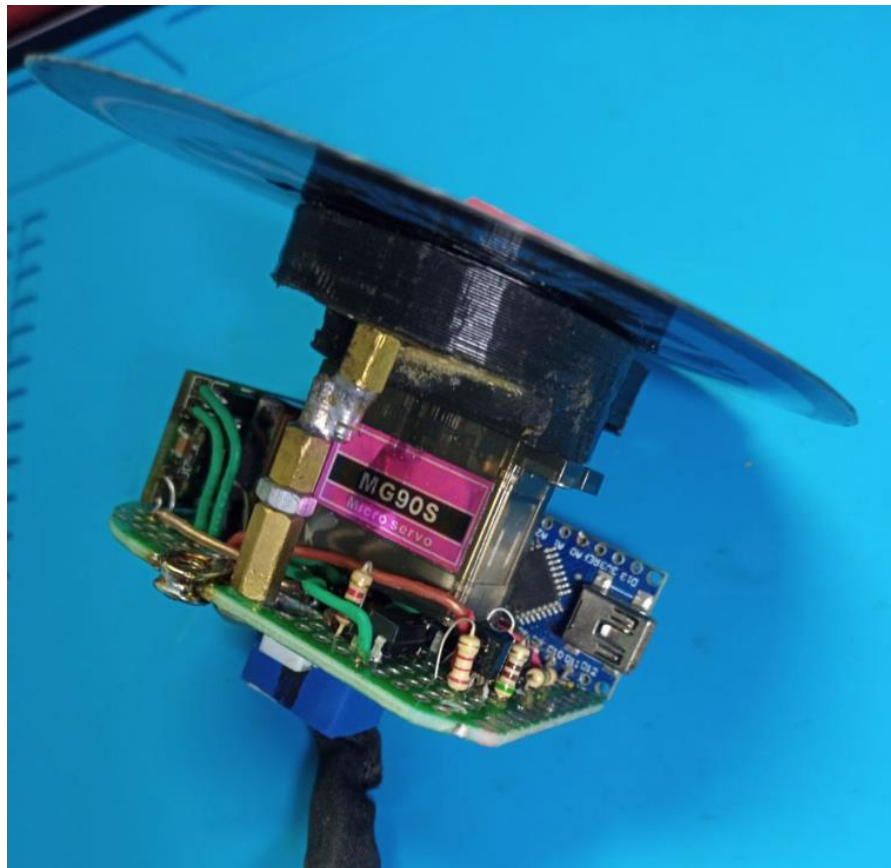


Рисунок 3.1 – Компонування елементів всередині корпусу

Плата Arduino керує всім процесом функціонування пристрою: зчитування імпульсу із транзистора оптопар, до якого підключений модуль датчика швидкості, обчислення частоти обертання (в кілометрах на годину) і передача значень на OLED дисплей з дублюванням на сервопривід. Також має бути два лічильника швидкості, один добовий, інший загальний. Добовий має обнулятися при кожному відключенні живлення, а загальний зберігати нове значення у EEPROM.

Для збирання даного пристрою потрібно знати правила техніки безпеки та володіти базовими навичками пайки. Пайка означає з'єднання металевих компонентів разом з розплавленим металом, який називається припоєм, який має нижчу температуру плавлення, ніж інші метали. Це важливий процес в електронній промисловості і є основним методом з'єднання електричних компонентів. Крім виготовлення друкованих плат (друкованих плат), пайка використовується в покрівлі, виготовленні ювелірних виробів, з'єднанні труб і сантехніці.

Для пайки використовується паяльник або пістолет з температурою менше 350°C і припій. Зазвичай припій виглядає як тонкий дріт або трубка. У середині трубки знаходиться кислотна суміш, яка називається флюсом, яка запобігає окисленню.

Хоча існують різні типи припою, зазвичай це металевий сплав свинцю або олова в поєднанні з латунню або сріблом, який має низьку температуру плавлення. Оскільки паяльник плавить цей метал, він потім використовується як клей, щоб склеїти шматки разом. Коли метал припою охолоне, він знову затвердіє в одну велику форму, яка з'єднує дві частини. В інтересах скорочення використання свинцю через проблеми з навколишнім середовищем та безпекою, зараз є деякі версії припою без свинцю. Ці альтернативи часто включають латунь, мідь, олово або срібло. Безсвинцевий припій має вищу температуру плавлення і може бути менш ефективним, ніж традиційний припій.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

Пайка друкованих плат - це ще один термін для процесу пайки електричних плат . Цей тип пайки є однією з основних технік, яку повинен освоїти кожен, хто бажає працювати з електронікою та електричними схемами. Хоча існує багато різних способів завершити процес пайки, найосновніше пояснення процесу пайки полягає в тому, що це спосіб з'єднання двох маленьких частин на поверхні друкованої плати, що означає «друкована плата». Іншими словами, пайка - це спосіб з'єднання двох або більше різних електричних компонентів на вашій платі.

В основі процесу пайки сама дія досить проста. Все, що вам потрібно для виконання найпростішої роботи паяння, — це паяльник, трохи припою та матеріали, які ви спаяєте. Паяльник трохи нагадує ручку і є інструментом, який дуже сильно нагрівається, розплавляючи припій і з'єднуючи його дві частини.

Хоча існують різні типи припою, зазвичай це металевий сплав свинцю або олова в поєднанні з латунню або сріблом, який має низьку температуру плавлення. Оскільки паяльник плавить цей метал, він потім використовується як клей, щоб склеїти шматки разом. Коли метал припою охолоне, він знову затвердіє в одну велику форму, яка з'єднує дві частини.

3.2 Програмування пристрою визначення швидкості руху

Внутрішній тактовий генератор частотою 16МГц допомагає працювати мікроконтролеру. Завдання полягає у тому, щоб створити програму для мікроконтролера, яка буде забезпечувати зчитування імпульсів постійно, обробляти та виводити на екран з дублюванням швидкості на сервопривід, а також підраховувати загальну та добову подолану відстань, а при відключенні живлення записувати загальну відстань в EEPROM, а добову скидати до нуля.

