

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему:

*Велокомп'ютер на базі Arduino*

Виконав: студент

IV курсу, групи СІс-42

спеціальності

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

*Грухін М.В.*

(прізвище та ініціали)

Керівник

(підпис)

*Жаровський Р.О.*

(прізвище та ініціали)

Нормоконтроль

(підпис)

*Луцик Н.С.*

(прізвище та ініціали)

Завідувач кафедри

(підпис)

*Осухівська Г.М.*

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

«\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студента Грухіна Михайла Вікторовича  
(прізвище, ім'я, по батькові)

1. Тема роботи Велокомп'ютер на базі Arduino

Керівник роботи Жаровський Руслан Олегович, к.т.н., доцент кафедри КС  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «26» квітня 2024 року № 4/7-468

2. Термін подання студентом завершеної роботи 24.06.2024 р.

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Загальна архітектура системи (A1)

2. Структурна схема системи (A1)

3. Схема електрична принципова (A1)

4. Алгоритм роботи програмного забезпечення (A1)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., професор кафедри МТ</i>		

7. Дата видачі завдання

26.04.2024

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>01.02-09.02.2024</i>	
2	<i>Аналіз технічного завдання</i>	<i>05.02-11.02.2024</i>	
3	<i>Визначення і аналіз вимог до веловелокомп'ютера</i>	<i>10.05-15.05.2024</i>	
4	<i>Проектування структури веловелокомп'ютера</i>	<i>16.05-20.05.2024</i>	
5	<i>Проектування програмної частини</i>	<i>21.05-28.05.2024</i>	
6	<i>Проектування апаратної частини</i>	<i>29.05-06.06.2024</i>	
7	<i>Налаштування і тестування веловелокомп'ютера</i>	<i>07.06-10.06.2024</i>	
8	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>10.06-15.06.2024</i>	
9	<i>Оформлення кваліфікаційної роботи</i>	<i>16.06-20.06.2024</i>	
10	<i>Попередній захист кваліфікаційної роботи</i>	<i>14.06.2024</i>	
11	<i>Захист кваліфікаційної роботи</i>	<i>24.06-28.06.2024</i>	

Студент

\_\_\_\_\_ (підпис)

*Грухін Михайло Вікторович*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Жаровський Руслан Олегович*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Велокомп'ютер на базі Arduino // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Грухін Михайло Вікторович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія» // Тернопіль, 2024 // с.– 89, рис. – 40, аркушів А1 – 4, бібліогр. – 15.

Ключові слова: велокомп'ютер, Arduino, датчик Холла.

У даній кваліфікаційній роботі виконано розробку програмно-апаратного комплексу для отримання інформації про велопоїздку.

Пояснювальна записка містить 4 розділи.

В першому розділі здійснюється аналіз предметної області. Проведено огляд вимог до велокомп'ютера, а також визначені задачі кваліфікаційної роботи.

В другому розділі описані компоненти і особливості проектування велокомп'ютера.

В третьому розділі здійснено проектування плати велокомп'ютера, наведено процес налагодження і тестування.

В четвертому розділі розглянуті питання охорони праці та безпеки життєдіяльності.

## ABSTRACT

Arduino-based bicycle computer // Bachelor's thesis // Grukhin Mykhailo Viktorovych // TNTU, specialty 123 "Computer engineering" // Ternopil, 2024 // p.–89, fig. - 40, sheets A1 – 4, bibliography. - 15.

Keywords: bicycle computer, Arduino, Hall sensor.

In this qualification work, the software and hardware complex for obtaining information about a bicycle trip was developed.

The explanatory note contains 4 sections.

In the first section, the subject area is analyzed. A review of the requirements for the bicycle computer was carried out, as well as the tasks of the qualification work were defined.

The second section describes the components and design features of the bicycle computer.

In the third section, the design of the bike computer board is carried out, the debugging and testing process is given.

In the fourth chapter, issues of labor protection and life safety are considered.

## ЗМІСТ

ВСТУП .....	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ .....	10
1.1 Аналіз вимог до системи.....	10
1.2 Аналіз можливих рішень поставленого завдання .....	11
1.3 Постановка задач кваліфікаційної роботи.....	17
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА.....	18
2.1 Розробка структури велокомп'ютера.....	18
2.2 Обґрунтування вибору апаратного забезпечення.....	19
2.2.1 Розробка схеми електричної принципової .....	19
2.2.2 Вибір та розрахунок елементів.....	21
2.3 Обґрунтування вибору програмного забезпечення.....	33
2.3.1 Розробка програми велокомп'ютера .....	33
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	41
3.1 Апаратна частина .....	41
3.2 Налагодження та експериментальні дослідження.....	44
3.3 Виготовлення велокомп'ютера.....	47
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..	53
4.1 Долікарська допомога при вивихах .....	53
4.2 Проведення інструктажів з охорони праці.....	55
ВИСНОВКИ.....	59

					<b>КС КРБ 123.312.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата	<i>Велокомп'ютер на базі Arduino</i>	Літ.	Арк.	Аркуші
Розроб.		Грухін М.В.					6	
Перевір.		Жаровський				<b>ТНТУ, каф. КС, гр. СІс-42</b>		
Реценз.								
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
Додаток А. Технічне завдання.....	62
Додаток Б. Характеристики NX4827P043-011R.....	69
Додаток В. Алгоритм роботи програмного забезпечення.....	72
Додаток Г. Лістинг програмного забезпечення.....	73

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК СКОРОЧЕНЬ

GPS	Global Positioning System
RPM	revolutions per minute — одиниця вимірювання частоти обертання
ОС	операційна система
ТЗ	технічне завдання

					<i>КС КРБ 123.312.00.00 ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



## ВСТУП

Велокомп'ютер – це електронний пристрій для визначення параметрів і статистичних даних таких як швидкість і пройдена відстань, а також додаткові параметри. Це підрахунок каденса, середньої, максимальної і мінімальної швидкості, відображення фактичного часу і часу в дорозі, пробіг за добу і загальний пробіг, температура, вологість, тиск, підрахунок пульсу.

Однак запроваджувати велокомп'ютери стали лише з 1990 року, коли в період різького і потужного підйому велоіндустрії та використання велосипедів, як засобу пересування, так і в спорті, зокрема. Перший електронний велокомп'ютер був розроблений в 1981 році, а перший у світі бездротовий велокомп'ютер AERO 8 був створений VDO в 1991 році.

У той період перед розробниками виникло ряд нових і непростих завдань. Як створити компактний пристрій який володіє зручними органами управління та користувальницьким інтерфейсом, надійним кріпленням і корпусом, стійким до дії ударів, вібрації, різних погодних умов, який забезпечуватиме велосипедиста інформацією про швидкість руху: поточну, середню, максимальну швидкість, відхилення поточної швидкості від середніх, часових характеристик подорожі, таких як: час в дорозі, секундомер, годинник, таймер.

Моделі сучасних на той момент часу велокомп'ютерів відразу ж завоювали популярність.

Метою кваліфікаційної роботи є розробка комп'ютеризованої системи на базі Arduino, яка буде визначати показники подорожі і буде виводити їх на екран.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Аналіз вимог до системи

Перші велокомп'ютери розроблялись для використання у напівпрофесійному чи професійному велоспорті де спортсмен за один день проїжджає від 100 до 250 км. Саме цим людям інформація про пройдений шлях і про стан організму є корисною в плані статистики тренувань, так і в плані оптимального функціонування організму велосипедиста.

Однак разом із популяризацією велоспорту персональні велокомп'ютери почали використовувати звичайні велосипедисти. Звичайно такі побутові велокомп'ютери мали простіший функціонал і меншу точність однак більшість базових функцій вони підтримують.

Одним із таких важливих для здоров'я велосипедиста параметрів, окрім швидкості та пробігу, що становлять основу велокомп'ютера, є моніторинг каденсу.

Каденс – це частота обертів педалей за хвилину. Нормальний каденс велосипедиста становить від 80 до 110 обертів за хвилину [1]. Якщо велосипедист не дотримується цього параметра, то це загрожує проблемами з суглобами колін, а також з'являється загроза інвалідності. Тільки через ці три параметри велокомп'ютер вже становить чималу цінність у напівпрофесійному та професійному середовищі.

Також дуже важливий моніторинг пульсу. Кожен велосипедист має тримати свій пульс у певних межах. Існують межі для ефективного тренування серцевої системи, межі для ефективного схуднення та межі підвищеного зносу серцевої системи.

					<b>КС КРБ 123.312.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Грухін М.В.</i>			<i>Аналіз технічного завдання</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Жаровський</i>					10	
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-42</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Необхідність залишатися в цих межах свідчить про важливість даного параметра, проте він дуже витратний оскільки необхідно використовувати фізіологічні датчики, які мають мати високу точність і зазвичай реалізовані як окремий повноцінний пристрій.

## 1.2 Аналіз можливих рішень поставленого завдання

Перед тим як приступити до виконання кваліфікаційної роботи необхідно визначити основні компоненти, функціонал, а також загальну компоновку веловелокомп'ютера. Найпростіший велокомп'ютер складається з основного блоку, датчика швидкості та датчика оборотів (каденсу).

Основний блок містить у собі всю електронну частину веловелокомп'ютера:

- процесор;
- електронну обв'язку;
- дисплей;
- живлення;
- кронштейн кріплення.

На рис. 1.1 зображено приклад основного блоку веловелокомп'ютера.



Рисунок 1.1 – Основний блок веловелокомп'ютера

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.312.00.00 ПЗ

Арк.

11

Датчиком швидкості є датчик холу в комплекті з магнітом, так само як і датчик оборотів (каденс). Приклади датчиків швидкості та обертів зображені рис. 1.2 і 1.3.



Рисунок 1.2 – Датчик швидкості



Рисунок 1.3 – Датчик обертів (каденсу)

Умовно велокомп'ютери можна розділити на:

- дотові велокомп'ютери;

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- бездротові велокомп'ютери;
- GPS велокомп'ютери;
- SMART -велокомп'ютери.

Найпростіший варіант це дротові велокомп'ютери – це ті комп'ютери, у яких всі датчики підключені до основного блоку через фізичні кабелі. Як правило такі велокомп'ютери, мають невеликий набір функцій, та монохромний дисплей. Керування таким велокомп'ютером здійснюється з допомогою однієї, або декількох фізичних кнопок.

Приклад такого веловелокомп'ютера зображено рис. 1.4.



Рисунок 1.4 – Дротовий велокомп'ютер

Плюси цього виду: невисока ціна, простота конструкції, невеликі витрати ресурсів на живлення, відсутність завад від інших пристроїв.

Мінуси цього виду: наявність дротів, що проходять по всьому велосипеду, ймовірність розриву дротів.

Бездротові велокомп'ютери - це велокомп'ютери, в яких датчики передають всю зібрану інформацію бездротовим інтерфейсом у вигляді

закодованого сигналу. Приклад бездротового веловелокомп'ютера наведено на рис.1.5.



Рисунок 1.5 – Бездротовий велокомп'ютер

Плюси даного виду веловелокомп'ютера: відсутність проводів, компактніший і кращий зовнішній вигляд порівняно з провідним велокомп'ютером, відсутність ймовірності обриву проводів. Відсутність перехідних контактів. Зручність знімання велокомп'ютера з велосипеда з метою запобігання крадіжці.

Мінуси цього велокомп'ютера: потреба в джерелі живлення для кожного датчика. Великі витрати на батареї. Можливі завади та погіршення роботи веловелокомп'ютера під час проїзду під високовольтними лініями електропередач.

GPS -велокомп'ютери – це комп'ютери з вбудованим модулем GPS, що дозволяє дізнаватися швидкість велосипеда за даними супутників, переглядати за наявності спеціального графічного додатка пройденого



маршруту, а також бачити дані про положення над рівнем моря, температуру та атмосферний тиск. Один із таких велокомп'ютерів зображений на рис.1.6.



Рисунок 1.6 – Велокомп'ютер із модулем GPS

Плюси цього виду велокомп'ютера: Більший функціонал. Можливість підключення до персонального велокомп'ютера для перегляду та аналізу даних. Можливість поновлення прошивки.

Мінуси цього велокомп'ютера: Мінімальний час роботи через більше енергоспоживання. Необхідність ПК для перегляду всіх даних. Можлива втрата GPS сигналу. Велика вартість.

Smart-велокомп'ютери – це комп'ютери, які знаходяться на рівні приблизно рівному рівню стільникових телефонів. У таких комп'ютерів є операційна система, так само є датчики пульсу, температури, вологості та тиску. Дані комп'ютери можуть безпосередньо під час руху коригувати ваше

навантаження на серце і за зібраною інформацією скласти режим вашої їзди. Крім цього відбувається запис маршрутів та параметрів поїздки для самостійного аналізу. Приклад такого веловелокомп'ютера показано рис. 1.7.

Плюси цього велокомп'ютера це висока продуктивність. Висока функціональність. Інтуїтивно зрозумілий інтерфейс. Сенсорне керування. Можливість контролю фізіологічних параметрів велосипедиста.

Мінуси цього велокомп'ютера: Дуже висока вартість. Менш стійкий при падіннях, оскільки використовує великий сенсорний екран. Велике енергоспоживання.



Рисунок 1.7 – Smart-велокомп'ютер

Існуючі пристрої мають деякі недоліки і не можуть повною мірою задовольнити всі вимоги велосипедистів. Тому необхідно розробити власний пристрій, який матиме перевагу перед існуючими і в подальшому дасть можливість додавати нові функції без кардинальних змін архітектури пристрою.



### 1.3 Постановка задач кваліфікаційної роботи

Метою кваліфікаційної роботи є розробка велокомп'ютера з можливістю вимірювання швидкості і пройденої відстані, по можливості створення діючого макета пристрою.

Для досягнення цієї мети було поставлено, а потім виконано такі завдання:

- аналіз існуючих рішень;
- розробка схеми електричної принципової;
- вибір елементів схеми;
- розробка друкованої плати;
- розробка програми пристрою;
- виготовлення пристрою;
- налагодження пристрою.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Розробка структури велокомп'ютера

Для опису загальної структури велокомп'ютера необхідно розробити функціональну схему системи. На цій схемі показані основні функціональні блоки які необхідно буде розробити в кваліфікаційній роботі (рис. 2.1).

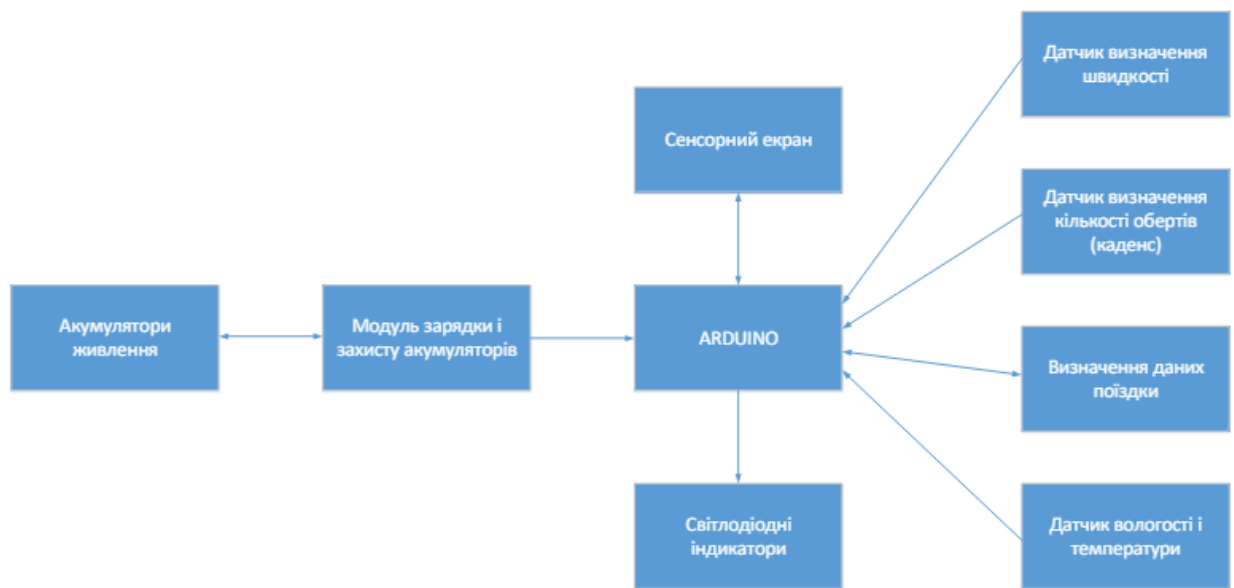


Рисунок 2.1 – Загальна архітектура системи

На цій схемі наведені основні функціональні елементи, які забезпечують роботу велокомп'ютера. На базі функціональних компонентів велокомп'ютера необхідно розробити структурну схему пристрою та визначити принцип дії пристрою. Тому проектування велокомп'ютера розпочнемо з апаратної частини.

					<b>КС КРБ 123.312.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Грухін М.В.			<i>Проектна частина</i>	<b>Лім.</b>	<b>Арк.</b>	<b>Аркушіє</b>
Перевір.		Жаровський					18	
Реценз.						<b>ТНТУ, каф. КС, гр. СІс-42</b>		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

## 2.2 Обґрунтування вибору апаратного забезпечення

### 2.2.1 Розробка схеми електричної принципової

Структурна схема пристрою наведена на кресленні КС КРБ 123.312.00.00 Е1.

Розглянемо призначення і принцип дії окремих елементів. Через роз'єм, гніздового типу XS1, подається живлення з акумулятора, на пристрій. Для безпечної експлуатації пристрою та для захисту акумулятора від швидкої деградації встановимо плату захисту, яка забезпечить захист акумулятора від глибокого розряду, від перезаряду, від короткого замикання та переплюсування.

Потім напруга надходить на транзисторний ключ, який комутує плюс у схемі. Також крім ключа, напруга з акумулятора надходить і на тактову кнопку з двома контактами, нормально-замкнутий і нормально-розімкнутий.