Датчик Холла використовується для виявлення магніта, закріпленого на карданному валі. Коли біля датчика Холла проходить магніт, він подає логічну

одиницю на плату Arduino, яка в свою чергу буде проводити наступні обчислення за формулою 3.1.

$$SPD = \text{impuls} * (2 * 3,14 * r) / 1000. \quad (3.1)$$

Де SPD – вирахована величина, готова для виведення на екран; *impuls* вхідні імпульси, прийняті з датчика; *r* – радіус.

Але після проведених тестів було виявлено, що такий тип обробки даних піддається шумам, що дається взнаки під час експлуатації. Саме тому постало питання як фільтрувати значення, які приходять від датчика. Щоб ефективно підтримувати та аналізувати велику кількість даних датчика в реальному часі, часто використовують техніку фільтрації, яка добре відображає характеристики вихідних даних.

Методом проб і помилок було вибрано самий оптимальний тип фільтрації даних:

```
int GetFiltrSpeed(int spd,int filtSpeed)
{if(spd>0&&spd<160&&spd>filtSpeed)
{ filtSpeed++;}
else if(spd>=0&&spd<160&&spd<filtSpeed)
{ filtSpeed--;}
return filtSpeed;}

```

Цей тип фільтрації даних зарекомендував себе непогано, і навіть при штучно наведених шумах продемонстрував завадостійкість на достатньому рівні.

3.3 Блок-схеми роботи пристрою

На рисунку 3.2 зображено загальну блок-схему роботи пристрою, а на рисунку 3.3 зображено блок-схему екранів OLED дисплею.