Нормально-розімкнутий контакт використовується для включення велокомп'ютера, а нормальнозамкнутий використовується як подача сигналу на мікроконтролер для подальшої обробки. Використовуватиметься сигнал логічного нуля, для вимикання велокомп'ютера або переведення його в режим налаштування та вихід з цього режиму. А транзисторний ключ виконуватиме функцію самоблокування і підтримуватиме живлення у схемі, поки воно потрібно і розривати його, коли вже буде не потрібно, без участі механічних контактів.

Після транзисторного ключа, напруга з акумулятора надходить на підвищуючий DC/DC перетворювач, який видає на своєму виході стабільні 5 вольт в основну мережу живлення велокомп'ютера.

Для того, щоб відстежувати велокомп'ютером рівень заряду акумулятора і демонструвати це користувачеві цього пристрою з виходу плати захисту акумулятора подається напруга на другий транзисторний ключ

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Відповідно до технічного завдання в якості мікроконтролера планується використовувати платформу Arduino. Мікроконтролер Arduino буде виконувати всі основні функції велокомп'ютера, вимірювання, а також обчислення.

Роз'єми ХР2, ХР3, використовуються для комутації сигналу на мікроконтролер. Сигнал з датчиків Холла, які підключаються до цих роз'ємів, а також підведення живлення 5 вольт.

Роз'єм ХР4 служить для комутації з мікроконтролером датчика температури та вологості, для отримання інформації, а також для підведення живлення до даного датчика.

Через роз'єм Х1 підключається дисплей з резистивним сенсором. Через цей роз'єм відбувається комутація на дисплей і сенсор, в даному випадку необхідно 5 вольт, а також шина для обміну інформацією дисплея, сенсора і мікроконтролера.

Якщо у велокомп'ютері присутні датчики Холла, то необхідно полегшити їх установку для цього необхідно при їх встановленні на велосипед візуально контролювати чи спрацьовує датчик Холла чи ні, для цього було вирішено застосувати пристрій порівняння і для візуальної частини на схемі передбачені світлодіодні індикатори.

Розглянемо принцип роботи велокомп'ютера докладніше на основі схеми зображеної на кресленні КС КРБ 123.312.00.00 Е1.

Коли необхідно увімкнути велокомп'ютер, потрібно натиснути на тактову кнопку та підтримати її натиснутою близько 1-2 секунд, після чого можна відпустити і пристрій залишиться увімкненим.

При натисканні на тактову кнопку, ми закорочуємо транзисторний ключ, тим самим подаємо напругу на перетворювач який перетворює напругу з акумулятора в стабільні 5 вольт, далі мікроконтролер включається, починаючи виконання програми і один з пунктів якої є відкрити транзисторний ключ.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Надалі напруга вже проходить, через нього і це буде продовжуватися доти, поки ключ не буде знову закритий.

Велокомп'ютер через роз'єм ХР4 отримує дані від датчика температури та вологості та відправляє їх на дисплей, а через роз'єми ХР2 та ХР3 мікроконтролер отримує дані з датчиків Холла, якщо велосипед рухається. Далі ці дані обробляються, підраховується швидкість і оберти і інформація відправляється на дисплей через шину даних.

Також буде потреба налаштувати датчики Холла і проводити інші налаштування велокомп'ютера, для цього буде передбачено режим налаштування та калібрування. Для його активації натискаємо на кнопку включення і тримаємо близько 3 секунд, доки не почне світитися зелений світлодіод, потрібно відпустити кнопку і можна робити налаштування та зберігати їх.

Після того, як все було зроблено, необхідно знову натиснути на кнопку і почекати близько 3 секунд, поки не згасне зелений світлодіод, після чого, можна знову використовувати велокомп'ютер.

На основі проведеного вище аналізу була розроблена схема електрична принципова наведена на кресленні КС КРБ 123.312.00.00 ЕЗ.

### 2.2.2 Вибір та розрахунок елементів

Відповідно до технічного завдання, живлення велокомп'ютера має бути автономним, у зв'язку з цим для живлення використовуватиметься акумулятор або кілька акумуляторів, з'єднаних послідовно або паралельно.

Розглянувши всі плюси і мінуси потенційних акумуляторів був обраний літій-іонний акумулятор типорозміру 18650. Причини, що послужили вибору даного акумулятора, наступні: досить висока напруга для перетворення, що дозволяє використовувати всього один акумулятор або кілька сумарної ємності. Також цей акумулятор дозволяє протягом майже всієї роботи

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримувати достатній струм навантаження до самого розряду акумулятора [4].

Для живлення велокомп'ютера підійдуть акумулятори з наступними характеристиками:

- номінальна напруга – 3,7 В;
- ємність акумулятора - 2400 міліампер/годин;
- вбудований захист від перезаряду - є;
- тип акумулятора - літій-іонна (Li - ion );
- типорозмір – 18650;
- більше 500 циклів заряду-розряду без ефекту пам'яті.

Так як акумулятори були вибрані літій-іонні, для них як згадувалось раніше необхідна плата захисту пристрою. Було вибрано, вже готовий модуль TP 4056 [5], зовнішній вигляд даного модуля зображений на рисунку 2.2. Тому було вирішено використовувати вже готовий модуль через його компактні розміри, завдяки поверхневому монтажу і ціна готового модуля виходить дешевше, ніж установка таких елементів у схему.

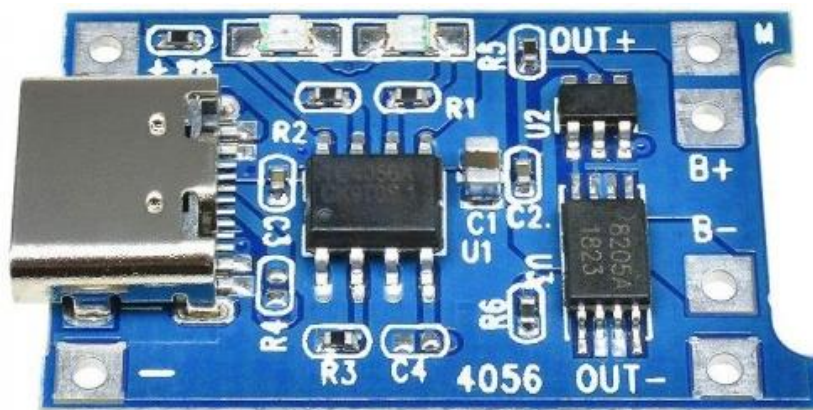


Рисунок 2.2 – Зовнішній вигляд модуля захисту TP 4056

Модуль має наступні характеристики:

- напруга живлення – від 4,5 до 8,0 В;
- струм споживання трохи більше – 1,0 А;

- програмований струм заряду - від 0,1 до 10 А;
- напруга закінчення заряду акумулятора – 4,2 В;
- вбудований захист від переплюсування;
- вбудований захист від перевантаження струмом - обмеження 3 А;
- вбудований захист від глибокого розряду акумулятора – 2,4 В;
- наявність індикації (заряду) шляхом червоного світлодіода;
- наявність індикації (кінець зарядки) шляхом зеленого світлодіода;
- наявність USB роз'єму - типу Type-C;
- розміри модуля 15x25,5 мм;
- вага модуля 2,1 г.

Схема даного модуля зображена на рисунку 2.3. Даний модуль зібраний на мікросхемі TP4056. Ця мікросхема є контролером зарядки літій-іонних акумуляторів з вбудованим термодатчиком.

Цей модуль не передбачає використання вбудованих в акумулятори датчиків температури, тому для відключення цієї функції вивід 1 мікросхеми з'єднаний з мінусом живлення. Так само мікросхема може йти в сон і виходити з нього і дана функція так само не була виведена в цьому модулі, тому вихід 8 мікросхеми з'єднаний з плюсом живлення, так як зі сну мікросхема виходить при надходженні на 8 вихід логічної одиниці.

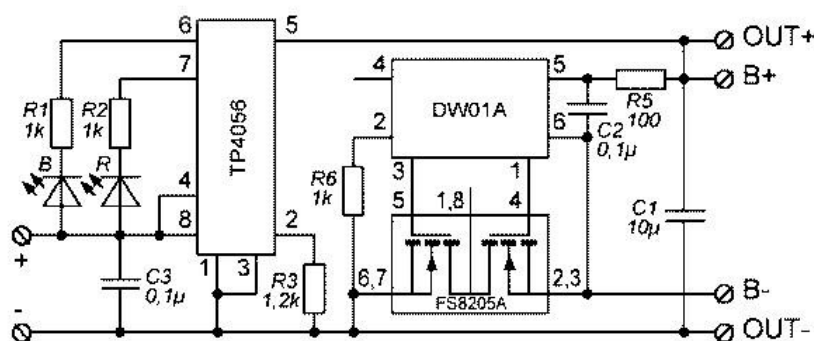


Рисунок 2.3 – Схема модуля TP 4056

Для захисту від глибокого розряду, короткого замикання та обмеження споживаного струму використано мікросхему DW01 та здвоєні ключі MOSFET FS8205.

Мікросхема DW01 є контролером для літій-іонного акумулятора і стежить за споживанням струму і переплюсуванням навантаження, а також досягнення акумулятором напруги 2,4 В для його відключення. Захист здійснюється за допомогою двох польових транзисторів.

Здвоєний польовий ключ FS8205 відповідає всім необхідним характеристикам для повноцінної роботи захисту.

Розглянемо призначення контактів цього модуля. контакти IN + та IN - служать для підключення вхідної (зарядної) напруги, якщо потрібно використовувати інше джерело живлення, а не від USB роз'єму.

Контакти В + і В - є входом вбудованого захисту і до них підключається один акумулятор або два паралельно. А також контакти OUT + і OUT - служать виходом захисту та підключення навантаження, у разі спрацьовування захисту акумулятор буде відключений від навантаження.

Далі розглянемо підвищувальний перетворювач напруги. Даний перетворювач необхідний у схемі для підвищення напруги до 5 В. Було обрано модуль МТ 3608, який має наступні характеристики:

- вхідна напруга – від 2 до 24 В;
- максимальна вихідна напруга – 28 В;
- максимальний вхідний струм – 2 А;
- ККД перетворювача  $\leq 93\%$ ;
- розміри модуля 17x36 мм;
- вага модуля 1,4 г.

Зовнішній вигляд модуля зображений на рисунку 2.4. Модуль побудовано на базі мікросхеми МТ 3608, яка являє собою регульований перетворювач напруги [6]. Мікросхема виробляється в корпусі SOT 23.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		



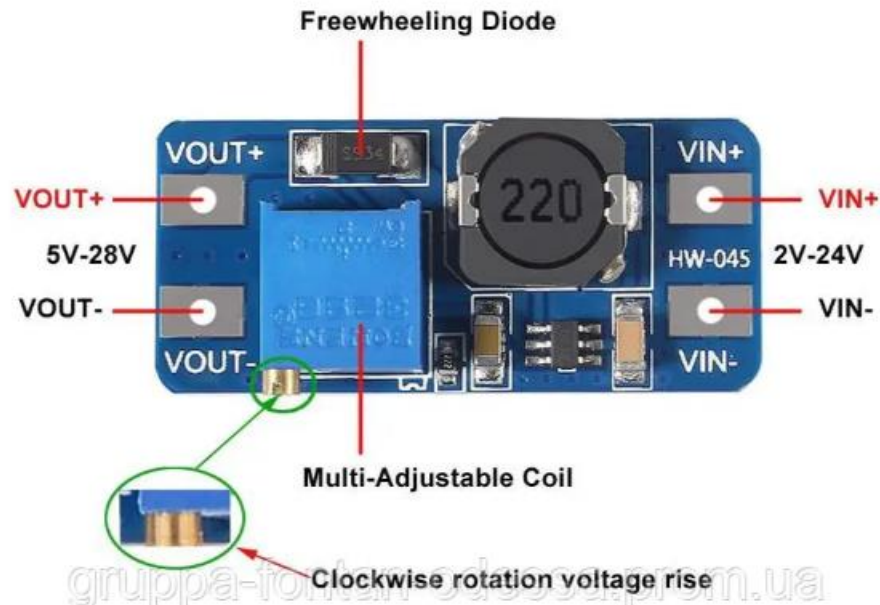


Рисунок 2.4 – Зовнішній вигляд модуля MT 3608

Так само згідно зі схемою нам необхідний мікроконтролер для здійснення всіх обчислень, відповідно до технічного завдання був обраний готовий модуль Arduino nano (рис. 2.5). Даний модуль був обраний через свої компактні розміри, вже встановлений на нього обв'язці необхідного для роботи велокомп'ютера функціоналу, а також невеликої вартості.

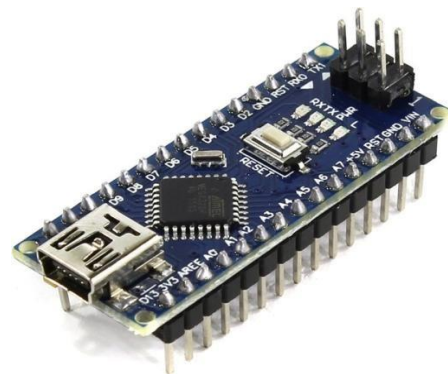


Рисунок 2.5 – Зовнішній вигляд модуля Arduino nano

Як видно з рисунка 2.6, модуль складається з мікроконтролера Atmega 328.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

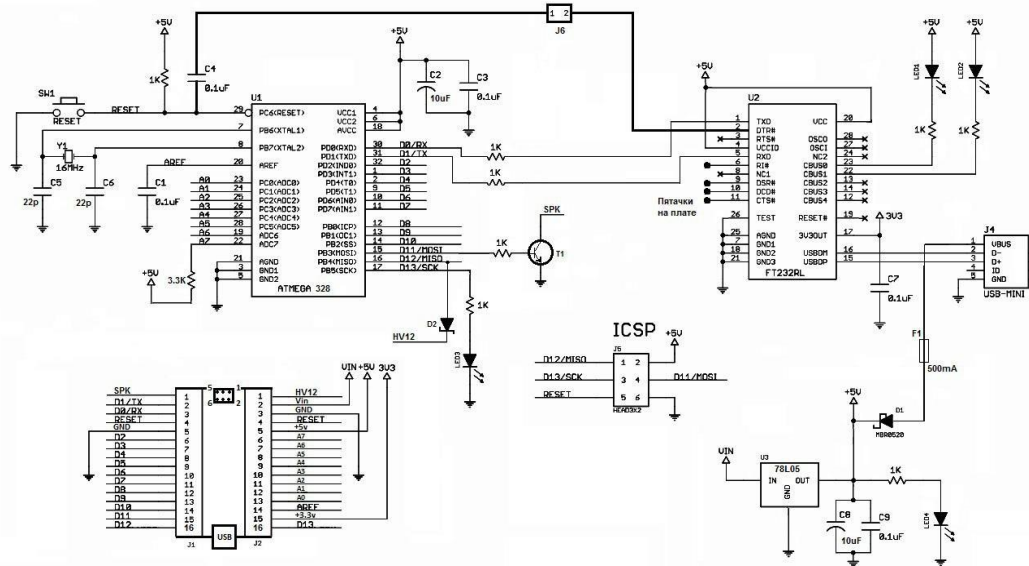


Рисунок 2.6 – Схема модуля Arduino nano

Є необхідні світлодіоди, що сигналізують про наявність живлення на мікроконтролері, про наявність логічної одиниці на 13 виході модуля, а також ще два з маркуванням RX і TX, за якими можна відстежувати передачу і прийом сигналів по послідовному порту. Програму мікроконтролер отримує по послідовному порту, який інтегрований у мікроконтролер. Щоб цю програму до послідовного порту відправляти, з комп'ютера, в даному модулі використана мікросхема FT 232. Ця мікросхема являє собою однокиповий перехідник з USB в асинхронний послідовний інтерфейс UART, протокол USB повністю інтегрований в цю мікросхему. Також присутній лінійний стабілізатор, для живлення модуля більшим за 5 В. Цей модуль має такі характеристики, взяті з сайту виробника, наведені на рисунку 2.7.

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.312.00.00 ПЗ

Арк.

26

Характеристики:

Мікроконтролер	Atmel ATmega168 або ATmega328
Робоча напруга (логічний рівень)	5В
Напруга живлення (рекомендована)	7-12В
Напруга живлення (гранична)	6-20В
Цифрові входи/виходи	14 (з яких 6 можуть використовуватися як ШИМ-виходи)
Аналогові входи	8
Максимальний струм одного висновку	40 мА
Flash-пам'ять	16 КБ (ATmega168) або 32 КБ (ATmega328) з яких 2 КБ використовуються завантажувачем
SRAM	1 КБ (ATmega168) або 2 КБ (ATmega328)
EEPROM	512 байт (ATmega168) або 1 КБ (ATmega328)
Тактова частота	16 МГц
Розміри плати	1.85 см x 4.3 см

Рисунок 2.7 – Характеристики модуля Arduino nano

Розглянемо призначення виходів модуля. На рисунку 2.8 зображені всі виходи модуля та їх назви.

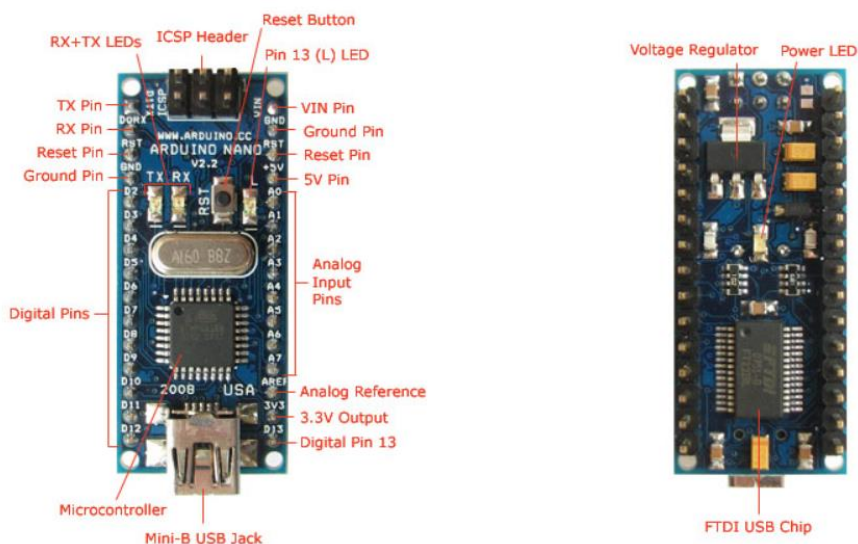


Рисунок 2.8 – Розташування та найменування виходів Arduino nano .