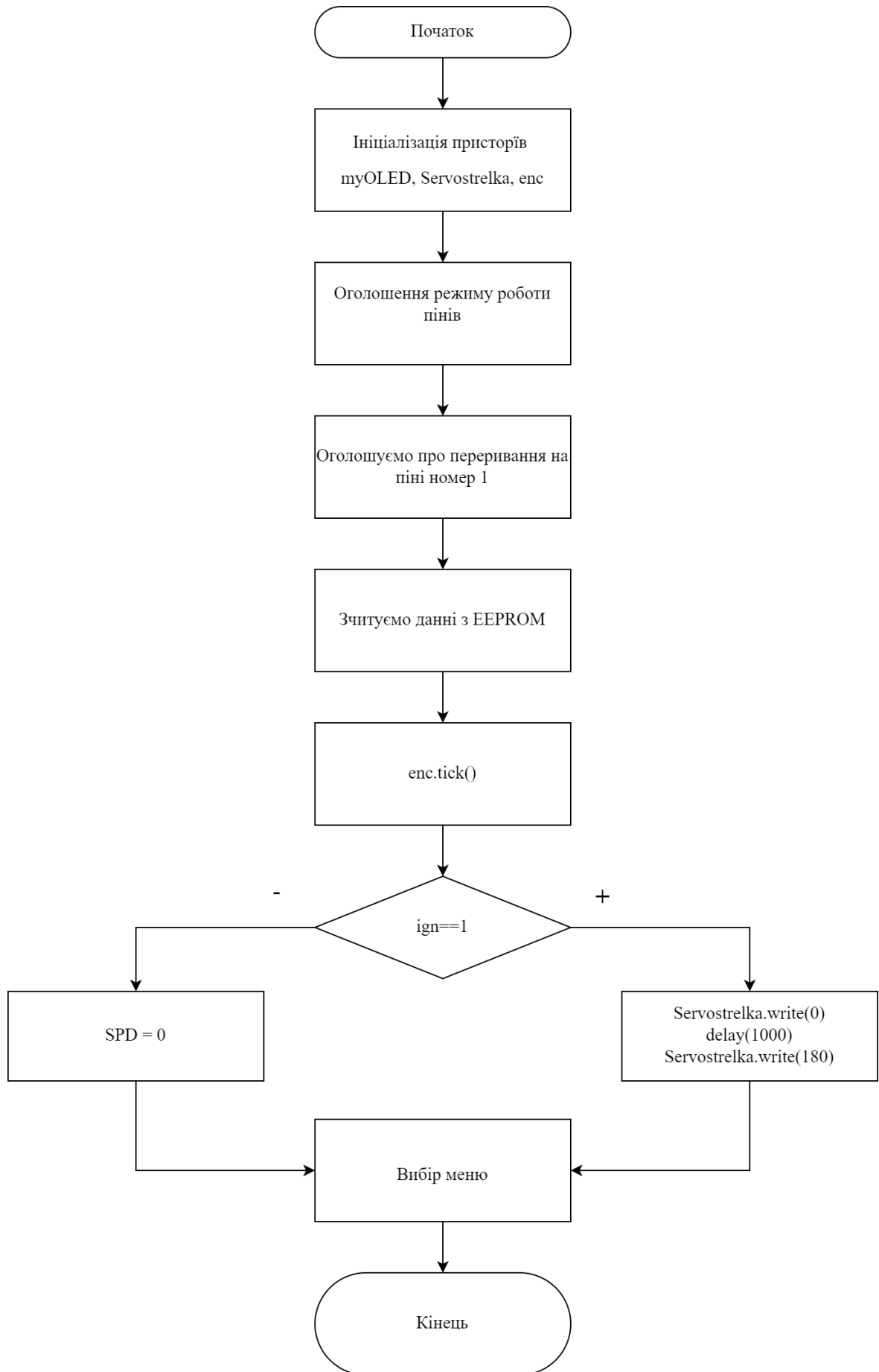


Рисунок 3.2 – Загальна блок-схема роботи пристрою

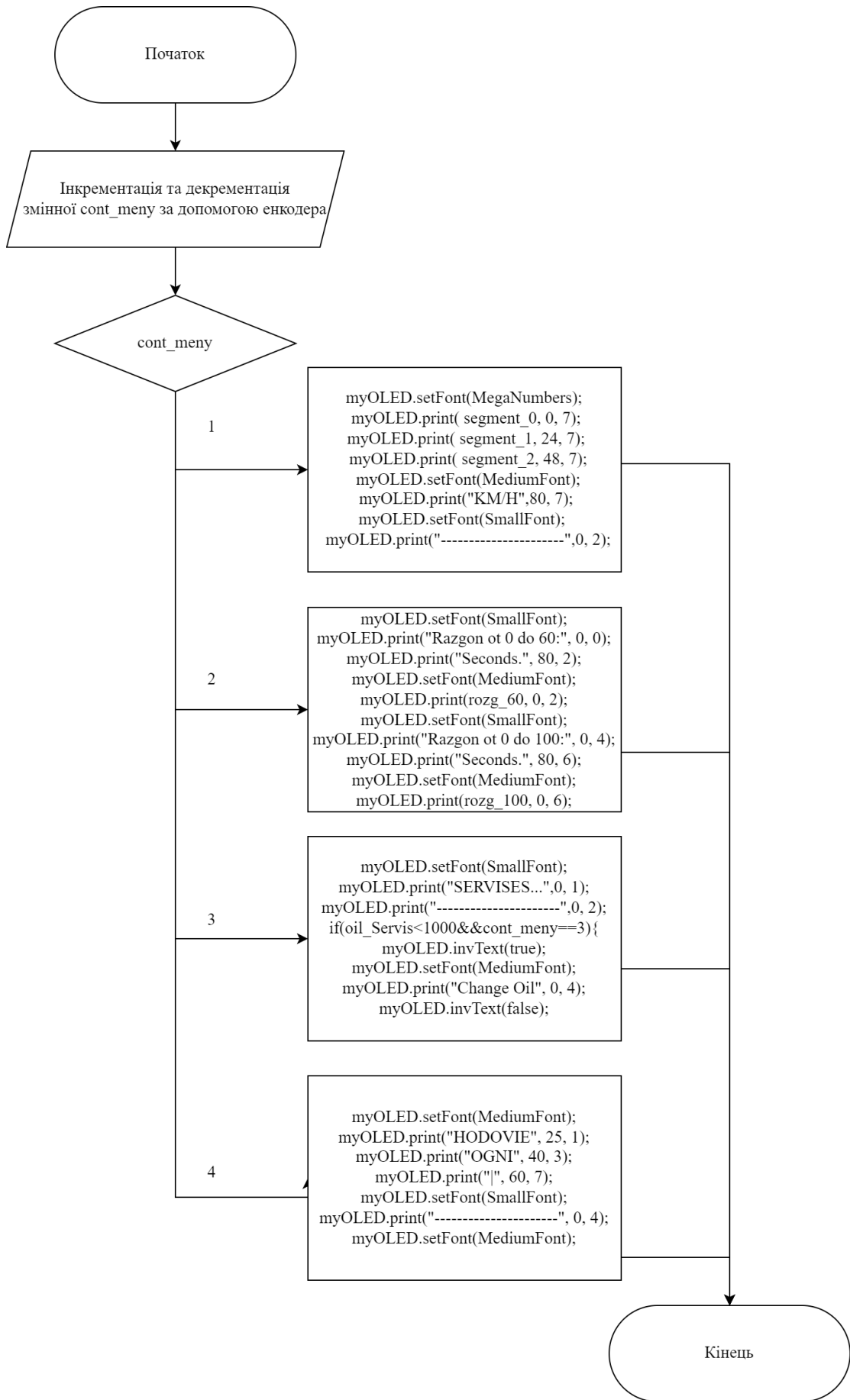


Рисунок 3.3 – Блок-схема перемикання екранів пристрою

На рисунку 3.4 наведено структурну схему пристрою, а на рисунку 3.5 зображено схему підключення пристрою, яка була реалізована за допомогою онлайн-інструменту для проектування електронних схем circuito.