Почнемо зліва-зверху, виводи TX та RX [7]. Це послідовна шина в даному модулі, а дані виводи використовуються для отримання та передачі даних за інтерфейсом UART. Далі вихід Reset використовується для можливості перезавантаження мікроконтролера, якщо немає можливості зробити це натисканням на кнопку Reset, яка так само встановлена на платі модуля.

Вихід з назвою Ground використовується для загального мінуса всього модуля. Потім йдуть виходи Digital, це означає, що дані цифрового виходу і можуть налаштовуватися як на вхід, так і на вихід. Вони нумеруються, починаючи від нуля, і позначаються англійською літерою D, усього на модулі 14 цифрових виходів, 6 можуть генерувати широтно-імпульсну модуляцію, з роздільною здатністю 8 біт.

Також модуль оснащений роз'ємом Mini - B USB для здійснення прошивки мікроконтролера або отримання даних з того ж мікроконтролера при обробці ним сигналів. З правого боку ж ми бачимо останній 13 цифровий вихід, який вже був розглянутий, так само поряд з ним присутній вихід напруги 3,3 В, він потрібен для живлення датчиків з напругою живлення 3,3 В.

Наступний вихід, який бачимо на рисунку 20, це Analog Reference. Даний вихід служить для подачі на нього опорної напруги, щоб надалі аналого-цифровий перетворювач мікроконтролера працював з ним, це необхідно для роботи зі слабкими сигналами, для підвищення точності.

Потім можна бачити, що йдуть виходи під назвою Analog Input. Це аналогові входи, які надходять на аналого-цифровий перетворювач. У даному мікроконтролері 10 бітний перетворювач. Всього в даному модулі 8 аналогових входів, які можна використовувати як цифрові.

Далі ми бачимо вихід для подачі на нього 5 В, або якщо модуль вже отримує живлення, наприклад від USB або стабілізатора, тоді можна взяти 5 В, наприклад для живлення датчиків. Так само поруч знаходиться висновок Vin, це і є вхід для подачі живлення, яке більше 5 В, але не більше гранично допустимої напруги, вказаної в характеристиках на даний модуль. Поруч зверху модуля можна побачити роз'єм ICSP. Він дає можливість використовувати наш модуль як USB- програматор.

Необхідно звернути увагу на те, що деякі виходи модуля Arduino nano мають кілька функцій, як наприклад вказувалося вище, що деякі виходи можуть генерувати широтно-імпульсну модуляцію, а зокрема це цифрові

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

виходи під наступними номерами: 3,5,6,9,10 і 11. А так само, на цифрових виходах 1 і 0 є шина послідовного інтерфейсу, що описувалося трохи вище.

На цифрових виходах 2 і 3 є апаратне переривання. Тобто на даних цифрових виходах можна викликати переривання програми на передньому або задньому фронті імпульсу, а так само при зміні значення. На аналогових виходах 4 і 5 здійснюється зв'язок по I2C інтерфейсу. Саме вихід 4 є SDA а, вихід 5 SCL .

Для комфортного використання велокомп'ютера велосипедистом було вирішено додати до пристрою датчик температури та вологості. Було обрано датчик температури DHT 11, зовнішній вигляд даного датчика зображено рисунку 2.9. Також основні характеристики датчика, взяті з сайту виробника, зображені рисунку 2.10.

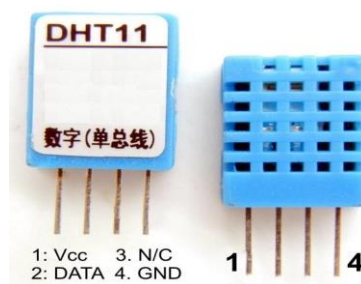


Рисунок 2.9 – Зовнішній вигляд датчика температури DTH 11

#### Характеристики:

- Модель виробника: ASAIR DHT11
- Визначення вологості: 5 - 95% RH  $\pm$  5% (макс.)
- Визначення температури: -20 ~ +60  $^{\circ}$ C  $\pm$  2% (макс.)
- Живлення: 3.5-5.5 В
- Частота опитування: не більше 1 Гц
- Розміри 15.5 x 12 x 5.5 мм

#### Виводи:

1. VCC (3-5в живлення)
2. Data Out - вивід даних
3. NC - не використовується
4. Загальний

Рисунок 2.10 – Основні характеристики датчика DTH 11

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Для підрахунку велокомп'ютером швидкості, а також каденса, так само необхідні будь-які датчики, які здатні фіксувати один оборот фізичного тіла, тобто колеса та педалей. Для цього ідеально підійде Датчик Холла.

Датчик Холла – це прилад, що фіксує магнітне поле та його напруженість. Зовнішній вигляд датчика зображено на рисунку 2.11.

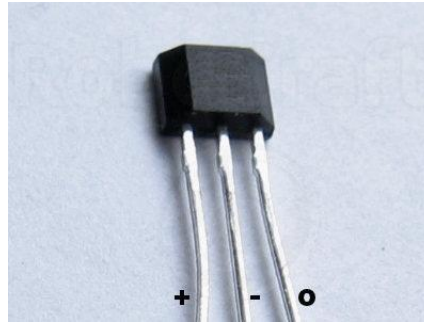


Рисунок 2.11 – Зовнішній вигляд датчика Холла

Виходячи з необхідного завдання, було обрано цифровий датчик Холла, а саме 44Е 402. Даний датчик має такі характеристики:

- напруга живлення – від 3,8 до 30 В;
- споживаний струм у режимі спокою – 3 мА;
- споживаний струм у режимі спрацьовування – 8 мА;
- тип чутливості до поля – уніполярний;
- індукція включення при 25<sup>0</sup>С- 60 Гаус;
- індукція вимкнення при 25<sup>0</sup>С - мінус 60 Гаус;
- час наростання сигналу – 1,5 мікросекунди;
- макс вихідний струм - 20мА;
- температурний діапазон - від мінус 40 до плюс 150<sup>0</sup>С;
- корпус – ТО-92.

Цифрові датчики Холла працюють дані датчики наступним чином [9]. Якщо в магнітне поле, створене магнітом, помістити пластину з струмом, що протікає через неї, то електрони в даній пластині будуть відхилятися в напрямку, перпендикулярному напрямку струму в пластині. Також слід

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

зазначити, що полярність магнітного поля важлива і впливає на те, в який саме бік відхилитимуться електрони. Завдяки різній щільності електронів на сторонах пластини буде створюватися різниця потенціалів, яку можна виміряти, цим датчики Холла і займаються.

В даному випадку, цифровий датчик Холла, видасть нам лише факт відсутності або наявності магнітного поля, що ідеально підходить для реєстрації одного обороту тіла, що обертається, з великою швидкістю. Звернемо увагу на рисунок 2.12, коли значення індукції вище за поріг спрацювання, датчик видає логічну одиницю, а коли нижче порога спрацювання, то логічний нуль. Також наявність зони нечутливості, називається гістерезисом. Вона потрібна для виключення помилкових спрацювань датчика на перешкоди.

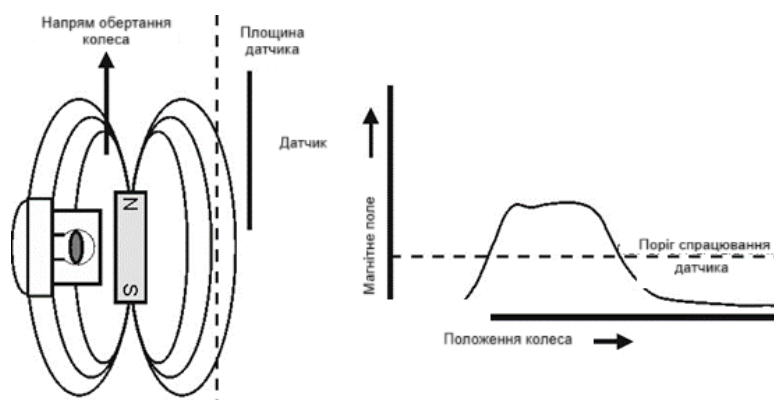


Рисунок 2.12 –Гістерезис датчика Холла

Для того, щоб візуально взаємодіяти з велокомп'ютером, а також керувати ним, необхідний дисплей з сенсором. Була обрана модель Nextion NX4827T043-011R дисплея з сенсором у зборі [10]. Основні характеристики цього дисплея наведені в додатку Б.

Зовнішній вигляд дисплея зображений рисунку 2.13, і навіть зовнішній вигляд його тильної боку із зазначенням всіх основних вузлів рисунку 2.14.





Рисунок 2.13 – Зовнішній вигляд дисплея

Дисплей Nextion складається з:

- Мікроконтролера GD32F103R8T6;
- flash пам'ять Winbond W25Q256FG (16 МБ, 100000 циклів перезапису);
- роз'єм для SD карток пам'яті;
- RAM пам'ять Winbond W9864G6KH-6 (8 МБ);
- PLD Altera MAX II EPM570T144C5N;
- драйвер резистивного дисплея ХРТ2045;
- роз'єм UART інтерфейсу зв'язку та живлення.

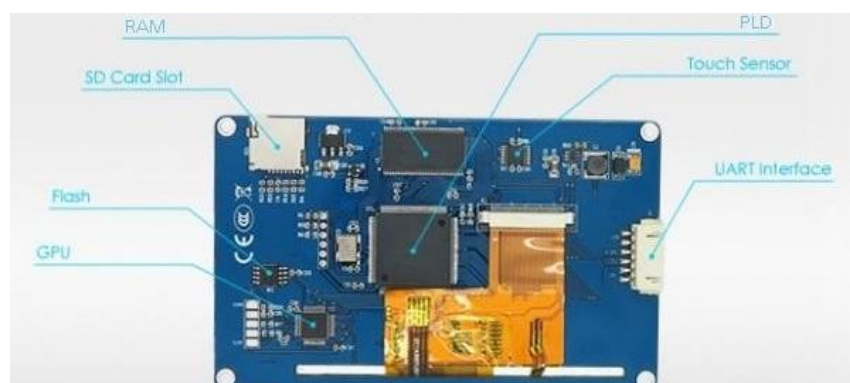


Рисунок 2.14 – Основні вузли дисплея

Даний модуль дисплея був обраний через високу продуктивність, а так само завдяки тому, що це окремий пристрій, він не навантажуватиме основний

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32



мікроконтролер. А також резистивний сенсор ідеально підходить для велокомп'ютера, тому що не реагує на воду і дотик інших предметів, тому що йому потрібно саме натискання, а не торкання.

Далі слід вибрати світлодіоди та резистори до них. Необхідно чотири світлодіоди, два на налаштування і контроль датчиків Холла і ще два для системних повідомлень велокомп'ютера: повідомлень про наближення до повної розрядки акумулятора, про повну розрядку акумулятора, про виходження в режим налаштувань та виконання алгоритму вимкнення. Розглянемо трохи докладніше системні повідомлення.

Для цього завдання найкраще підійде один RGB світлодіод, і плюс до нього другий одиничний світлодіод. Другий світлодіод для того, щоб була можливість безперервного візуального контролю за двома датчиками Холла, тому що в RGB світлодіоді всього три кристали.

В якості RGB світлодіода була обрана модель FYL -5019 RGBW1C. В якості синього світлодіода була обрана модель FYL -5002 UBC1F.

Також були розраховані і підібрані інші компоненти схеми для забезпечення роботи велокомп'ютера.

### 2.3 Обґрунтування вибору програмного забезпечення

Для виконання основної та обчислювальної роботи пристрою було обрано мікроконтролер, відповідно для нього необхідно написати програму. Для того щоб написати програму, було розроблено блок-схему алгоритму програми наведену в додатку В.

#### 2.3.1 Розробка програми велокомп'ютера

Далі на основі розробленої схеми алгоритму роботи тепер можна написати програму для мікроконтролера. Повна програма для велокомп'ютера з коментарями та поясненнями вказана у додатку Г.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Як відомо, середовище розробки Arduino побудовано на бібліотеках мовою програмування C ++.

Так як у схемі є модулі, які працюють спільно з бібліотеками, від початку програми підключаються використовувані бібліотеки рисунок 2.15.

```
# include < EEPROM. h > // Бібліотека для роботи з внутрішньою
пам'яттю ардуїно
# include " DHT. h " // Бібліотека для роботи з датчиками
температури
# include < SoftwareSerial. h > // Бібліотека для програмного
створення послідовного порту, на потрібних цифрових висновках
# include < Nextion. h > // Бібліотека для роботи з дисплеєм
```

Рисунок 2.15 – Підключення використовуваних бібліотек

Після цього оголошуються всі змінні до роботи у програмі. З усіма змінними можна ознайомитись у додатку Г. Далі слідує цикл setup, в якому встановлюються всі початкові значення та виконуються перші команди. Цей алгоритм виконується один раз при кожному включенні мікроконтролера. У цьому циклі встановлюється швидкість передачі послідовного порту для роботи з дисплеєм.

Після чого підключається переривання виходів, до яких підключені датчики Холла. Це необхідно, щоб у будь-який час виконання будь-якої частини коду програми проводився підрахунок швидкості та відстані.

Також з осередків пам'яті мікроконтролера зчитуються збережені значення довжини кола колеса, необхідних для правильного підрахунку швидкості і відстані.

А також запускається датчик температури та вологості. Так як цей цикл дуже об'ємний, з ним можна ознайомитись у додатку Г.

Після чого слідує цикл підрахунку швидкості void sens \_ a та цикл підрахунку каденсу педалей void sens \_ b. Лістинг наведено на рисунку 2.16.

Швидкість – векторна фізична величина, що характеризує швидкість переміщення та напрямок руху матеріальної точки щодо обраної системи відліку; за визначенням, дорівнює похідній радіус-вектора точки часу.

```

void sens _ a () // Зробимо підрахунок поточної швидкості та
добового пробігу
{
    if ( millis ()- lastturn > 80) // Захист від випадкових
вимірів
    {
        SPEED = w _ length /(( float )( millis ()- lastturn
)/1000) * 3.6; // Розрахунок швидкості, км/год lastturn = millis
()); // Запам'ятовуємо час останнього обороту
        DIST = DIST + w _ length / 1000; // Додаємо довжину колеса
до дистанції при кожному його обороті (ділимо на 1000 для
переведення метрів у кілометри)
        flag _ sum = 1; // Привласнюємо змінної значення 1
        flag _ vk = 1; // Привласнюємо змінної значення 1
    }
}
void sens _ b () // Зробимо підрахунок поточних обертів
педалей {
    RPM = 60/(( float )( micros ()- lastflash )/1000000); //
Розрахунок оборотів, об/хв
    lastflash = micros (); // Запам'ятовуємо час останнього оберту
}

```

Рисунок 2.16 - Лістинг циклу підрахунку швидкості та цикл підрахунку каденсу педалей

За завданням необхідно підраховувати швидкість руху велосипеда, для цього використовували формулу (2.1)

$$V = L/T, \quad (2.1)$$

де  $V$  – швидкість, км/год;  $L$  - Довжина кола колеса, см;  $T$  - період (час одного обороту), с.

Також крім швидкості розраховується і пробіг велосипеда, оскільки довжина кола колеса відома. Для цього оголошуємо змінну при кожному повному обороті колеса і додаємо до неї значення довжини кола колеса.

Крім швидкості, також необхідно підраховувати каденс педалей, це дуже актуально при поїздках на велосипеді. Каденс це кількість обертів педалей за хвилину.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Кількість обертів за хвилину можна визначити за формулою (2.2)

$$rpm = \frac{60}{(tm - tn) \cdot 1000000}, \quad (2.2)$$

де  $tm$  час протягом якого відбувається один оборот, мікросекунд;  $tn$  минулий час за яке відбувався один оборот, мікросекунд.

Після цього йдуть функції, необхідні вимірювання напруги мікроконтроллером.

Далі починається основний цикл усієї програми loop, яка виконується в циклічному режимі нескінченно. З повною версією даного циклу можна ознайомитись у додатку Г.

Спочатку йде цикл обробки швидкості та пробігу. Виконує функції, за рахунок умови: якщо більше секунди немає спрацьовування датчика Холла, то присвоюємо змінній швидкості значення нуль. Так само, якщо велосипед зупинився і швидкість дорівнює нулю, весь пробіг, набраний за час останнього руху велосипеда, додається до повного пробігу.

Після цього йде цикл обробки обертів. Даний цикл виконаний на основі умови: якщо більше секунди не спостерігається спрацьовування датчика Холла, то привласнюємо змінній обертів значення нуль.

Після чого слідує цикл вимірювання температури і вологості з датчика. Вимірювання проводяться кожні 250 мілісекунд.

Як можна побачити з додатка Г, наступним є цикл зберігання пробігу. Цей цикл досить великий і виконує кілька функцій. Побудований він на кількох умовах.

Перша умова виконується при вимиканні велокомп'ютера. Коли запускається цикл відключення, спочатку на дисплей велокомп'ютера виводиться відповідне повідомлення по команді мікроконтролера. Потім відбувається збереження повного пробігу, а також значення добового пробігу після коми. Після чого велокомп'ютер вимикається. А відключається він,

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

встановлюючи логічний нуль на виході мікроконтролера, що відповідає за відкриття та закриття транзисторного ключа, що комутує напругу від акумулятора до перетворювача напруги.

Також є умова в даному циклі, яка створена з метою продовження часу життя комірок пам'яті мікроконтролера. Ця умова спрацьовує через кожні 500 км.

Третя ж умова створена з метою, коли велокомп'ютер працює з останніми комірками пам'яті і за умовою необхідно знову перейти на наступні нові комірки, ця умова, повертає перші початкові комірки, після чого все знову повторюватиметься.

Наступним великим циклом є той, який відповідає за вибір довжини кола колеса. У цьому циклі містяться значення довжини кола коліс для всіх поширених різновидів. Реалізовано все на багатьох умовах, кожна з яких містить певне значення довжини кола колеса та інформацію про це, відправляючи її на дисплей велокомп'ютера. Керується ця умова віртуальними кнопками з екрана, одна з них веде вгору за значеннями, інша вниз. Після того, як потрібне значення вибрано, при натисканні на третю віртуальну кнопку, це значення записується в комірку пам'яті мікроконтролера і на дисплей відправляється повідомлення ok, повідомляючи про те, що значення успішно збережено.

Далі можна спостерігати цикл скидання пробігу. Цей цикл влаштований таким чином, що при натисканні на віртуальні кнопки, скидається або добовий або повний пробіг, на вибір. Цей цикл побудований за умови, якщо натиснута кнопка скидання повного пробігу і підтвердження, тоді змінним зберігання значень пробігів обнуляються.

Але якщо натиснути на скидання добового пробігу та підтвердження, то обнулюється, тільки значення добового пробігу. По суті, це лише потрібно, щоб не перезапускати велокомп'ютер.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі йде цикл виміру напруги. Напруга з акумулятора подається на аналоговий вхід, де оцифровується, порівнюється з внутрішньою опорною напругою контролера 1.1 вольта і після математичних обчислень отримуємо вимірне значення. Коли умова стає вірною, що значення наближується до 3,2 В. RGB світлодіод у схемі починає моргати помаранчевим кольором попереджаючи про розряд. При значенні 3 В - 10 разів моргає червоним кольором спільно з помаранчевим і після 10 разів відключає велокомп'ютер, автоматично.

Так як у схемі є кнопка, то і повинна бути можливість управління за допомогою неї. Так і було зроблено в даному пристрої, оскільки наступним йде цикл управління з кнопки та вимкнення. Ця кнопка спочатку має функцію включення велокомп'ютера, шляхом замикання контактів транзисторного ключа. Але коли пристрій вже увімкнено, ця кнопка виконує дві функції, це вимкнення пристрою, а також перехід у режим внесення налаштувань. Через те, що у кнопки всього один вільний контакт, який можна задіяти, а функції, які необхідно задавати цією кнопкою дві, було прийнято рішення зробити все на декількох умовах.

Ми затискаємо кнопку, на вході мікроконтролера з'являється логічна одиниця, виконується умова, за якої змінна, створена для управління іншими умовами, починає збільшуватися. Навіщо це зроблено. Значення змінної один, ми затиснули кнопку, після чого виконується функція додавання, тобто у змінної стає значення два, відбувається секундна пауза, потім три і так далі. Коли значення доходить до трьох, одна з умов стає вірною, після чого починає світитися RGB світлодіод, що говорить про те, що велокомп'ютер вмикається в режим налаштування, і він увімкнеться, якщо відпустити кнопку, тоді почне виконуватися інша умова, про яку нижче. Якщо умова почала виконуватися, значення для даної змінної присвоюється значення десять і утримуючи кнопку, ми доводимо його до тринадцяти і тоді спрацьовує умова відключення режиму налаштування та обнулення змінної.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Але якщо кнопку не відпустити, то цикл виконання продовжиться, і на значенні п'ять спрацює друга умова, RGB світлодіод засвітиться червоним кольором і це означатиме, що виконується друга функція, покладена на дану кнопку - відключення велокомп'ютера.

Після цього видно передостанній цикл програми, який називається цикл діагностики. Називається він так, тому що створений для того, щоб візуально спостерігати за роботою датчиків Холла. А спостерігати за ними необхідно для їх встановлення чи заміни або якщо необхідно переконатися у їхній працездатності. Коли увімкнено режим налаштування велокомп'ютера, потрібно натиснути на віртуальну кнопку на дисплеї, тоді запусниться умова, яка надішле на дисплей повідомлення on, що говорить про те, що ця функція увімкнулась і виконається наступна умова, яку вимкне зелений світлодіод. Коли на входах, до яких підключені датчики Холла з'являтиметься логічний нуль, світлодіоди включатимуться і коли логічна одиниця, вони вимикатимуться. Для вимикнення необхідно також натиснути на віртуальну кнопку, розташовану на дисплеї і на дисплей надішлеться повідомлення off, що повідомляє нам, що дана функція відключилася, а також спрацює перша умова, яка включить зелений світлодіод.

І останній цикл у цій програмі, це цикл надсилання інформації на дисплей. Тут все гранично просто, перша умова служить для того, щоб при русі велосипеда на дисплей відправлялися тільки потрібні в даний момент значення. Це добовий пробіг, швидкість і каденс, при зупинці спрацює друга умова, тобто на дисплей відправляється значення швидкості і каденса що дорівнює нулю і останнє актуальне значення добового пробігу і автоматично переходить на третю умову, яка при стоянці відправляє інші параметри, що залишилися у велокомп'ютера. Так само коли вмикається, режим налаштування велокомп'ютера жодна з цих умов не виконується і ніяких значень у послідовний порт не надсилається.

На цьому розробка програми для контролера закінчується. Повноцінний код програми для мікроконтролера показаний у додатку Г.

Однак, ще необхідно створити програму для дисплея. У спеціальній програмі розробки Nextion Editor створюється робочий екран пристрою, на які з обраного мене поміщаються блоки, а саме кнопки, вікно для тексту і є інші. Кожному блоку надаються унікальне ім'я в межах проекту, вводяться налаштування, це текст і тип шрифту. З блоком можна працювати з Ардуїно. Приклад програми для розробки показаний на рисунку 2.17.

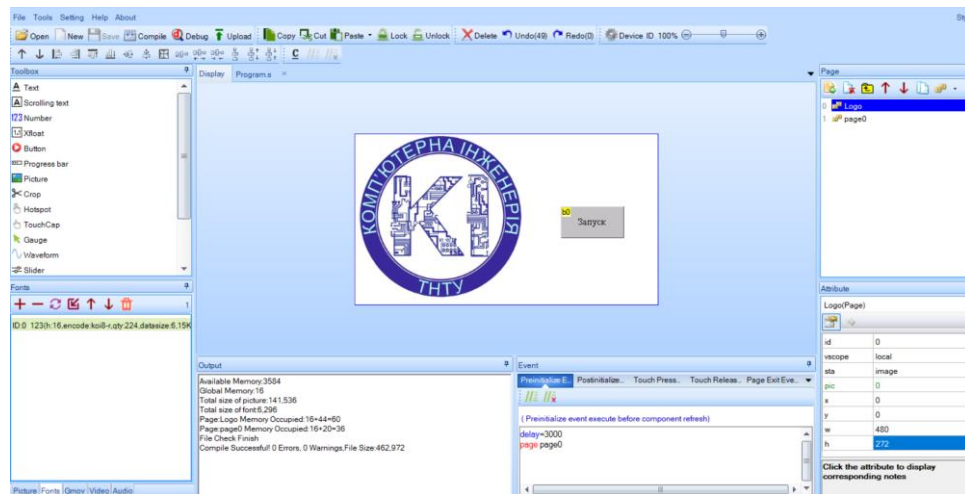


Рисунок 2.17 – Приклад виду вікна програми Nextion Editor .

Під час роботи в програмі, було розроблено чотири сторінки для дисплея, перша сторінка зображена на рисунку 2.17. Ця сторінка є початковою, яка вітає користувача при включенні пристрою, після того, як буде натиснута кнопка «Запуск», велокомп'ютер більше не повернеться на цю сторінку, поки його знову не ввімкнути.



## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Апаратна частина

Після того, як була розроблена схема електрична принципова, а так само був проведений вибір та розрахунок елементів для пристрою, далі необхідно розробити друковану плату.

Було прийнято рішення розробити друковану плату так само, як електричну принципову в пакеті Dip Trace, а саме у програмі PCB Layout. Після того, як схема електрична принципова з програми Schematic готова, схема зображена на рисунку 3.1.

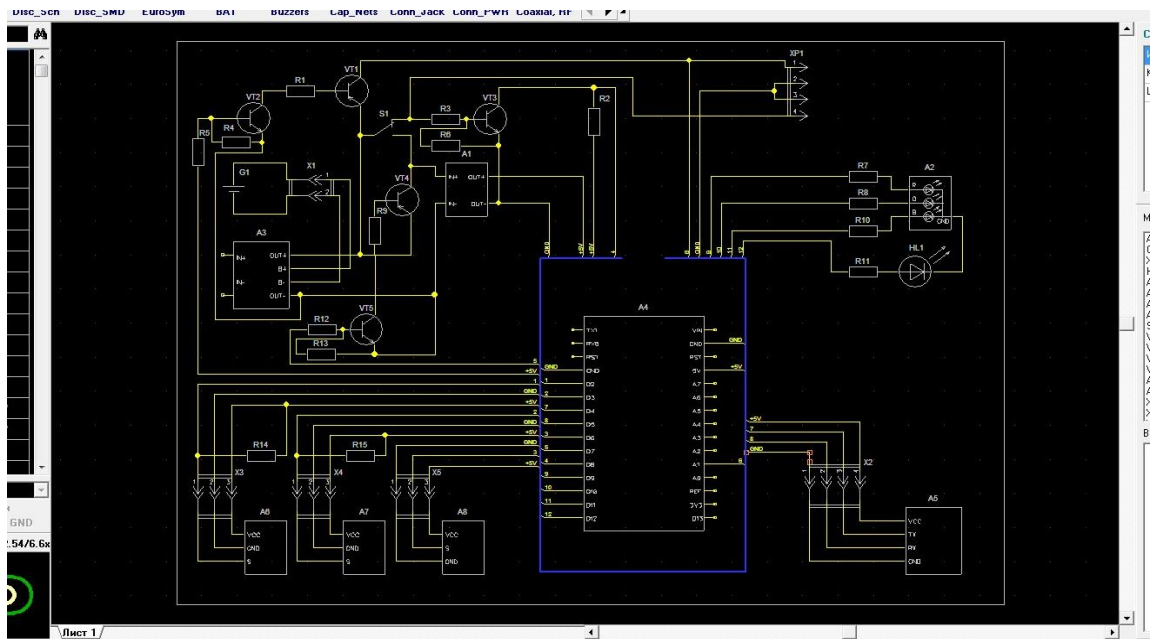


Рисунок 3.1 – Готова схема для компіляції на друковану плату

Необхідно зайти в меню і скористатися функцією, перетворити на плату. Після чого Dip Trace автоматично відкриває програму Schematic з корпусами радіоелементів та встановленими зв'язками між ними згідно з намальованою

					<b>КС КРБ 123.312.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Грухін М.В.			<b>Практична частина</b>	Літ.	Арк.	Аркушів
Перевір.		Жаровський					41	
Реценз.						<b>ТНТУ, каф. КС, гр. СІс-42</b>		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

схемою електричною принциповою. Малюємо необхідних розмірів межу друкованої плати і розставляємо всередині неї корпуси радіоелементів. Так виходить база до трасування друкованої плати, межі друкованої плати та корпуси зі зв'язками для велокомп'ютера зображені рисунку 3.2.

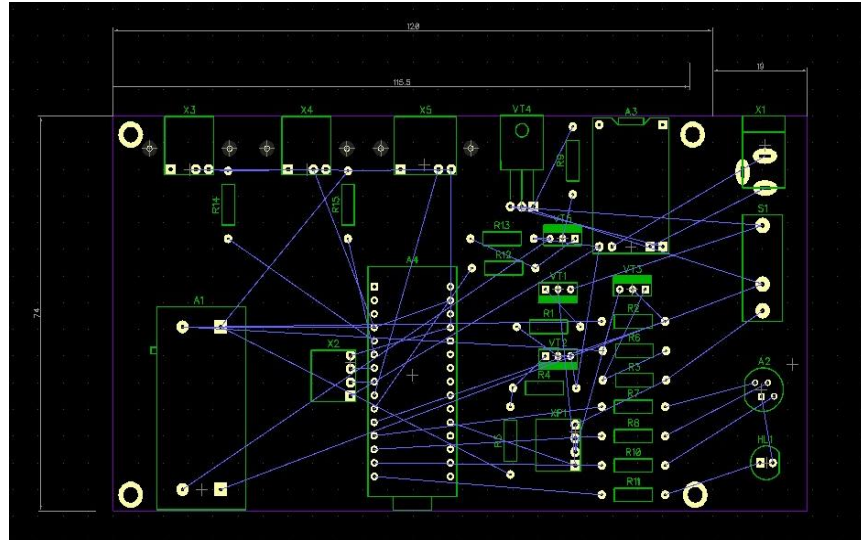


Рисунок 3.2 – Зображення меж друкованої плати та зв'язку радіоелементів

Далі необхідно запусити функцію перетворити на плату, після чого буде вже друкована плата, так як всі зв'язки перетворюються на доріжки автоматично. На рисунку 3.3 зображена страсована друкована плата.

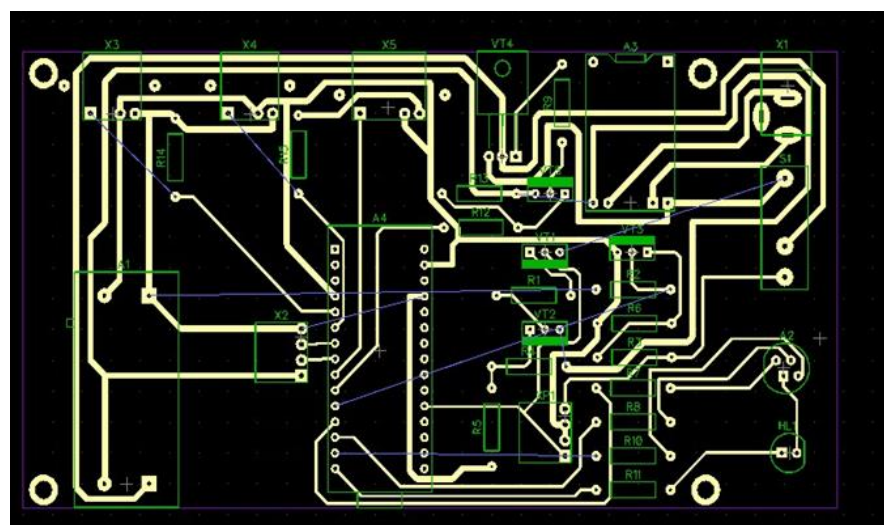


Рисунок 3.3 – Страсована друкована плата

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Потім необхідно визначити ширину доріжок для того, щоб виключити просідання по напрузі і струму. Для цього скористаємось онлайн калькулятором розрахунок ширини доріжок друкованої плати.

Відповідно до отриманих результатів, необхідно відредагувати друковану плату. Збільшити ширину доріжок, а також правильно спозиціонувати доріжки для отримання правильного зазору між доріжками та контактами радіоелементів. Так само вільні ділянки друкованої плати необхідно металізувати, для зменшення часу, що витрачається на виготовлення друкованих плат. Результат відредагованої друкованої плати зображено рисунку 3.4.

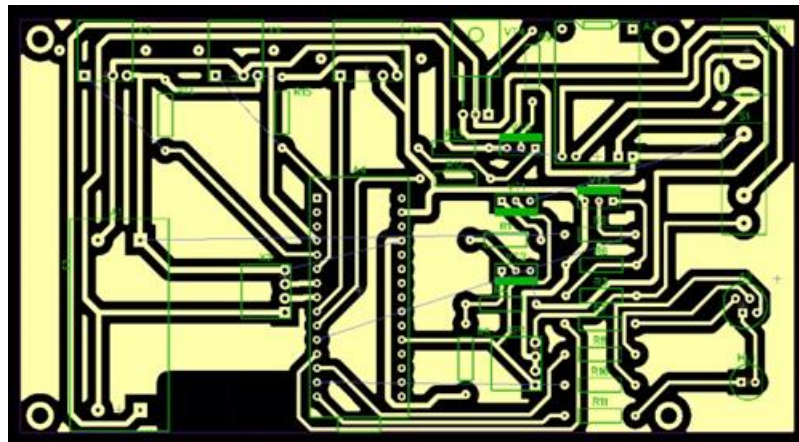


Рисунок 3.4 – Розроблена друкована плата.

В результаті виконаної роботи було отримано друковану плату з розмірами 139x74 мм. Ширина сигнальних доріжок становить 0,7 міліметра, а ширина доріжок живлення в 1 міліметр. Підсумкова ширина доріжок виконана ширше із запасом, для більш надійної роботи пристрою. На платі не вдалося уникнути перемичок і загальна їх кількість складає 8 штук.

Креслення друкованої плати наведено на рисунку 3.5.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

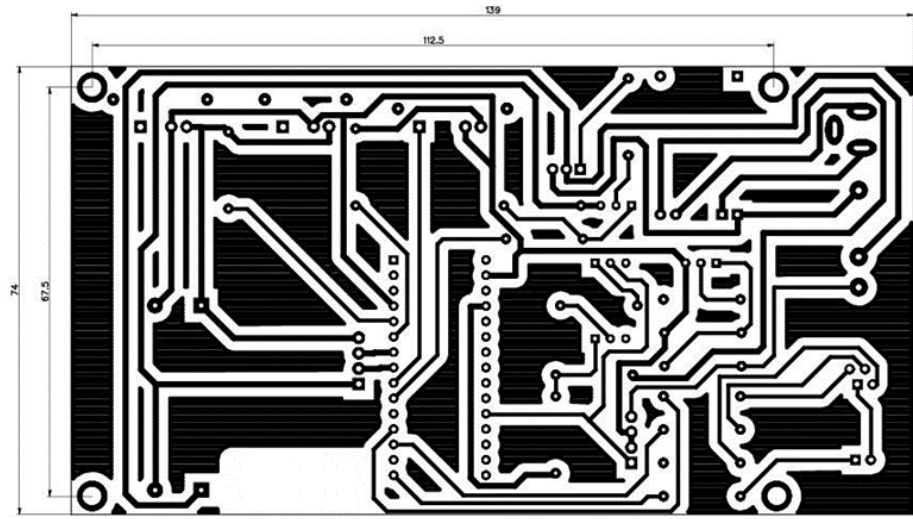


Рисунок 3.5 – Розроблена друкована плата

### 3.2 Налагодження та експериментальні дослідження

Перед тим як було виготовлено велокомп'ютер, спочатку збирався макет, на якому відпрацьовувалася схема та програма пристрою. Загальний вигляд макета зображений на рисунку 3.6.