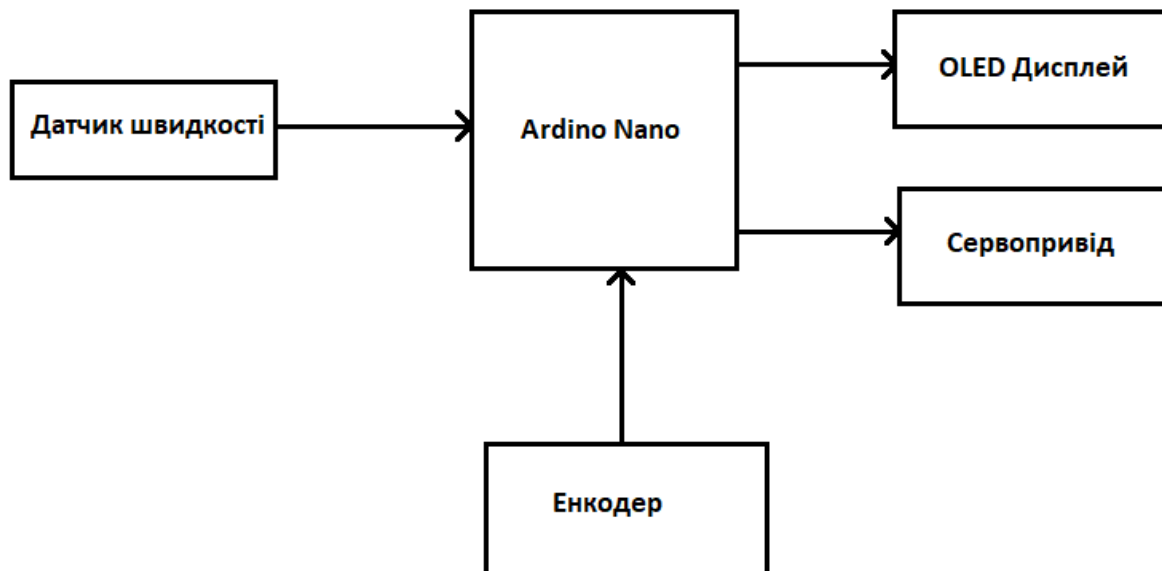


Рисунок 3.4 – Структурна схема пристрою

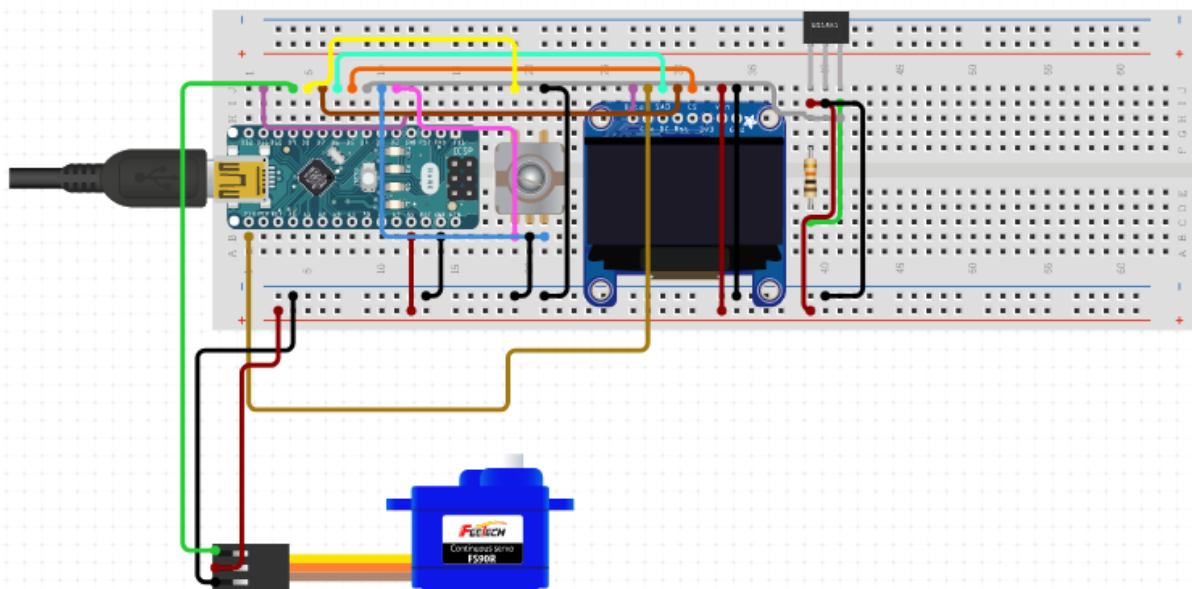


Рисунок 3.5 – Схема підключення пристрою

Зм.	Арк.	№ докум.	Підпис	Дата

3.4 Реалізація пристрою

В результаті було спроектовано бюджетний та компактний програмно-технічний пристрій для вимірювання миттєвої швидкості руху з добовим та загальним лічильником відстані з можливістю встановлення у корпус з під стандартного спідометра з незначним втручанням в конструкцію корпусу спідометра.

На рисунках 3.4 та 3.5 зображено зовнішній вигляд спроектованого пристрою.



Рисунок 3.4 – Робота самого пристрою з індикацією загального пробігу

Продовження таблиці 3.1

Понижаючий модуль Mini 360	20
Датчик Холла	33
Конденсатори, резистори, діоди та транзистори	30
Витратні матеріали	15
Загалом	525

Отже, було проведено калькуляцію затрат на розробку пристрою для визначення миттєвої швидкості руху. Після здійсненого розрахунку стало зрозуміло, що витрати на комплектуючі склали 525 грн, і це означає, що даний пристрій набагато дешевший за найвідоміші аналоги. Цей пристрій може встановлюватися з певними модифікаціями в будь-які марки авто.

3.6 Висновки

В даному розділі було проведено проектування пристрою, визначення його основних цілей за допомогою блок-схем, побудовано структурну та загальну схему роботи пристрою, а також реалізовано повністю готовий до роботи пристрій.

Для збирання даного пристрою потрібно знати правила техніки безпеки та володіти базовими навичками пайки, з якими ознайомлено в даному розділі. Пайка означає з'єднання металевих компонентів разом з розплавленим металом, який називається припоєм, який має нижчу температуру плавлення, ніж інші метали. В ході тестування було виявлено низьку завадостійкість до шумів. Для вирішення цієї проблеми було розроблено фільтр значень, який відкидає невалідні дані. Після вищенаведених покращень у пристрою підвищилася завадостійкість, а також збільшилася надійність, що відобразилося на продуктивності.

ВИСНОВКИ

В ході наукової роботи було проведено аналіз та вибір елементної бази для лічильника миттєвої швидкості руху, а саме аналіз функціонування пристрою, було вибрано тип мікроконтролера на якому буде побудований пристрій. Також було ознайомлено з програмним забезпеченням, а саме Arduino IDE та принципом роботи датчика швидкості на основі ефекту Холла.

На даний момент є багато морально застарілих автомобілів, які все ще активно використовуються, у яких стандартний прилад виміру швидкості вийшов з ладу, а нових приладів немає в наявності, або покупка останніх є нерентабельною.

Тому виникає завдання розробки бюджетного та компактного програмно-технічного пристрою для вимірювання миттєвої швидкості руху з добовим та загальним лічильником відстані з можливістю встановлення у корпус з під стандартного спідометра з незначним втручанням в конструкцію корпусу спідометра. Потрібно отримати імпульси з датчика через оптопару, вирахувати за формулою і вивести на OLED дисплей з дублюванням швидкості на сервопривід.