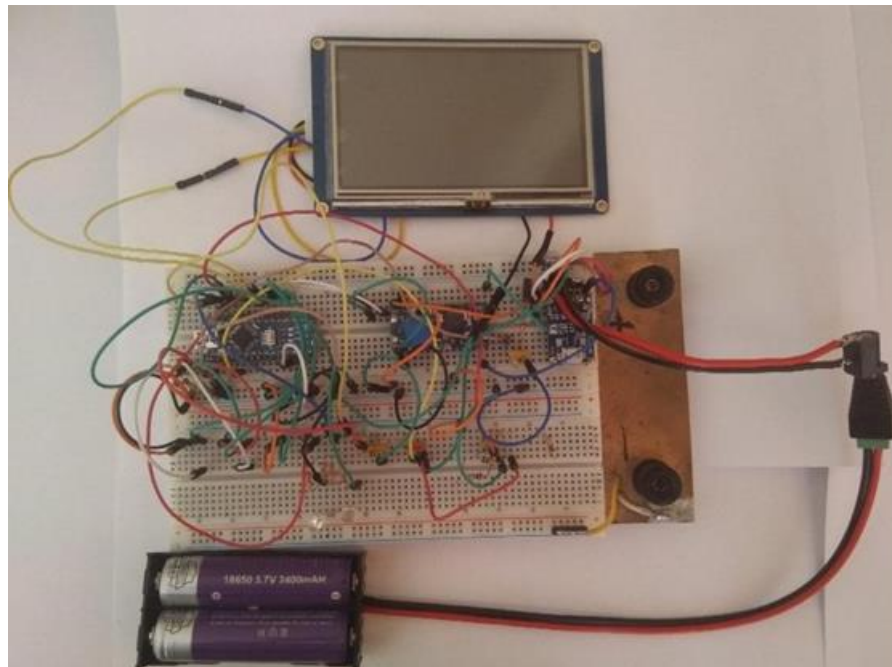


Рисунок 3.6 – Загальний вигляд макету пристрою

У ході роботи з макетом було виявлено та усунуто такі недоліки. При відключенні датчиків Холла через наведення виводилися випадкові значення швидкості та каденсу. Для вирішення цієї проблеми, на вході мікроконтролера до яких підключаються датчики Холла, були встановлені підтягуючі резистори на 10 кОм.

Коли датчик Холла спрацьовує, на його виході з'являється мінус живлення, відповідно на входах з'являється логічний нуль, після чого і запускається алгоритм, а резистори виключають коротке замикання живлення схеми.

Для вимірювання напруги акумуляторів, вони були безпосередньо підключені до аналогового входу мікроконтролера. однак при спробі вимкнути пристрій, він продовжував споживати напругу через аналоговий вхід і мікроконтролер не відключався. Тому в схему був включений транзисторний ключ, щоб при відключенні живлення, він відключав акумулятори від аналогового входу.

Також у велокомп'ютері є налаштування датчиків Холла, а точніше візуальний контроль за їх спрацюванням при їх установці або заміні. Спочатку цю функцію виконувала мікросхема здвоєного операційного підсилювача. Дана мікросхема працювала в режимі компаратора, так само для її функціонування використовувався діляк напруги з двох резисторів, а також керамічний конденсатор як фільтр живлення мікросхеми, на виходах операційних підсилювачів два світлодіоди з струмообмежувальними резисторами.

Надалі стало зрозуміло, що на мікроконтролері залишаються вільні ресурси для виконання цього завдання. У результаті кілька рядків коду дозволило позбавитися частини радіоелементів, залишивши лише два світлодіоди і струмообмежуючі резистори до них.

Також після складання пристрою, необхідно було провести налагодження для його подальшого функціонування. Необхідно включити велокомп'ютер та підключити двоканальний вольтметр або одноканальний по

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

черзі спочатку до одного входу роз'єму, потім до другого. Після чого необхідно отримати різницю між вхідною та вихідною напругою та внести її в програму велокомп'ютера.

Подальші налаштування повинні проводитися вже власником велокомп'ютера, а саме, введення значення радіуса колеса, це необхідно для коректного підрахунку швидкості та пройденої відстані велокомп'ютером. У цьому нічого складного.

Почнемо з того, що будь-які маніпуляції в налаштуваннях велокомп'ютера слід проводити затиснувши на кілька секунд кнопку включення велокомп'ютера, поки не почне світитися зелений світлодіод, потім кнопку потрібно відпустити і проводити необхідні дії. Якщо кнопку не відпустити, то зелений світлодіод перестане світитися і почне світитися червоний сигнал про те, що виконується алгоритм збереження пробігу і вимикання велокомп'ютера.

Після всіх маніпуляцій слід знову затиснути на кілька секунд кнопку включення і зелений світлодіод перестане світитися, сигналізуючи про те, що велокомп'ютер вийшов з режиму налаштувань і знову готовий працювати.

Введення правильного радіуса колеса дуже просте. На дисплеї розташовуються три віртуальні кнопки, одна відповідає за перемотування вперед, а друга за перемотування назад. Значення радіуса вибирається з вже готових найпоширеніших значень. Після того, як ви вибрали потрібний радіус, він буде написаний між кнопками, необхідно натиснути на третю кнопку, яка зберігає це значення і записується в незалежну пам'ять. Якщо ви натиснули випадково на вибір радіусу, то поки ви не натиснете кнопку зберегти, нічого не зміниться.

Так само в майбутньому, коли накопичиться певні значення пробігу, його можна скинути, як повний, так і добовий, щоправда, якщо вимкнути та включити пристрій добовий пробіг скидається автоматично, але значення після коми не скидається, для збереження повного актуального пробігу, а ця кнопка дозволяє скинути і це.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		



На дисплеї в середині також є три віртуальні кнопки, вона скидає добовий, друга повний пробіг. Третя кнопка посередині підтверджує, що ви скидаєте пробіг, це зроблено так само як захист від випадкових натискань.

І нарешті, функція візуально контролю над спрацюванням датчиків Холла. Це необхідно при встановленні та заміні датчиків, а також якщо є необхідність переконатися в їх працездатності. У правій частині дисплея знаходиться віртуальна кнопка для увімкнення та вимкнення цієї функції, а нижче напис off, це говорить про те, що ця функція вимкнена. При натисканні на кнопку з'явиться напис on, що говорить про те, що дана функція ввімкнулася, а так само почнуть світитися два синіх світлодіоди, що також дає розуміння, що ця функція активна. При спрацюванні одного або двох датчиків холу світлодіоди будуть відключатися, можна безперервно стежити відразу за двома датчиками холу. Після того, як мета досягнута, необхідно вимкнути цю функцію.

На цьому налагодження та налаштування велокомп'ютера закінчується.

### 3.3 Виготовлення велокомп'ютера

Після того, як розроблена електрична плата, розраховані та обрані елементи схеми, а також розроблена друкована плата та написана програма для роботи пристрою, необхідно зібрати велокомп'ютер.

Спочатку для складання необхідно виготовити друковану плату. Для друкованої плати велокомп'ютера був обраний фольгований склотекстоліт товщиною 1,5 міліметра і товщиною мідної фольги 35 мікрметрів, один із найпоширеніших і найдоступніших значень фольгованого склотекстоліту.

Далі під час збоки був підключений дисплей до велокомп'ютера через відповідний роз'єм на платі. А також найвищі частини схеми були заклеєні термостійким скотчем для уникнення коротких замикань на платі дисплея, а також небажаного тертя (рис. 3.7-3.8).

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

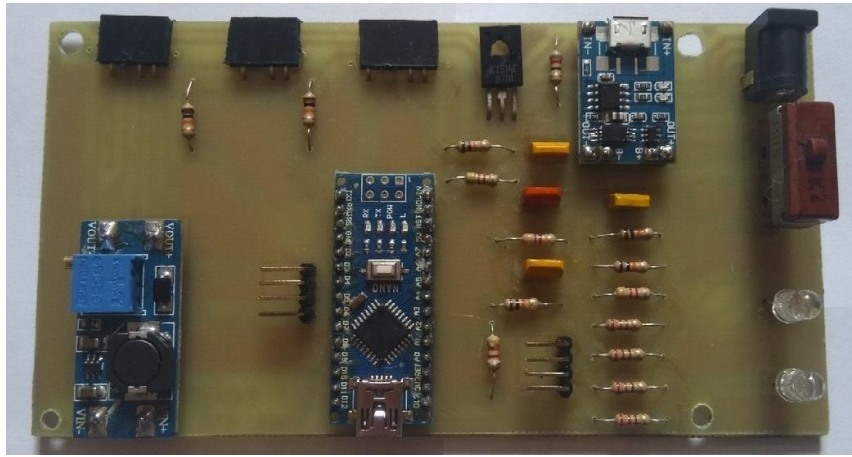


Рисунок 3.7 – Готова друкована плата з встановленими елементами

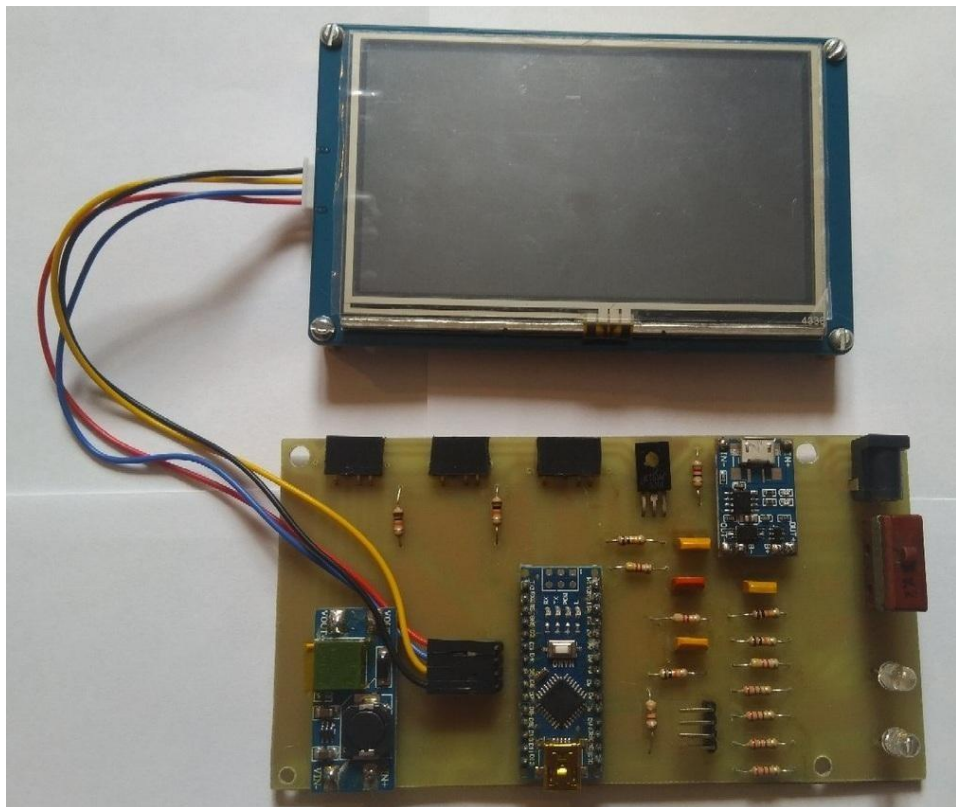


Рисунок 3.8 – Підключення дисплея до відповідного гнізда

Після чого здійснено кінцеве складання велокомп'ютера шляхом жорсткого кріплення дисплея до друкованої плати.

Дане кріплення було виконане за допомогою латунних стійок та коротких болтів М3. Повністю зібраний велокомп'ютер зображений на рисунках 3.9-3.10.

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.312.00.00 ПЗ

Арк.

48





Рисунок 3.9 – Зібраний велокомп'ютер вид зверху



Рисунок 3.10 – Зібраний велокомп'ютер вид збоку

Після самого велокомп'ютера була зібрана вся периферія, а саме розпаяні датчики Холла, датчик температури, а також утримувач акумуляторів типорозміру 18650 з роз'ємом для підключення до велокомп'ютера.

Датчики Холла встановлені на маленькій платі зі світловою індикацією роботи. До них припаяний роз'єм виготовлений з планки PLS, з якої був видалений один вихід, щоб даний датчик не можна було вставити в інші роз'єми.

Підключення роз'єму до датчика Холла виконано гнучким багатожильним проводом типу МГТФ перетином 0,12 мм<sup>2</sup>. Готовий датчик Холла з роз'ємом зображений на рисунках 3.11 та 3.12.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

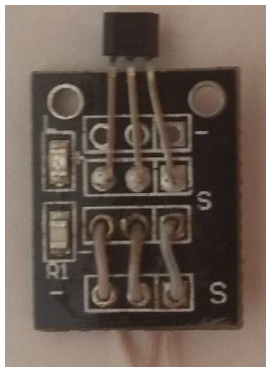


Рисунок 3.11 – Встановлений на плату датчик Холла

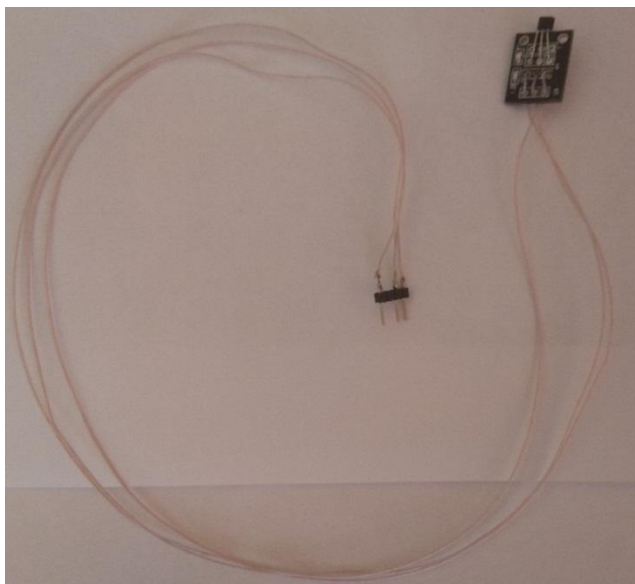


Рисунок 3.12 – Готовий датчик Холла з роз'ємом

Датчик температури підключений до роз'єму виготовленої з планки PLS, з якої було видалено два виходи, щоб даний датчик не можна було вставити в інші роз'єми.

Підключення роз'єму до датчика температури виконано гнучким багатожильним проводом типу МГТФ перетином  $0,12 \text{ мм}^2$ . Також сигнальний вихід підключено до живлення через резистор  $10 \text{ кОм}$  для уникнення хибних показань і перешкод. Цей резистор припаяний прямо на датчик. Готовий датчик температури з роз'ємом зображений на рисунках 3.13 та 3.14.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.13 – Датчик температури із встановленим резистором

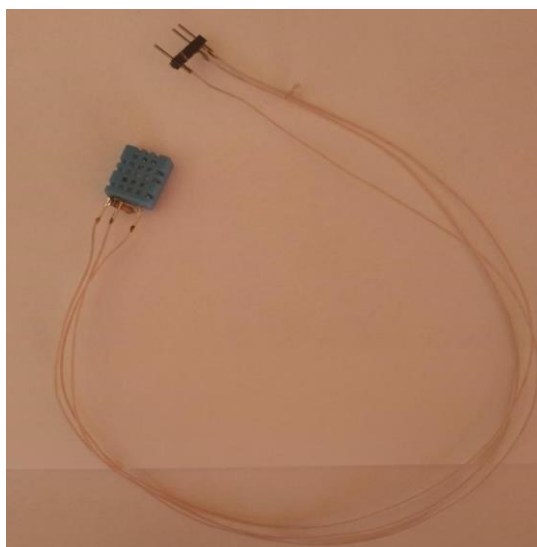


Рисунок 3.14 – Готовий датчик з роз'ємом

Для підключення живлення до велокомп'ютера обраного тримач для акумуляторів був припаяний провід типу PSC (01-6105-6) перетином 1 мм<sup>2</sup>, а також підключений роз'єм JR -55 з клемною колодкою. Готовий тримач акумуляторів з роз'ємом зображений на рисунку 3.15. Повністю зібраний прототип велокомп'ютера показано на рисунку 3.16.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

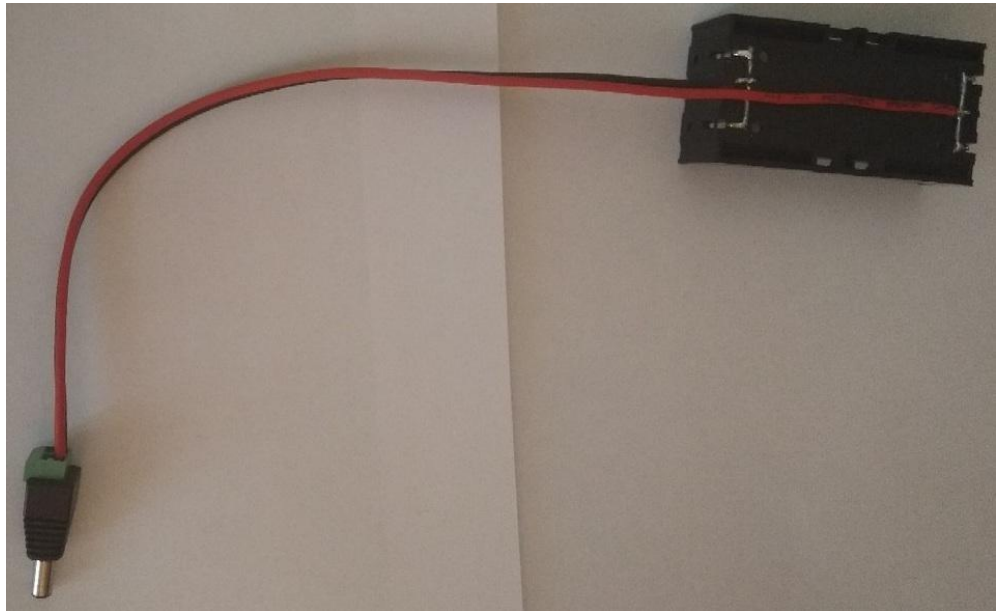


Рисунок 3.15 – Готовий утримувач акумуляторів із роз'ємом

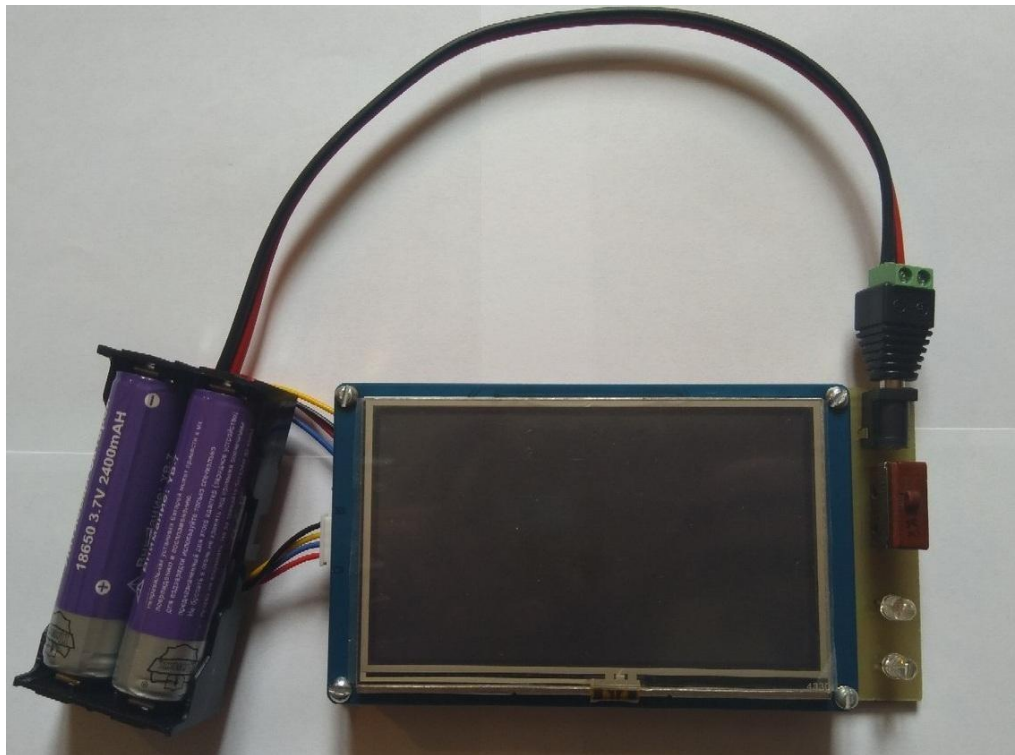


Рисунок 3.16 – Прототип велокомп'ютера із підключеними акумуляторами

Таким чином було розроблено велокомп'ютер в частині головного програмно - апараного модуля, підбрані датчики, схема живлення.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Долікарська допомога при вивихах

Вивих – різновид травм, що характеризується як порушення конфігурації суглобових поверхонь. Це вид травми, за якої суглобовий кінець однієї кістки зміщується за межі суглобової поверхні іншої, котра разом із першою утворює суглобову пару.

Також термін «вивих» і «підвивих» застосовують до деяких уражень кришталика ока. Коли відбувається повний розрив зонулярних волокон, а кришталик в результаті цього розміщений за межами зіниці – іде мова про вивих кришталика. Як відбувається частковий розрив зонулярних волокон, а кришталик децентралізований, але частково розміщений у межах зіниці, то тоді це трактують як підвивих кришталика.

Вивихи за часом поділяються на [13]:

- свіжі – до 3-х діб після травми;
- несвіжі – від 3-х діб до 3-х тижнів;
- застарілі – більше 3-х тижнів.

Також вивихи за особливістю поділяються на:

- звичний вивих – це багаторазові вивихи без надмірних фізичних зусиль;
- невірний вивих – це, як правило, свіжий вивих, який через різні причини не вдається усунути. Причиною може бути інтерпозиція капсули, зв'язок, сухожилків

Ознаки вивиху:

- сильний біль у кінцівці у ділянці суглоба, особливо при рухах;
- значний набряк ураженого суглоба;

					<b>КС КРБ 123.312.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Грухін М.В.			Безпека життєдіяльності, основи охорони праці	Літ.	Арк.	Аркуші
Перевірів		Жаровський					53	
Консульт.		Пилипець М.І.				ТНТУ, каф. КС, гр. СІс-42		
Н. Контр.								
Затверд.		Осухівська Г.М.						

- блідність шкіри навколо суглоба;
- деформація ділянки суглоба;
- відсутність активних і неможливість пасивних рухів у суглобі;
- кінцівка зафіксована у неприродньому положенні;
- зміна довжини кінцівки (видиме укорочення або подовження).

Порядок надання домедичної допомоги при вивихах:

- при наявності кровотечі із судин – зупинка кровотечі джгутом чи імпровізованими джгутами (закрутка, перетягання ременем), накладання стискальної пов'язки;
- при відкритому вивиху накладання первинної (асептичної) пов'язки з метою профілактики вторинного мікробного забруднення;
- транспортна іммобілізація: аутоіммобілізація, підручними засобами.

Задачі транспортної іммобілізації:

- кінцівка зафіксована у неприродньому положенні;
- зупинка кровотечі (артеріальної);
- попередження травматичного шоку;
- накладання стерильної пов'язки на рану;
- проведення іммобілізації табельними або підручними засобами.

Основні принципи транспортної іммобілізації:

- а) для знерухомлення ушкодженої кінцівки необхідно іммобілізувати два суміжних суглоби, а при вивиху плечової або стегнової кістки три суглоби;
- б) моделювання шини слід проводити по здоровій кінцівці, або на тому, хто її буде накладати, тобто на медпрацівнику;
- в) при іммобілізації кінцівки необхідно надати їй фізіологічне положення, а якщо це неможливо, то таке положення, при якому кінцівка менш за все травмується;
- г) при відкритих вивихах вправлення не виконують, накладають стерильну пов'язку і кінцівку фіксують в тому положенні, в якому вона знаходилась в момент ушкодження;

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

- д) іммобілізацію накладають поверх одягу і взуття потерпілого, між шиною і кінцівкою потерпілого кладуть м'яку ватно-марлеву підкладку;
- е) при відкритих вивихах на рану слід накласти стерильну пов'язку;
- ж) при фіксації шини не можна закрити місце накладання джгута, щоб була можливість коригувати стан джгута;
- з) іммобілізована кінцівка перед транспортуванням в холодну пору року повинна бути обов'язково утеплена з метою профілактики відмороження.
- и) під час перекладання потерпілого з нош, ушкоджену кінцівку повинен тримати помічник.

При використанні веловелокомп'ютера під час руху необхідно не відволікатись від дороги. У випадку вивиху необхідно вміти надавати першу долікарську допомогу.

#### 4.2 Проведення інструктажів з охорони праці

Інструктажі з питань охорони праці проводяться на всіх підприємствах, установах і організаціях незалежно від характеру їх трудової діяльності, підлеглості і форми власності. Мета інструктажу - навчити працівника правильно і безпечно для себе і навколишнього середовища виконувати свої трудові обов'язки [14].

Інструктажі за часом і характером проведення поділяють на:

- а) вступний інструктаж проводиться з усіма працівниками, які щойно прийняті на роботу (постійну або тимчасову), незалежно від їх освіти, стажу роботи за цією професією або посади; працівниками, які знаходяться у відрядженні на підприємстві й беруть безпосередню участь у виробничому процесі; з водіями транспортних засобів, які вперше в'їжджають на територію підприємства; учнями, вихованцями та студентами навчально-виховних закладів перед початком трудового й професійного навчання в лабораторіях, майстернях на полігонах.

Вступний інструктаж проводить спеціаліст відділу охорони праці або особа, що призначена наказом для проведення цієї роботи.

Запис про проведення вступного інструктажу робиться в спеціальному журналі, а також у документі про прийняття працівника на роботу, де розписуються інструктуючий та проінструктований працівники;

б) первинний інструктаж проводиться на робочому місці до початку роботи з новоприйнятим працівником або працівником, який буде виконувати нову для нього роботу, студентом, учнем та вихованцем перед роботою в майстернях, лабораторіях, дільницях тощо. Первинний інструктаж проводиться індивідуально або для групи осіб спільного фаху за програмою, складеною з урахуванням вимог відповідних інструкцій з охорони праці та інших нормативних актів про охорону праці, технічної документації і орієнтованого переліку питань первинного інструктажу, викладених в додатку до Типового положення про навчання, інструктаж та перевірку знань з питань охорони праці. Програма первинного інструктажу розробляється керівником цеху чи дільниці, узгоджується зі службою охорони праці і затверджується роботодавцем, керівником навчального закладу або відповідного структурного підрозділу.

Усі працівники після первинного інструктажу на робочому місці проходять стажування протягом 2-15 змін під керівництвом досвідчених кваліфікованих робітників або спеціалістів, що призначаються наказом (розпорядженням) по підприємству, цеху, дільниці, виробництву. В окремих випадках стажування може не призначатися, якщо робітник має стаж роботи за своєю професією не менше трьох років, а робота, яку він виконуватиме, для нього знайома з попереднього місця праці;

в) повторний інструктаж проводиться на робочому місці з усіма працівниками: на роботах із підвищеною небезпекою - один раз на квартал; на інших роботах - один раз у півріччя. Мета інструктажу - поновити знання та уміння виконувати працівником роботу правильно і безпечно. Проводиться

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		



інструктаж індивідуально або для групи працівників, що виконують однотипні роботи, за програмою первинного інструктажу в повному обсязі;

г) позаплановий інструктаж проводиться з працівниками на робочому місці або в кабінеті охорони праці у таких випадках:

- при введенні в дію нових або змінених нормативних актів про охорону праці;

- при зміні технологічного процесу, заміні або модернізації устаткування, приладів та інструментів, вихідної сировини, матеріалів та інших факторів, що впливають на охорону праці;

- при порушенні працівником нормативних актів, що може призвести до травми, отруєння або аварії;

- на вимогу працівника органу державного нагляду або вищої за ієрархією державної чи господарської організації при виявленні недостатнього знання працівником безпечних прийомів праці і нормативних актів про охорону праці;

- при перерві в роботі виконавця робіт більше, ніж 30 календарних днів (для робіт із підвищеною небезпекою), а для решти робіт - більше 60 днів.

Позаплановий інструктаж проводиться індивідуально або для групи працівників спільного фаху. Обсяг і зміст інструктажу визначається для кожного окремого випадку залежно від причин і обставин, що викликали необхідність його проведення.

Цільовий інструктаж проводиться у таких випадках:

- при виконанні разових робіт, що не пов'язані безпосередньо з основними роботами працівника;

- при ліквідації наслідків аварії і стихійного лиха;

- при виконанні робіт, що оформляються нарядом-допуском, письмовим дозволом та іншими документами;

- в разі проведення екскурсій або організації масових заходів з учнями та вихованцями (екскурсії, походи, спортивні заходи).

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Цільовий інструктаж фіксується нарядом-допуском або іншим документом, що дозволяє проведення робіт.

Первинний, повторний, позаплановий та цільовий інструктажі проводить безпосередньо керівник робіт (начальник виробництва, цеху, дільниці, майстер, інструктор виробничого навчання, викладач тощо). Перевірка знань здійснюється усним опитуванням або за допомогою технічних засобів навчання, а також перевіркою навичок виконання робіт відповідно до вимог безпеки.

Первинний, повторний та позаплановий інструктажі, стажування та допуск до роботи реєструються в спеціальних журналах. При цьому обов'язкові підписи як інструктованого, так і інструктуючого. Журнали інструктажів повинні бути пронумеровані, прошнуровані і скріплені печаткою.

Працівники, що не пов'язані з обслуговуванням обладнання, використанням інструменту, збереженням сировини, матеріалів, можуть бути звільнені від первинного, повторного та позапланового інструктажу за наказом (розпорядженням) керівника підприємства по узгодженню з державним інспектором Держпраці.

Роботодавець або керівник структурного підрозділу зобов'язаний видати працівнику примірник інструкції з охорони праці за його професією або вивісити її на робочому місці.

В розділі розглянуто питання з безпеки життєдіяльності та охорони праці.

Долікарська допомога при вивихах, є важливим питанням, оскільки особи, що катаються на велосипеді, повинні вміти надавати домедичну допомогу у випадку травмування.

А також розглянуто питання проведення інструктажів з охорони праці, оскільки це питання є важливим для безпеки персоналу на підприємствах і організаціях.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У ході роботи над кваліфікаційною роботою бакалавра було розроблено велокомп'ютер на базі мікроконтролера Arduino. Даний велокомп'ютер дозволяє стежити за здоров'ям велосипедиста при тренуваннях або туристичних поїздках на велосипеді.

Під час роботи було проведено аналіз сучасних аналогів, а також у спеціальному пакеті DipTrace розроблена схема електрична принципова, друкована плата пристрою.

Відображення інформації і керування здійснюється за допомогою сенсорного дисплею.

Також було розроблено алгоритм і реалізоване програмне забезпечення в середовищі Arduino, а також виконано налагодження та доопрацювання пристрою.

Розроблений велокомп'ютер дозволяє відстежувати наступні показники:

- дані каденсу;
- пройденої відстані;
- моніторинг швидкості руху;
- добовий пробіг.

Ця робота відповідає завданню, завдання та цілі проекту досягнуто.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Посібник для початківців із каденсу: що це таке, чому це важливо та як покращити його URL: <https://velocentr.pl.ua/posibnyk-dlya-pochatkivciv-iz-kadensu-scho-ce-take-chomu-ce-vazhlyvo-ta-yak-pokraschyty-yogo> (дата звернення: 15.05.2024).
2. Як позбутися від брязкоту контактів при підключенні кнопки до Arduino - що таке брязкіт контактів URL: <https://yak.koshachek.com/articles/jak-rozbutisja-vid-brjazkotu-kontaktiv-pri.html> (дата звернення: 16.05.2024).
3. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
4. Види акумуляторів, відмінності та особливості URL: [http://www.powerinfo.com/Тип\\_акумулятора.php](http://www.powerinfo.com/Тип_акумулятора.php) (дата звернення: 20.05.2024).
5. TP 4056 Модуль для заряджання Li-ion 5V 1A URL: <https://myproject.com.ua/tp-4056-modul-dlja-zarjadzhannja-li-ion-type-c-5v-1a-ua.html> (дата звернення: 30.05.2024).
6. MT3608 Підвищувач регульований перетворювач DC-DC URL: <https://myproject.com.ua/mt3608-pidvischujuchij-regulovanij-dc-dc-modul-ua.html> (дата звернення: 30.05.2024).
7. Технічний опис Arduino Nano URL: <http://arduino.cc/Hardware/ArduinoBoardNano>. (дата звернення: 01.06.2024).
8. DHT11 DHT22 датчик вологості та температури, підключення до Arduino URL: <https://qazf.com.ua/dht11-dht22-arduino/> (дата звернення: 01.06.24).
9. Датчик Холла URL: <https://corelamps.com/elektronika/datchyk-kholla/> (дата звернення: 01.06.2024).
10. Nextion Enhanced NX4827K043 - технічний опис дисплея 4,3 дюйма, навчальні посібники URL: <https://mysku.com/blog/china-stores/44446.html> (дата звернення: 05.06.2024).

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU.Tern.: TNTU. 2024. Vol 109. No 1. P. 54–65.

12. Yatsyshyn V., Pastukh O., Zharovskyi R., Shabliy N. Software tool for productivity metrics measure of relational database management system. Mathematical Modeling. No 1 (48). 2024. p. 7-17.

13. Долікарська допомога при вивихах // Режим доступу: <https://moz.gov.ua/article/health/travmuwannja-vidi-persha-dopomoga-taporadi/> (дата звернення: 10.06.2024).

14. Дуднікова І.І. Безпека життєдіяльності. Навч. посібник. – 2-ге вид., доп. – К.: Вид-во Європ. ун-ту, 2013. 268 с.

15. Осухівська Г.М., Луцик Н.С., Тиш Є.В., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

					<b>КС КРБ 123.312.00.00 ПЗ</b>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А.  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“ \_\_\_\_ ” \_\_\_\_\_ 2024 р

ВЕЛОКОМП'ЮТЕР НА БАЗІ ARDUINO

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 6 листках

**Вид робіт:**

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

\_\_\_\_\_ к.т.н., доц. Жаровський Р.О.

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІс-42

\_\_\_\_\_ Грухін М.В.

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**Тернопіль 2024**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Велокомп'ютер на базі Arduino».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.312.00.00

### 1.2 Виконавець

Студент групи СІс-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Грухін Михайло Вікторович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ від 26 квітня 2024 року № 4/7-468 .

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 29.04.2024 р.

Плановий термін завершення виконання кваліфікаційної роботи – 24.06.2024 р.

### 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.



Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90%, наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Система що розробляється призначена для визначення статистичних даних про велопробіг.

### 2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка комп'ютеризованої системи, яка використовує датчики Холла для вимірювання даних про поїздку на велосипеді.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

#### 3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система відеонгляду повинна складатись з:

- контроллера ардуіно;
- передбачено живлення від акумулятора;

- сенсорного екрану для відображення і керування велокомп'ютером.