Для роботи з мікроконтролерами типу ATMEGA 328P використовують програмне середовище для розробки Arduino IDE.

Arduino IDE – це програмне забезпечення з відкритим вихідним кодом, яке використовується для написання та завантаження коду на плати Arduino. IDE підходить для різних операційних систем, таких як Windows, Mac OS X і Linux. Він підтримує мови програмування C та C++. IDE тут означає інтегроване середовище розробки.

Також було проведено проектування пристрою, визначення його основних цілей за допомогою блок-схем, побудовано структурну та загальну схему роботи пристрою, а також реалізовано повністю готовий до роботи пристрій.

Для збирання даного пристрою потрібно знати правила техніки безпеки та володіти базовими навичками пайки, з якими ознайомлено в даному

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

розділі. Пайка означає з'єднання металевих компонентів разом з розплавленим металом, який називається припоєм, який має нижчу температуру плавлення, ніж інші метали.

В ході тестування було виявлено низьку завадостійкість до шумів. Для вирішення цієї проблеми було розроблено фільтр значень, який відкидає невалідні дані. Після вищенаведених покращень у пистрою підвищилася завадостійкість, а також збільшилася надійність, що позитивно відобразилося на продуктивності.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Бабич О.О. Визначення безконтактних методів вимірювання температури нагрітих тіл. *Системи озброєння і військова техніка*. 2011. № 1. С. 69-71.
2. Александрова Н.М. Методи вимірювання температури нагрітих тіл. *Системи озброєння і військова техніка*. 2014. № 3. С. 81-89.
3. Назаренко Л.А. Фізичні основи джерел світла: навч. посіб. Харків: ХНАМГ, 2019. 206 с.
4. Брао І. Аналіз проблематики та перспективних напрямів розвитку безконтактної термометрії. *Вимірювальна техніка та метрологія*. 2014. Вип. 75. С. 40-44.
5. Назаренко Л.А. Фізичні основи джерел світла: навч. посіб. Харків: ХНАМГ, 2016. 209 с.
6. Скорик Б.И. К вопросу применения бесконтактных методов измерения температуры нагретых. *Системи оброб. інформації*. 2017. Вип. 9. С. 129-132.
7. Андреев А.Н. Оптические измерения. А.Н. Андреев, Е.В. Гаврилов, Г.Г. Ишанин и др.: Учеб. пособие. – М.: Университетская книга: Логос, 2012. 416 с.
8. Ramapatruni S., Narayanan S. N. Anomaly detection models for smart home security, *Proceedings of 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*, IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2019. P. 19-24.
9. Yamauchi, M. Anomaly Detection in Smart Home Operation From User Behaviors and Home Conditions, 2020, Vol. 66, P. 183–192.
10. Molle M., Sohraby K., Venetsanopoulos A., Space-Time Models of Asynchronous CSMA Protocols for Local Area Networks, *IEEE Journal on Selected Areas in Communications*, Vol.: 5 Iss. 6, 1987 pp. 956-96
11. Дембовский В.В. Технологические измерения и приборы в металлургии: учебное пособие. СПб.: СЗТУ, 2004. 70 с.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

12. Поліщук В.М. Фотометрія: конспект лекцій. В.М. Поліщук. Харків: ХНАМГ, 2014. 147 с.
13. Поліщук В.М. Фотометрія: конспект лекцій. Харків: ХНАМГ, 2012. 147 с.
14. Фрунзе А.В. Пирометры спектрального отношения: преимущества, недостатки и пути их устранения. *Фотоника*, 2009. № 4 С.32.
15. Протасов А.Г. Многоканальный стенд для контроля температуры. Вісник НТУУ “КПІ”. *Серія ПРИЛАДОБУДУВАННЯ*, 2015. № 49(1) . С. 61-68.
16. Protasov A. Application of FEMLAB Software for Simulation of the Thermal Method for Nondestructive Testing. American Society for Engineering Education. Annual conference and Exposition. . June 14-17 2014. Austin, Texas, USA, pp. 182-191.
17. Муравьёв А.В. Основные тенденции, проблемы и перспективы развития дисплейной наноэлектроники. Неруйнівний контроль в контексті асоційованого членства України в Європейському союзі: матеріали 2-гої науково-технічної конференції з міжнародною участю. – Польща, Люблін, 2018. С. 10-11.
18. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. СПб. БХВ-Петербург. 2016. 320 с.
19. J. Lee, V. Bagheri, H.A. Kao, «A cyber-physical systems architecture for industry 4.0-based manufacturing systems». *Manufacturing Letters*. 2015. Vol. 3, pp. 18-23,
20. Li B. S. X., Wan B., Wang C., Zhou X., Chen X. Definitions of predictability for cyber physical systems. *J. of Systems Architecture*. 2016. DOI: 10.1016/j.sysarc.2016.01.007.
21. Arduino Laser Infrared Thermometer URL: <https://www.instructables.com/Arduino-Laser-Infrared-Thermometer/> (10.05.2021).
22. Галаган Р.М., Муравьёв А.В., Томашук А.С., Модель восстановления серии изображений из смазанного изображения для решения задачи высокоточного измерения диаметра и температуры излучающих объектов. Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій (матеріали IV Міжнародної науково-технічної конференції). 2019. С. 169-171.

23. Автоматика для запобігання вибухам і пожежам: посіб. О.А. Дерев'янка та інш. Харків. 2016. 279 с.

24. Поскачей А.А., Чубаров Е.П., Энергоатомиздат. *Опτικο-електронные системы измерения температуры*. 2013. 246 с.

					КВРКІ 180102.18.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

Додаток А (обов'язково)

Копія креслення “Схема апаратних з'єднань системи”

Схема підключення пристрою

Структурна схема пристрою

```

graph TD
    A[Датчик швидкості] --> B[Arduino Nano]
    B --> C[OLED Дісплей]
    B --> D[Сервопривід]
    E[Енкодер] --> B
    
```

КвКЛ. 1801.02.18.01.02 E8		Листок	Матр	Масштаб
Зм. / Зміст	№ докум.	Датум		
Друк	Виконав	Перевірив		
Д.Кочур	В.Кочур	А.Кочур	Листок 1	Листок 1
Д.Кочур	В.Кочур	А.Кочур		
Д.Кочур	В.Кочур	А.Кочур		
Д.Кочур	В.Кочур	А.Кочур		
ХТУ, зр.КЛ-18-01				

КвКЛ. 1801.02.18.01.02 E8

Додаток Г

Лістинг коду

```
#define CLK 6
#define DT 5
#define SW 4
#define PRECISE_ALGORITHM
#include "GyverEncoder.h"
#include <EEPROM.h>
#include <iarduino_OLED_txt.h> // Підключаємо бібліотеку iarduino_OLED_txt.
iarduino_OLED_txt myOLED(0x3C);
extern uint8_t MediumFont[];
extern uint8_t SmallFont[];
extern uint8_t MegaNumbers[]; // Підключаємо шрифт
#include <Servo.h>
Servo Servostrelka;
Encoder enc(CLK, DT, SW, TYPE1); // для роботи з кнопкою одразу вибираємо тип
volatile float stala=0.4530;
volatile float timer_SPD,timer1;
volatile int SPD;
volatile long odo_counter=1,odo1_counter=1,serv_counter=1;
unsigned long timer_strelka;
float ServEE, OdoEE;
float odometer,odometer1=25.0,servis,odoTemp;
float svitlo_Timer=0;
float TimerFiltrSpd=0;
float timer_Rozg=0,rozg_60=100,rozg_100=100, max_Rozg60=100,max_Rozg100=100;
float oil_Temp=0,oil_Servis=0;
float timerVccOLED=0;
int svitlo_flag,fara_State=0;
int servo_SPD=0;
int FiltrSpeed=0;
int rozg_Flag=0;
int timer_save=2000;
int in_SPD=3;
int ign_pin=9,ign_flag;
int save_pin=8;
int saveFlag=1;
bool ign,save;
int SPD_temp = 0,max_SPD;
int cont_meny=1,clcMeny=0,holMeny=0,presMeny=0,douMeny=0;
int segment_0=0,segment_1=0,segment_2=0,SPD_in=0;
int fara_out=11;
int dispVcc=10,flagOLEDVcc=3;
void setup() {
//Serial.begin(9600);
pinMode(save_pin,INPUT_PULLUP);//8-й вхід збереження пробіга при вимкнені живлення
коли HIGH записуєм данні в еепром
pinMode(ign_pin,INPUT_PULLUP);//pin9 вход включение зажигания
```

```

pinMode(in_SPD,INPUT_PULLUP);//pin3 вхід спідометра
pinMode(fara_out,OUTPUT);//pin 11 виход включення днівних ходових огнів
pinMode(disVcc,OUTPUT);//pin 10 виход живлення OLED дисплея
attachInterrupt(1, RAPID, RISING); //объявляємо приривання
Servostrelka.attach(7); //підключаємо до 7 піна сервомотор
enc.setType(TYPE1);
enc.setPinMode(HIGH_PULL);

EEPROM.get(0,odoTemp);
EEPROM.get(0,OdoEE);
EEPROM.get(5,max_SPD);
EEPROM.get(10,max_Rozg60);
EEPROM.get(15,max_Rozg100);
EEPROM.get(20,oil_Temp);
EEPROM.get(20,ServEE);
EEPROM.get(25,svitlo_flag);
}
void loop() {
enc.tick();
ign = digitalRead(ign_pin);//обробка включення зажигання
if (ign == 0&& ign_flag==0){
timer_strelka=millis();
Servostrelka.write(0);

ign_flag=1;
}
if(ign == 0&& ign_flag==1&&millis()-timer_strelka>1000){
Servostrelka.write(180);
ign_flag=2;
}
if(ign==1){
ign_flag=0;
timer_SPD=0;
Servostrelka.write(180);
}
if(ign_flag==2)
{ timerVccOLED=millis();}

if(flagOLEDVcc==3)
{
digitalWrite(disVcc,HIGH);
if(millis(>1000&&flagOLEDVcc==3) // перше включення після подачі напруги
{
myOLED.begin();
flagOLEDVcc=1;
}
}
}
if(ign_flag==2&&flagOLEDVcc==0)
{
myOLED._sendCommand(0xAF);
flagOLEDVcc = 1;
}

```

```

}
if(ign==1&&flagOLEDVcc==1&&(millis()-timerVccOLED>300000))
{
myOLED._sendCommand(0xAE);           // дисплей гасне
flagOLEDVcc = 0;
}
if(micros()-timer1>1000000.0)
{timer_SPD=0;
SPD=0;
}
SPD=((stala/(timer_SPD/1000000))*3.60);

if(millis()-TimerFiltrSpd>50.0)
{
FiltrSpeed = GetFiltrSpeed(SPD,FiltrSpeed);
TimerFiltrSpd=millis();
}
servo_SPD = map(FiltrSpeed, 10,120, 180,0);
if(ign_flag==2&&FiltrSpeed>0&&FiltrSpeed<120){
Servostrelka.write(servo_SPD);
}
if(ign_flag==2&&FiltrSpeed>max_SPD)
{
max_SPD=FiltrSpeed;
}
///розбиваємо швидкість на сегменти
segment_0=FiltrSpeed/100;
segment_1=(FiltrSpeed%100)/10;
segment_2=FiltrSpeed%10;
odometer = odoTemp+(stala*odo_counter)/1000; //вираховуєм загальний пробіг
odometer1 = (stala*odo1_counter)/1000; // вираховуєм суточний пробіг
save = digitalRead(save_pin); // зберігаєм загальний пробіг при вимкненні живлення
if(save==0&&millis())>10000&&saveFlag==1)
{
EEPROM.put(5,max_SPD);
if(odometer>OdoEE){ EEPROM.put(0,odometer);}
if(oil_Servis<ServEE){ EEPROM.put(20,oil_Servis);}
saveFlag=2;
}
if(FiltrSpeed==0&&ign_flag==2)
{
rozg_Flag=0;
timer_Rozg=millis();
}
if(FiltrSpeed>=60&&rozg_Flag==0&&ign_flag==2)
{
rozg_60=(millis()-timer_Rozg)/1000;
if(rozg_60<max_Rozg60)
{
max_Rozg60=rozg_60;
EEPROM.put(10,rozg_60);
}
}