### 3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість.

### 3.1.3 Вимоги по діагностуванню системи

Для діагностування системи відстеження використовуються інструменти діагностування основних процесів системи, які дозволяють перевірити основні функціональні елементи системи.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

### 3.1.4 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки додаванню нових функцій і модулів.

## 3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

### 3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

### 3.3 Вимоги до безпеки

Система електроживлення повинна забезпечувати захист від перерозряду, перегріву.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск  $760 \pm 25$  мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

Загальна архітектура системи (A1)

Структурна схема системи (A1)

Схема електрична принципова (A1)

Алгоритм роботи програмного забезпечення (A1)

\*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	01.02-09.02.2024
2	Аналіз технічного завдання	05.02-11.02.2024
3	Визначення і аналіз вимог до веловелокомп'ютера	10.05-15.05.2024
4	Проектування структури веловелокомп'ютера	16.05-20.05.2024
5	Проектування програмної частини	21.05-28.05.2024
6	Проектування апаратної частини	29.05-06.06.2024
7	Налаштування і тестування веловелокомп'ютера	07.06-10.06.2024
8	Безпека життєдіяльності, основи охорони праці	10.06-15.06.2024
9	Оформлення кваліфікаційної роботи	16.06-20.06.2024
10	Попередній захист кваліфікаційної роботи	14.06.2024
11	Захист кваліфікаційної роботи	24.06-28.06.2024

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б.

Характеристики NX4827P043-011R

Nextion Models

Nextion Type	Intelligent Series
Nextion Models	NX4827P043-011R (4.3 inch resistive touchscreen without enclosure)

Specifications

	Data	Description
Color	65K 65536 colors	16 bit 565, 5R-6G-5B
Layout size	120mm(L)×74mm(W)×6.2mm(H)	
Visual Area (V.A.)	105.50mm(L)×67.20mm(W)	
Active Area (A.A.)	95.04mm(L)×53.86mm(W)	
Resolution	480×272 pixel	Also can be set as 272×480
Touch type	Resistive	
Touches	> 1 million	
Backlight	LED	
Backlight lifetime (Average)	>30,000 Hours	
Brightness	300nit	0% to 100%, the interval of adjustment is 1%
Weight	98g	

Electronic Characteristics

	Test Conditions	Min	Typical	Max	Unit
Operating Voltage		4.65	5	6.5	V
Operating Current	VCC=+5V, Brightness is 100%	–	220	–	mA
	SLEEP Mode	–	170	–	mA

Power supply recommend : **5V, 1.0A, DC**

## Working Environment & Reliability Parameter

	Test Conditions	Min	Typical	Max	Unit
Working Temperature	5V, Humidity 60%	-20	25	70	°C
Storage Temperature		-30	25	85	°C
Working Humidity	25°C	10%	60%	90%	RH

## Interfaces Performance

	Test Conditions	Min	Typical	Max	Unit
Serial Port Baudrate	Standard	2400	9600	921600	bps
Output High Voltage (TXD)	IOH=1mA	3.0	5.0	V <sub>in</sub>	V
Output Low Voltage(TXD)	IOL=-1mA		0.1	0.2	V
Input High Voltage(RXD)		3.0	5.0	V <sub>in</sub>	V
Input Low Voltage(RXD)		-0.7	0.0	1.3	V
Serial Port Mode	3.3V/5.0V TTL				
Serial Port	4Pin_2.54mm				
USB interface	NO				
SD card socket	Yes (FAT32 format), support maximum 32G Micro SD Card * presence of *.tft file on microSD: socket is exclusive to upgrade Nextion firmware/HMI design * Intelligent Series only: see Instruction Set / Editor Guide for microSD card runtime usage				
Extended IO	8 Digital extended GPIO				
	IO0-IO7 support input, output and component binding event * IO pin / ports are not exclusive, limit current draw to 1mA recommended				
	IO6-IO7 support PWM				
RTC	built-in RTC support (Battery type: CR1220)				

V<sub>in</sub>: the input voltage of power supply

## Memory Features

Memory Type	Test Conditions	Min	Typical	Max	Unit
FLASH Memory	Store fonts and images			120	MB
User Storage	EEPROM			1024	BYTE
RAM Memory	Store variables			512	KB
Instruction Buffer	Instruction Buffer			4096	BYTE

## Audio Features

Speaker	Parameter	Min	Typical	Max	Unit
Power	–	0.5	–	1.5	W
Audio Connector Type : 1.25T-2-2A (1.25mm pitch 2-pin housing)					





## Додаток Г.

### Лістинг програмного забезпечення

```
//-----Бібліотеки-----
# include < EEPROM. h > // Бібліотека для роботи з внутрішньою
    пам'яттю ардуїно
# include " DHT.h " // Бібліотека для роботи з датчиками температури
# include < SoftwareSerial. h > // Бібліотека для програмного
    створення послідовного порту
# include < Nextion. h > // Бібліотека для роботи з дисплеєм

//-----Змінні для послідовного порту та дисплея---
SoftwareSerial nextion (4, 5); // Цифровий висновок 4 як TX і 5 як RX
Nextion myNextion ( Nextion, 9600); // Швидкість передачі для
    дисплея 9600 bps

//-----Змінні для підрахунку обертів-----
unsigned long Lastflash ; // Змінна зберігання часу підрахунку
    обертів
int RPM ; // Змінна зберігання обертів як цілих чисел

//-----Змінні для підрахунку швидкості-----
unsigned long lastturn ; // Змінні зберігання часу для підрахунку
    швидкості
float SPEED ; // Змінна зберігання швидкості у вигляді десяткового
    дробу
float w _ length = 2.050; // Довжина кола колеса в метрах (початкова
    26")
int flag _ circle = 0; // Перехідне значення для вибору довжини кола

//-----Змінні для підрахунку та збереження відстані-
float DIST ; // Змінна зберігання добової відстані у вигляді
    десяткового дробу
int DIST _ T ; // Змінна зберігання сумарної відстані як цілого
    числа
int dist _ past ; // Змінна зберігання пробігу при минулій зупинці
int dist _ transit ; // Змінна для математичних процесів, перехідне
float dist _ comma ; // Змінна для зберігання значення добового
    пробігу після коми
int dist _ excess ; // Змінна зберігання добового пробігу без коми
    для математичних дій
int cell _ hi =6; // Номер комірки пам'яті для зберігання пробігу
    старшого біта
int cell _ low = 7; // Номер осередку пам'яті для зберігання пробігу
    молодшого біта
int cell _ comma = 8; // Номер осередку пам'яті для зберігання
    добового пробігу після коми
int hi _ c ; // Змінна для старшого біта значення зміни комірок (
    addr )
```

```

int low_c ; // Змінна для молодшого біта значення зміни комірок (
  addr )
int hi_d ; // Змінна для старшого біта умови зміни комірок ( dist
  _con )
int low_d ; // Змінна для молодшого біта умови зміни комірок (
  dist _con )
int dist_con = 500; // Змінна зберігання значення умови зміни
  комірок пам'яті
int addr = 0; // Значення для зміни комірок пам'яті
byte hi ; // Змінна для старшого біта сумарного пробігу
( DIST_T ) byte low ; // Змінна для молодшого біта сумарного пробігу
( DIST_T ) int flag_sum = 0; // Змінна до роботи з пробігом
int flag = 0; // Змінна для виконання умови збереження та вимкнення

//-----Змінні для скидання та збереження відстані

int reset = 0; // Змінна для умови скидання пробігу
int reset_2=0; // Змінна для умови скидання пробігу
int reset_3=0; // Змінна для умови скидання пробігу

//-----Змінні для вимірювання температури та
  вологості

DHT dht (6, DHT 11); // Прописуємо порт для роботи з датчиком
int wet = 0; // Змінна для зберігання вологості
int temp = 0; // Змінна для зберігання температури

//-----Змінні для вимірювання напруги-----

const byte debug = 0; // Для налагодження через консоль встановити
  значення 2

const float typVbg = 1.025; // Ця константа для вимірювання напруги
float Vcc = 0.0; // Змінна для зберігання вимірюваної напруги

# define A_PIN 1 // Тимчасове ім'я для константи (аналоговий вхід)
# define COUNT 5 // Тимчасове ім'я для константи

int i ; // змінна для оператора
for float curVoltage ; // Змінна для зберігання
int flag_leave = 1; // Прапор для роздільної здатності (захист
  акумулятора, відсікання)
float cut_off = 2.90; // Значення відсічення (захист акумулятора)
float cut_notice = 3.10; // Значення, у якому працюватиме світлове
  оповіщення
int shift = 0; // Змінна для кількості моргання світлодіода
  попередження

const int led_r = 9; // Вихід контролера для червоного кольору RGB
  світлодіода
const int led_g = 10; // Вихід контролера для зеленого кольору RGB
  світлодіода

```

```

const int led _ b = 11; // Вихід контролера для синього кольору RGB
    світлодіода
const int led _ bb = 12; // Вихід контролера для синього світлодіода
const int vt _ on = 13;

const int blocking _ Vdd = 7; // Вихід контролера для самоблокування
    живлення контролера

//-----Змінні для вимикання та керування з кнопки---
    -----
const int button _ off =8; // Вихід контролера для контакту кнопки
    вимкнення
int flag _ button ; // Змінна для читання статусу кнопки
int flag _ off = 0; // Прапор для умови вимикання або керування
int flag _ vk = 0; //Прапор для умови передачі даних
int flag _ vk _ 2 = 0; //Прапор для умови передачі даних
int flag _ vk _ 3 = 1; //Прапор для умови передачі даних

//-----Змінні для діагностики-----
int flag _ mc = 0; // Прапор до виконання умов
int flag _ mc _ 2=0; // Прапор до виконання умов
int d _ 2 = 0; // Змінна для моніторингу датчиків холу
int d _ 3 = 0; // Змінна для моніторингу датчиків холу

//-----Цикл настройки-----

void setup () // Встановимо потрібні параметри в даному циклі
{
    Serial. begin (9600); // Відкриваємо порт і встановлюємо швидкість
        передачі
myNextion. init (" page 0"); // Виводимо початкову сторінку на
        дисплеї
attachInterrupt (1, sens _ a, RISING ); // Підключаємо переривання
        на 3 пін при підвищенні сигналу
attachInterrupt (0, sens _ b, RISING ); // Підключаємо переривання
        на 2 пін при підвищенні сигналу
pinMode ( led _ r, OUTPUT ); // 9 пін працюватиме на вихід
pinMode ( led _ g, OUTPUT ); // 10 пін працюватиме на вихід
pinMode ( led _ b, OUTPUT ); // 11 пін працюватиме на вихід
pinMode ( led _ bb, OUTPUT ); // 12 пін працюватиме на вихід pinMode (
    ( vt _ on, OUTPUT ); // 13 пін працюватиме на вихід pinMode (
    blocking _ Vdd, OUTPUT ); // 7 пін працюватиме на вихід
digitalWrite (blocking_Vdd, HIGH); // Подамо логічну одиницю на 7
    (включаємо живлення контролера)
digitalWrite ( led _ b, LOW ); // Подамо логічний нуль на 11
    (вимикаємо живлення кристала)
digitalWrite ( led _ bb, LOW ); // Подамо логічний нуль на 12
    (вимикаємо живлення світлодіода)
digitalWrite (vt_on, HIGH); // Подамо логічну одиницю на 13
    (включаємо живлення транзистора вимірювання)
analogWrite (led_r, 0); // Встановимо ШІМ червоного кристала
    світлодіода на виводі 9

```

```

analogWrite (led_g, 0); // Встановимо ШІМ зеленого кристала
    світлодіода на виводі 10
w_length=(float)EEPROM.read(1)/100.0; // Читаємо комірку пам'яті
    довжини кола колеса
hi_d=EEPROM.read(4); // Читаємо комірку пам'яті старшого біта умови
    зміни комірок при запуску системи (dist_con)
low_d = EEPROM.read (5); // Читаємо комірку пам'яті молодшого
    біта умови зміни комірок при запуску системи ( dist_con )
dist_con = word ( hi_d, low_d ); // Перетворюємо старший і
    молодший біт умови зміни комірок на єдине значення
hi_c = EEPROM.read (2); // Читаємо комірку пам'яті старшого біта
    значення зміни комірок при запуску системи ( addr )
low_c = EEPROM.read (3); // Читаємо комірку пам'яті молодшого біта
    значення зміни комірок при запуску системи ( addr )
addr = word ( hi_c, low_c ); // Перетворюємо старший і молодший
    біт значення зміни комірок на єдине значення
cell_hi = cell_hi + addr ; // Підсумовуємо значення початкового
    номера комірки та значення зміни комірок
cell_low = cell_low + addr ; // Підсумовуємо значення
    початкового номера осередку та значення зміни комірок
cell_comma = cell_comma + addr ; // Підсумовуємо значення
    початкового номера комірки та значення зміни комірок
hi = EEPROM.read ( cell_hi ); // Читаємо комірку пам'яті старшого
    біта сумарного пробігу під час запуску системи
low = EEPROM.read ( cell_low ); // Читаємо комірку пам'яті
    молодшого біта сумарного пробігу під час запуску системи
DIST_T = word ( hi, low ); // Перетворюємо старший і молодший біт
    сумарного пробігу на єдине значення
DIST = ( float ) EEPROM.read ( cell_comma ) / 100.0; // Читаємо
    комірку пам'яті добового пробігу при запуску системи (розподіл
    на 100 потрібно для збереження сотих часток відстані)
dht.begin (); // Запускаємо датчик вологості DHT 11

// Визначення опорної напруги
analogReference ( DEFAULT ); // DEFAULT INTERNAL використовувати Vcc
    як AREF
delay (100); //
Vcc = readVcc ();

if ( debug > 1)
{
    Serial.println ("---"); // Виводимо у послідовний порт цей
    символ
    delay (1000);
}

if ( debug > 1)
{
    Serial.print (" Vcc ="); // Виводимо у послідовний порт напис
    Serial.println ( Vcc ); // Виводимо в послідовний порт це
    значення змінної
    Serial.println ("---"); // Виводимо у послідовний порт цей
    символ
}

```

```

}
}

//-----Цикл обробки швидкості та відстані-----
void sens _ a () // підрахунок поточної швидкості та добового
    пробігу
{
    if ( millis ()- lastturn > 80) // Захист від випадкових вимірів
    {
        SPEED = w _ length /(( float )( millis ()- lastturn )/1000) *
            3.6; // Розрахунок швидкості, км/год
        lastturn = millis (); // Запам'ятовуємо час останнього обороту

DIST = DIST + w _ length / 1000; // Додаємо довжину колеса до
    дистанції при кожному його обороті (ділимо на 1000 для
    переведення метрів у кілометри)
flag _ sum = 1; // Привласнюємо змінної значення 1 умови виведення
    даних
flag _ vk =1; // Привласнюємо змінної значення 1 умови виведення
    даних
    }
}

//-----Цикл обробки обертів-----
void sens _ b () // Зробимо підрахунок поточних обертів педалей
{
    RPM = 60/(( float )( micros ()- lastflash )/1000000); //
    Розрахунок обертів, об / хв
    lastflash = micros (); // Запам'ятовуємо час останнього оберту
}

//-----Функції-----
float readVcc() {
byte i;
float result = 0.0;
float tmp = 0.0;

for (i = 0; i < 5; i++)
{
// Read 1.1V reference against AVcc
// Натисніть на reference до Vcc i the measurement to the internal
    1.1V reference
#if defined(__AVR_ATmega32U4__) || defined(__AVR_ATmega1280__)
    || defined(__AVR_ATmega2560__)
ADMUX = _BV (REFS0) | _BV (MUX4) | _BV (MUX3) | _BV (MUX2) | _BV (MUX1);
    #elif defined (__AVR_ATtiny24__) || defined(__AVR_ATtiny44__)
    || defined(__AVR_ATtiny84__)
ADMUX = _BV (MUX5) | _BV (MUX0);
    #elif defined (__AVR_ATtiny25__) || defined(__AVR_ATtiny45__) ||
    defined(__AVR_ATtiny85__)

```

```

ADMUX = _BV(MUX3) | _BV(MUX2);
#else
// works on an Arduino 168 or 328
ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
#endif

delay(3); // Wait for Vref to settle
ADCSRA |= _BV(ADSC); // Start conversion while (bit_is_set(ADCSRA,
    ADSC)); // Measuring

uint8_t low = ADCL; // must read ADCL first - it then locks ADCH
uint8_t high = ADCH; // unlocks both

tmp = (high<<8) | low;
tmp=(typVbg*1023.0)/tmp;
result = result + tmp;
delay(5);
}

result = result/5;
return result;
}

//----- Решта програми -----
void loop()
{
//-----Цикл обробки швидкості та відстані-----
-----
if (( millis ()- lastturn )>1000) // Якщо сигнал відсутній більше 1
    секунд, то...
{
    SPEED = 0; // Вважаємо, що змінна SPEED дорівнює 0
    if ( flag _ sum == 1) // Якщо рівність змінних дорівнює, то ...
        (крім підрахунку сумарного пробігу, захист від помилкових
        додавань сумарного пробігу)
    {
        dist _transit = DIST ; // Привласнюємо проміжною змінною
        значення добового пробігу
        dist transit = dist transit - dist past ; // Обчислюємо
        різницю добового пробігу на даний момент та значення добового
        пробігу при минулій зупинці
        DIST _ T = dist _ transit + DIST _ T ; // Різницю добового
        пробігу додаємо до загального пробігу
    dist_past = DIST ; // Надає змінній добового пробігу минулої
    зупинки, значення добового пробігу останньої зупинки
    flag _ sum =0; // Привласнюємо змінною роздільної здатності значення
    0 для умови виведення даних
    flag _ vk _ 2 = 1; // Привласнюємо змінній роздільної здатності
    значення 1 для умови виведення даних
    }
}
}