```

```

}
rozg_Flag=1;
}
if(FiltrSpeed>=100&&rozg_Flag==1&&ign_flag==2)
{
rozg_100=(millis()-timer_Rozg)/1000;
if(rozg_100<max_Rozg100)
{
max_Rozg100=rozg_100;
EEPROM.put(15,rozg_100);
}
rozg_Flag=2;
}
oil_Servis=oil_Temp-((stala*serv_counter)/1000);//вираховуємо загальний пробіг

if(oil_Servis<1000&&cont_meny==1){
myOLED.invText(true);
myOLED.setFont(MediumFont);// Вказуємо шрифт, потрібний для виведення цифр і
тексту.
myOLED.print("OIL!", 75, 4);// Виводимо текст починаючи з 0 стовпця
myOLED.invText(false);
}
if(ign_flag==2){svitlo_Timer=millis();}

if(svitlo_flag==1&&ign_flag==2&&FiltrSpeed>10&&fara_State==0)
{
digitalWrite(fara_out,HIGH);
fara_State=1;
}
if(svitlo_flag==1&&ign_flag==0&&fara_State==1&&millis()-svitlo_Timer>25000)
{
digitalWrite(fara_out,LOW);
fara_State=0;
}

if(svitlo_flag==0)
{
digitalWrite(fara_out,LOW);
}
if (enc.isRight()) // якщо був поворот направо, збільшуємо на 1
{
myOLED.clrScr();
cont_meny--;
}

if (enc.isLeft()) // якщо був поворот наліво, зменшуємо на 1
{
myOLED.clrScr();
cont_meny++; }

if(enc.isSingle()) //одинарний клік

```

```

{
  myOLED.clrScr();
  presMeny=1;
  clcMeny++;
  if(clcMeny>1)clcMeny=0;
}
else
{
  presMeny=0;
}
if(enc.isDouble())//подвійний клік
{
  myOLED.clrScr();
  if(cont_meny>9)
  {
    cont_meny/=10;
  }
  else
  {
    cont_meny*=10;
  }
  douMeny++;
  if(douMeny>1)holMeny=0;
}

if(enc.isHolded())//довге утримування
{
  myOLED.clrScr();
  holMeny=1;
}
else
{
  holMeny=0;
}
if(cont_meny==0)cont_meny=4;
if(cont_meny==5)cont_meny=1;
if(cont_meny==9)cont_meny=11;
if(cont_meny==12)cont_meny=10;
if(cont_meny==22)cont_meny=20;
if(cont_meny==19)cont_meny=21;
if(cont_meny==29)cont_meny=30;
if(cont_meny==31)cont_meny=30;
if(cont_meny==40)cont_meny=4;
switch(cont_meny){

```

case 1://тут виводимо швидкість і одометер, можна при натисненні на кнопку ще суточний і загальний одометер

```

myOLED.setFont(MegaNumbers);
myOLED.print( segment_0, 0, 7);
myOLED.print( segment_1, 24, 7);
myOLED.print( segment_2, 48, 7);

```

```

    myOLED.setFont(MediumFont);
    myOLED.print("KM/H",80, 7);
    myOLED.setFont(SmallFont);
    myOLED.print("-----",0, 2);

    if(clicMeny==0){
        myOLED.setFont(MediumFont);    //Вказуємо шрифт, потрібний для виведення цифр і
текста.
        myOLED.print(odometer, 0, 1);
    }
    else{
        myOLED.setFont(SmallFont);    //Вказуємо шрифт, потрібний для виведення цифр і
тексту.
        myOLED.print("Odo1:", 0, 1);
        myOLED.setFont(MediumFont);
        myOLED.print(odometer1, 35, 1);
    }
    break;
    case 10://ТУТ ВИВОДИМО МАХ ШВИДКІСТЬ
        myOLED.setFont(SmallFont);
        myOLED.print("-----",0, 2);
        myOLED.setFont(MegaNumbers);
        myOLED.print( max_SPD, 0, 7);
        myOLED.setFont(MediumFont);
        myOLED.print("Max speed:",0, 1);
        myOLED.setFont(MediumFont);
        myOLED.print("KM/H",75, 7);
        if(holMeny==1)
        {
            max_SPD=0;
            EEPROM.put(5,0);
        }
        break;
    case 2://другий екран меню

        myOLED.setFont(SmallFont); // Вказуємо шрифт, потрібний для виведення цифр і текста.
        myOLED.print("Razgon ot 0 do 60:", 0, 0); // Виводимо текст починаючи з
0 стовпця
        myOLED.setFont(SmallFont); // Вказуємо шрифт, потрібний для виведення цифр і
текста.
        myOLED.print("Seconds.", 80, 2); // Виводимо текст починаючи з 0 стовпця
        myOLED.setFont(MediumFont); // Вказуємо шрифт, потрібний для виведення цифр і
текста.
        myOLED.print(rozg_60, 0, 2); // Виводимо текст починаючи з 0 стовпця

        myOLED.setFont(SmallFont); // Вказуємо шрифт, потрібний для виведення цифр і текста.
        myOLED.print("Razgon ot 0 do 100:", 0, 4); // Виводимо текст починаючи з
0 стовпця
        myOLED.setFont(SmallFont); // Вказуємо шрифт, потрібний для виведення цифр і текста.
        myOLED.print("Seconds.", 80, 6); // Виводимо текст починаючи з 0 стовпця

```

```

        myOLED.setFont(MediumFont);    // Вказуємо шрифт, потрібний для виведення цифр і
текста.
        myOLED.print(rozg_100, 0, 6);    //Виводимо текст починаючи з 0 стовпця
        break;
        break;
        //3-й екран меню
        case 3://servises
            myOLED.setFont(SmallFont);
            myOLED.print("SERVISES...",0, 1);
            myOLED.print("-----",0, 2);
            if(oil_Servis<1000&&cont_meny==3){
                myOLED.invText(true);
                myOLED.setFont(MediumFont);
                myOLED.print("Change Oil", 0, 4);    //Виводимо текст починаючи з 0 стовпця
                myOLED.invText(false);
            }
            else
            {
                myOLED.setFont(MediumFont);    // Вказуємо шрифт, потрібний для виведення цифр і
тексту.
                myOLED.print("Change Oil", 0, 4);
            }    //Виводимо текст починаючи з 0 стовпця
            myOLED.setFont(MediumFont);    // Вказуємо шрифт, потрібний для виведення цифр і
тексту.
            myOLED.print("Km", 100, 7);    // Виводимо текст починаючи з 0 стовпця
            myOLED.setFont(MediumFont);    // Вказуємо шрифт, потрібний для виведення цифр і
тексту.
            myOLED.print(oil_Servis, 0, 7);    // Виводимо текст починаючи з 0 стовпця
            break;

        case 4://svitlo
            myOLED.setFont(MediumFont);
            myOLED.print("HODOVIE", 25, 1);
            myOLED.print("OGNI", 40, 3);
            myOLED.print("|", 60, 7);
            myOLED.setFont(SmallFont);
            myOLED.print("-----", 0, 4);
            myOLED.setFont(MediumFont);
            if(svitlo_flag==1){
                myOLED.invText(true);
                myOLED.print("ON", 20, 7);
                myOLED.invText(false);
            }
            else{myOLED.print("ON", 20, 7);
                if(svitlo_flag==0){
                    myOLED.invText(true);
                    myOLED.print("OFF", 80, 7);
                    myOLED.invText(false);
                }
            }
            else{ myOLED.print("OFF", 80, 7); }
            if(holMeny==1)

```

```

    {
    svtlo_flag=svtlo_flag==0?1:0;
    EEPROM.put(25,svtlo_flag);
    }
break;}}
int GetFiltrSpeed(int spd,int filtSpeed)
{
if(spd>0&&spd<160&&spd>filtSpeed)
{
    filtSpeed++;
}
else if(spd>=0&&spd<160&&spd<filtSpeed)
{filtSpeed--;}
return filtSpeed; }
void RAPID()
{
    timer_SPD=micros()-timer1;
    timer1=micros();
    if(ign_flag==2&&timer_SPD>10000)
    {odo_counter++;
    odo1_counter++;
    serv_counter++; }}

```

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011591429

Дата перевірки:
16.06.2022 08:10:33 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
16.06.2022 08:11:51 EEST

ID користувача:
100005591

Назва документа: Більковський_Розробка програмно-технічного пристрою для вимірювання миттєвої швид...
Кількість сторінок: 68 Кількість слів: 9200 Кількість символів: 71173 Розмір файлу: 7.35 MB ID файлу: 1011460524

2.98% Схожість

Найбільша схожість: 0.72% з джерелом з Бібліотеки (ID файлу: 1011339392)

2.21% Джерела з Інтернету

32

Сторінка 70

0.99% Джерела з Бібліотеки

82

Сторінка 70

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 10%**

ID: 105612 Название: Розробка програмно-технічного пристрою для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P Добавлено в БД: 2022-06-16 Авторы: Я. В. Біпковський Руководители: А. О. Нічепорук Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	61710	550	823 (1%)	12 (2%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Біньковський Ярослав Васильович

Тема: Програмно-технічний засіб для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано програмно-технічний пристрій для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P

2. Висновок про відповідність роботи дипломному завданню _____
Дипломний проект відповідає виданому завданню _____

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз предметної області. У другому розділі здійснено вибір елементної бази. У третьому розділі приведено програмно-апаратну реалізацію лічильника для вимірювання миттєвої швидкості руху.

4. Позитивні сторони роботи: _____ Здійснено практичну реалізацію програмно-технічного засобу для вимірювання миттєвої швидкості руху з лічильником відстані на базі _____

5. Негативні сторони роботи: В роботі не наведено порівняння запропонованого рішення із відомими аналогами, відсутні оцінки кількісних та якісних характеристик розробленою пристрою.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на задовільному рівні.

8. Інші зауваження: -

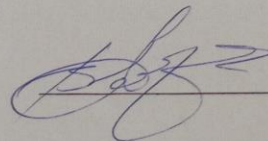
9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно» 3,25 (D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Чешаки Віктор Миколайович, к.т.н.

Доцент кафедри кібербезпеки

" 16 " 06 2022р.



Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорушенко Т. О.

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-19-1

ЗАЯВА

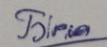
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

16.06.2022

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб для вимірювання миттєвої швидкості руху з лічильником відстані на базі мікроконтролера Atmega 328P

Автор: Біньковський Ярослав Васильович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Нічепорук Андрій Олександрович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

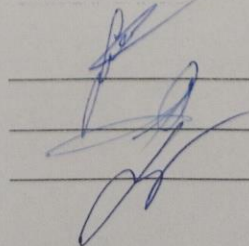
- 1) запозичення розміщені в розділі аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів щодо використаних програмних скриптів, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.98% і адресується до 112 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру практичної роботи і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



А.О. Нічепорук

С.М. Лисенко

Т. О. Говорушенко