```

```

//-----Цикл обробки обертів-----
---
if (( micros ()- lastflash )>1000000)// якщо сигналу немає більше 1
    секунди, то...
{
    RPM = 0; // Вважаємо, що RPM дорівнює 0
}

//-----Цикл вимірювання температури та вологості-
-----
wet = dht. readHumidity (); // Зчитуємо температуру (кожні 250 мс)
temp = dht. readTemperature (); // Зчитуємо вологість (кожні 250 мс)

//-----Цикл збереження пробігу-----
-- ---
if ( flag == 1)
{
    myNextion. init ( " page 4" ); // При вимиканні перемикаємо на
    сторінку прощання з пілотом
    delay (5000); // Затримка для того, щоб встигли прочитати текст
    перед вимкненням
// Командами highByte і lowByte розділимо значення сумарного пробігу
    на старший і молодший біт
    hi = highByte ( DIST _ T ); // Привласнюємо змінної старший біт
    значення пробігу
    low = lowByte ( DIST _ T ); // Привласнюємо змінної молодший байт
    значення пробігу
    EEPROM. write ( cell _ hi, hi ); // Записуємо у внутрішню комірку
    пам'яті старший байт
    EEPROM. write ( cell _ low, low ); // Записуємо у внутрішню комірку
    пам'яті молодший байт
// Даний алгоритм використовується для збереження значення після
    коми добового пробігу
    dist excess = DIST ; // Привласнюємо значення добового пробігу
    тимчасової змінної і позбавляємося коми
    dist _ comma = DIST - dist _ excess ; // Вираховуємо різницю цілого
    числа і числа з комою, в результаті залишається різниця добового
    пробігу після коми (дозвіл 0,00)

    EEPROM.write(cell_comma,(float)dist_comma*100.0); // Записуємо
    значення добового пробігу після коми (множення на 100 потрібно
    для збереження сотих часток відстані)
    digitalWrite ( blocking _ Vdd, LOW ); //Відключаємо живлення
    мікроконтролера
}
// Даний алгоритм використовується для того, щоб збільшити ресурс
    роботи внутрішніх комірок пам'яті мікроконтролера

if ( DIST _ T == dist _ con )
{
    dist _ con = dist _ con +500; // Додамо змінної умови значення,
    що дорівнює 500 км.
}

```

```

hi _ d = highByte ( dist _ con ); // Привласнюємо змінної
старший біт значення умови
low _ d = lowByte ( dist _ con ); // Привласнюємо змінної молодший
байт значення умови
EEPROM. write (4, hi _ d ); // Записуємо у внутрішню комірку
пам'яті старший байт
EEPROM. write (5, low _ d ); // Записуємо у пам'яті молодший
байт
addr = addr +3; // Збільшуємо значення змінної на три для
перемикання комірок пам'яті
// обмеження максимального перемикання комірок пам'яті ,Що б
повернутися до перших комірок
if ( addr == 1017)
{
    addr = 0; // Привласнюємо даної змінної для перемикання
комірок пам'яті значення нуль
}
hi _ c = highByte ( addr ); // Привласнюємо змінної старший біт
значення перемикання комірок пам'яті
low _ c = lowByte ( addr ); // Привласнюємо змінній молодший
байт значення перемикання комірок пам'яті
EEPROM. write (2, hi _ c ); // Записуємо у внутрішню комірку
пам'яті старший байт
EEPROM. write (3, low _ c ); // Записуємо у пам'яті молодший байт
cell _ hi =6; // При зміні комірок повертаємо початкове значення
змінної
cell _ low = 7; // При зміні комірок повертаємо початкове значення
змінної
cell _ comma = 8; // При зміні комірок повертаємо початкове значення
змінної
cell _ hi = cell _ hi + addr ; // Вважаємо нове значення змінних
значення комірок пам'яті
cell _ low = cell _ low + addr ; // Вважаємо нове значення змінних
значення комірок пам'яті
cell comma = cell comma + addr ; // Вважаємо нове значення змінних
значення комірок пам'яті
}

//-----Цикл вибору довжини кола колеса-----
-----
String message = myNextion. listen (); // Очікуємо потрібної
комбінації символів із послідовного порту для виконання умов
if ( message == "65 3 6 1 ffff ffff ffff ")
{
    flag _ circle = flag _ circle +1; // Інкрементуємо значення
змінної на 1
}
if ( message == "65 3 7 1 ffff ffff ffff ")
{
    flag _ circle = flag _ circle -3; // Декрементуємо значення
змінної на 1
}
if ( flag _ circle == 19)

```



```

{
    flag _ circle = 1; // Привласнюємо значення змінної рівне 1 для
    того, щоб виявитися на початку списку
}
if ( flag _ circle == -1)
{
    flag _ circle = 17; // Привласнюємо значення змінної рівне 1 для
    того, щоб опинитися в кінці списку
}
// Ці умови служать для вибору довжини кола колеса
if ( flag _ circle == 1) // 12,5"
{
    w _ length = 0.99; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (12.5)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 2; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 3) // 16"
{
    w _ length = 1.27; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (16)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 4; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 5) // 20"
{
    w _ length = 1.59; // Привласнюємо значення змінної службовця
    визначення швидкості myNextion. setComponentText (" t 22",
    String (20)); // Виводимо на дисплей значення обраної на даний
    момент довжини кола
    flag _ circle = 6; // Привласнюємо це значення змінної для
    подальшого очікування
}

    if ( flag _ circle == 7) // 24"
{
    w _ length = 1.91; // Привласнюємо значення змінної службовця
    визначення швидкості
    myNextion. setComponentText (" t 22", String (24)); // Виводимо
    на дисплей значення обраної на даний момент довжини кола
flag circle = 8; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 9) // 26"
{

```

```

    w _ length = 2.07; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (26)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 10; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 11) // 27"
{
    w _ length = 2.11; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (27)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 12; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 13) // 27,5"
{
    w _ length = 2.19; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (27.5)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 14; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 15) // 28"
{
    w _ length = 2.33; // Привласнюємо значення змінної визначення
    швидкості
myNextion. setComponentText (" t 22", String (28)); // Виводимо
    на дисплей значення обраної на даний момент довжини кола
flag circle = 16; // Привласнюємо це значення змінної для подальшого
    очікування
}

    if ( flag _ circle == 17) // 29"
{
    w _ length = 2.31; // Привласнюємо значення змінної службовця
    визначення швидкості
myNextion. setComponentText (" t 22", String (29)); // Виводимо на
    дисплей значення обраної на даний момент довжини кола
flag circle = 18; // Привласнюємо це значення змінної для подальшого
    очікування
}
// Дана частина коду служить для збереження значення довжини кола
    колеса
if ( message == "65 3 8 1 ffff ffff ffff ")
{
    EEPROM. write (1, ( float ) w _ length * 100.0); // Записати в
    комірку пам'яті значення довжини кола
}

```

```

w _ length = ( float ) EEPROM. read (1)/100.0; // Читаємо
комірку пам'яті значення довжини кола
flag _ circle = 0; // Присвоюємо значення змінної для
відключення вибору після збереження
myNextion. setComponentText ( " t 22", " Ok "); // Виводимо на
дисплей повідомлення про те, що налаштування довжини кола
успішно збережено
}

//-----Цикл скидання пробігу-----
---
if ( message == "65 3 a 1 ffff ffff ffff ")
{
    reset = 1; // Присвоюємо значення змінної 1 умови скидання
    (підтвердження)
}
if ( message == "65 3 9 1 ffff ffff ffff ")
{
    reset _2=1; // Привласнюємо змінну 1 для умови скидання (повний
    пробіг)
}
if ( message == "65 3 16 1 ffff ffff ffff ")
{
    reset _3=1; // Присвоюємо значення змінної 1 умови скидання
    (добовий пробіг) }
if ( reset == 1 && reset _2 == 1)
{
    DIST = 0; // Привласнюємо змінну значення 0
    DIST _ T =0; // Привласнюємо змінну значення 0
    dist _ past =0; // Привласнюємо змінну значення 0
    dist _ transit =0; // Привласнюємо змінну значення 0
    dist _ comma = 0; // Привласнюємо змінну значення 0
    dist _ excess =0; // Привласнюємо змінну значення 0
    cell _ hi =6; // Привласнюємо змінну значення 0
    cell _ low =7; // Привласнюємо змінну значення 0
    cell _ comma =8; // Привласнюємо змінну значення 0
    dist _ con =500; // Привласнюємо змінну значення 500
    addr =0; // Привласнюємо змінної значення 0
    delay (1000); // Робимо паузу в 1 секунду для стабільності
    hi = highByte ( DIST _ T ); // Привласнюємо змінної старший біт
    значення пробігу
    low = lowByte ( DIST _ T ); // Надає змінній молодший байт значення
    пробігу
    EEPROM. write ( cell _ hi, hi ); // Записуємо у внутрішню комірку
    пам'яті старший байт
    EEPROM. write ( cell _ low, low ); // Записуємо у пам'яті молодший
    байт
    EEPROM. write ( cell _ comma, ( float ) dist _ comma * 100.0); //
    Записуємо у пам'яті значення добового пробігу після коми
    (множення на 100 потрібно для збереження сотих часток відстані)
    hi _ d = highByte ( dist _ con ); // Привласнюємо змінної старший
    біт значення умови

```

```

low _ d = lowByte ( dist _ con ); // Привласнюємо змінної молодший
    байт значення умови
EEPROM. write (4, hi _ d ); // Записуємо у внутрішню комірку
    пам'яті старший байт
EEPROM. write (5, low _ d ); // Записуємо у внутрішню комірку
    пам'яті молодший байт
hi _ c = highByte ( addr ); // Привласнюємо змінної старший біт
    значення перемикачів комірок пам'яті
low _ c = lowByte ( addr ); // Привласнюємо змінній
    молодший байт значення перемикачів комірок пам'яті
EEPROM. write (2, hi _ c ); // Записуємо у пам'яті старший байт
EEPROM. write (3, low _ c ); // Записуємо у пам'яті молодший
    байт
reset =0; // Привласнюємо змінну 0 для завершення циклу
reset _ 2=0; // Привласнюємо змінну 0 для завершення циклу
}
if ( reset == 1 && reset _ 3)
{
    DIST = 0; // Привласнюємо змінну значення 0
    dist _ transit =0; // Привласнюємо змінну значення 0
    dist _ past =0; // Привласнюємо змінної значення 0
    delay (1000); // Робимо паузу в 1 секунду для стабільності
    myNextion. setComponentText ( " t 3", String ( DIST )); //
        Відправляємо на екран значення добового пробігу
    reset =0; // Присвоюємо значення змінної 0 завершення циклу
    reset _ 3=0; // Привласнюємо змінну 0 для завершення циклу
}

//-----Цикл вимірювання напруги-----
Vcc = readVcc (); // Зчитуємо точну напругу з A 1, де буде наш
    вольтметр
curVoltage = 0.0; // Обнулюємо значення curVoltage
for ( i = 0; i < COUNT ; i ++ ) // Запускаємо цикл for
{
    curVoltage = curVoltage + analogRead ( A_PIN ) ; // Читаємо
    значення з аналогового входу та підсумовуємо
    delay (10); // Робимо паузу 10 мс для стабільності
}
curVoltage = curVoltage / COUNT ; // Ділимо отримане значення на
    константу напруги живлення (5 вольт)
float v = ( curVoltage * Vcc ) / 1024.0; // Отримуємо виміряну
    напругу шляхом математичного обчислення

if ( debug > 1)
{
    Serial. print ( " V ="); // Виводимо у послідовний порт напис
    Serial. print ( v ); // Виводимо в послідовний порт це значення
    змінної
}
// У даній умові ми показуємо пілоту, що акумулятор близький до
    заданого розряду
if ( v <= cut _ notice )

```

```

    { analogWrite ( led _ r, 255); // Встановимо ШІМ червоного
кристала світлодіода на 9
    analogWrite ( led_g, 125 ); // Встановимо ШІМ зеленого кристала
світлодіода на виводі 10
    delay (150); // Робимо паузу, 1 секунду
    analogWrite ( led _ r, 0); // Встановимо ШІМ червоного кристала
світлодіода на 9
analogWrite ( led _ g, 0); // Встановимо ШІМ зеленого кристала
світлодіода на 10
delay (150); // Робимо паузу, 1 секунду
}
// показуємо, що акумулятор розряджений

if ( v <= cut _ off )
{
    flag _ leave = 0; // Привласнюємо змінної значення
analogWrite ( led _ r, 255); // Встановимо ШІМ червоного кристала
світлодіода на 9
delay (250);
analogWrite ( led _ r, 0); // Встановимо ШІМ червоного кристала
світлодіода на виводі 9
delay (300);
shift = shift +1; // Інкрементуємо значення змінної
}
// Даний алгоритм відключає живлення контролера після достатньої
кількості попереджень
if ( shift == 10)
{
    flag = 1; // Присвоюємо значення змінної 1 до виконання умови
виключення
}

//-----Цикл вимикання та управління з кнопки-----
-----
flag _ button = digitalRead ( button _ off ); // Читаємо статус
кнопки...
    if ( flag _ button == 1)
    {
        flag _ off = flag _ off +1; // Інкрементуємо змінну на 1
delay (500);
    }
// За такого значення вимикаємо велокомп'ютер
if ( flag _ off == 5 && flag _ button == 1)
    {
        flag _ off = constrain ( flag _ off, 0, 5);
        digitalWrite ( led _ g, LOW ); // Вимикаємо зелений світлодіод
        digitalWrite ( led _ r, HIGH ); // Включаємо червоний світлодіод
        flag = 1; // Привласнюємо змінною дане значення для виконання
умови
    }
// При такому значенні показуємо, що можна включити режим
налаштування
if ( flag _ off == 3)

```

```

{
    digitalWrite ( led_g, HIGH ) ; // Включаємо зелений світлодіод
}
// При такому значенні включаємо режим налаштування

if ( flag _ off >= 3 && flag _ button == 0)
{
    flag _ off = 10; // Привласнюємо змінною це значення до
    виконання умови
flag _ vk _3=0; // Привласнюємо змінною дане значення для виконання
    умови
}
// За такого значення вимикаємо режим налаштування

if ( flag _ off == 13)
{
    flag _ off = constrain ( flag _ off, 10, 13); // Обмежимо цю
    змінну, щоб точно вимкнути і не промахнутися
    digitalWrite ( led_g, LOW ) ; // Вимикаємо зелений світлодіод
    flag _ off = 0; // Привласнюємо змінною дане значення до
    виконання умови
    flag _ vk _3=1; // Привласнюємо змінною дане значення для
    виконання умови
}

//-----Цикл діагностики-----
if ( message == "65 3 b 1 ffff ffff ffff ")
{
    flag _ mc = flag _ mc +1; // Привласнюємо змінною дане значення
    для виконання умови
}

if ( flag _ mc == 1)
{
    myNextion. setComponentText ( " t 30", " On "); // Відправляємо
    на екран текстову інформацію про включення
    flag _ mc =2; // Привласнюємо змінною дане значення для
    виконання умови
}
if ( flag _ mc == 2)
{
    digitalWrite ( led_g, LOW ) ; // Вимикаємо зелений світлодіод
    d_2 = digitalRead (2); // Зчитуємо стан цифрового входу
    d_3 = digitalRead (3); // Зчитуємо стан цифрового входу

    if ( d _2 == 1)
    {
        digitalWrite ( led _ b, LOW ); // Подамо логічний нуль на 11
        висновок (вимикаємо живлення кристала)
    }

    if ( d _3 == 1)
    {

```

```

    digitalWrite ( led _ bb, LOW ); // Подамо логічний нуль на 12
вихід (вимикаємо живлення світлодіода)
}
if ( d _ 2 == 0)
{
    digitalWrite ( led _ b, HIGH ); // Подамо логічний нуль на 11
вихід (вимикаємо живлення кристала)
}
    if ( d _ 3 == 0)
{
    digitalWrite ( led _ bb, HIGH ); // Подамо логічний нуль на 12
вихід (вимикаємо живлення світлодіода)
}
}
}
if ( flag _ mc == 3)
{
    digitalWrite ( led _ b, LOW ); // Подамо логічний нуль на 11
вихід (вимикаємо живлення кристала)
    digitalWrite ( led _ bb, LOW ); // Подамо логічний нуль на 12
вихід (вимикаємо живлення світлодіода)
    digitalWrite ( led _ g, HIGH ); // Включаємо зелений світлодіод
    myNextion. setComponentText ( " t 30", " Off "); // Надсилаємо на
дисплей текстову інформацію про вимкнення
    flag _ mc = 0; // Привласнюємо змінною дане значення для
виконання умови
}

//-----Цикл відправки інформації на дисплей-----
----- if ( flag _ vk _ 2 == 1)
{
    myNextion. setComponentText ( " t 1", String ( SPEED )); //
Відправляємо на дисплей значення швидкості велосипеда
    myNextion. setComponentText ( " t 2", String ( RPM )); //
Відправляємо на дисплей значення каденсу
    myNextion. setComponentText ( " t 3", String ( DIST )); //
Відправляємо на екран значення добового пробігу
    flag _ vk =0; // Привласнюємо змінною це значення до виконання
умови
    flag _ vk _ 2=0; // Привласнюємо змінною дане значення до
виконання умови
    flag _ vk _ 3=1; // Привласнюємо змінною дане значення для
виконання умови
}
if ( flag _ vk == 1)
{
    myNextion. setComponentText ( " t 1", String ( SPEED )); //
Відправляємо на дисплей значення швидкості велосипеда
    myNextion. setComponentText ( " t 2", String ( RPM )); //
Відправляємо на екран значення каденса
    myNextion. setComponentText ( " t 3", String ( DIST )); //
Відправляємо на екран значення добового пробігу
}
if ( flag _ vk _ 3 == 1)

```

```
{  
    myNextion. setComponentText ( " t 11", String ( temp )); //  
    Відправляємо на дисплей значення температури  
    myNextion. setComponentText ( " t 12", String ( wet )); //  
    Відправляємо на дисплей значення вологості myNextion.  
    setComponentText ( " t 13", String ( DIST _ T )); // Відправляємо  
    на екран значення повного пробігу  
    myNextion. setComponentText ( " t 24", String ( v )); //  
    Відправляємо на дисплей значення напруги акумулятора  
    myNextion. setComponentText ( " t 25", String ( Vcc )); //  
    Відправляємо на дисплей значення напруги живлення  
    велокомп'ютерної мережі  
}  
}
```



