

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки


«Затверджую»  
Зав. кафедри теоретичної та  
прикладної системотехніки  
\_\_\_\_\_ д.т.н., проф. С. І. Шматков  
« \_\_\_ » \_\_\_\_\_ 2024 р

## Пояснювальна записка


до кваліфікаційної роботи  
бакалавра

на тему: «Програмна модель ШІМ контролера»

Захищено на засіданні  
Атестаційної комісії № 42  
протокол № \_\_\_ від \_\_.06.2024 р.  
Оцінка \_\_\_\_\_ / \_\_\_\_\_  
Голова Атестаційної комісії  
\_\_\_\_\_ **СКОБ Ю. О.**

Виконав:  
студент 4 курсу, групи КІ-41  
Галузь знань: 12 – Інформаційні  
технології  
Спеціальність: 123 – Комп'ютерна  
інженерія.  
**ЦИБА Богдан Олегович** 

**Керівник:**  
к.ф.-м.н., доцент ЗВО  
**КОТВИЦЬКИЙ Альберт Тадеушевич**  


**Рецензент:** к.т.н., доцент ЗВО кафедри  
електроніки і управляючих систем  
**РЕВА Сергій Миколайович**  


## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи складає 53 сторінки, із яких 33 сторінок основної частини з 14 рисунками, 3 таблицею, 9 найменуваннями списку використаних джерел та двома додатками.

Метою кваліфікаційної роботи є аналіз та моделювання програмної моделі ШІМ контролера для ефективного управління потужністю в електронних системах.

**Об'єкт дослідження** – це процес програмної симуляції технології ШІМ.

**Предмет дослідження** – симуляція програмної моделі ШІМ з використанням мікроконтролера.

Проблема, яка вирішується у даній кваліфікаційній роботі, полягає в аналізі та моделюванні програмної моделі ШІМ (Широтно-імпульсна модуляція) контролера для ефективного управління потужністю в електронних системах.

Область застосування – промисловість, електроніка споживчого призначення, альтернативна енергетика, робототехніка.

**Ключові слова:** ШІМ, ATmega328, SimulIDE, Arduino, C, ефективність електронних систем, програмне моделювання.

## ABSTRACT

An explanatory note to a bachelor's thesis consists of an introduction, three sections, conclusions, a list of sources used, and two appendices. The total volume of the work is 53 pages, of which 33 pages are the main part with 14 figures, 3 tables, 9 titles of the list of used sources and two appendices.

The purpose of the qualification work is the analysis and modeling of the PWM controller software model for effective power management in electronic systems.

The object of research is the process of software simulation of PWM technology. The subject of the study is the simulation of the PWM software model using a microcontroller.

The problem solved in this qualification work is the analysis and modeling of the PWM (Pulse Width Modulation) software model of the controller for effective power management in electronic systems.

Field of application – industry, consumer electronics, alternative energy, robotics.

Keywords: PWM, ATmega328, SimulIDE, Arduino, C, efficiency of electronic systems, software simulation.

## ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ШИРОТНО-ІМПУЛЬСНА МОДУЛЯЦІЯ.....	7
1.1 Загальна інформація о ШІМ.....	7
1.2 Основні параметри ШІМ.....	8
1.3 Переваги і недоліки.....	10
1.4 Застосування ШІМ контролерів.....	12
1.5 Способи реалізації ШІМ.....	14
Висновки за розділом 1 .....	15
РОЗДІЛ 2. ПІДГОТОВКА АПАРАТНОЇ ЧАСТИНИ .....	18
2.1 Вибір платформи.....	18
2.2 ATmega328.....	20
2.2.1 Архітектура.....	20
2.2.2 Характеристики.....	20
2.2.3 Особливості та можливості.....	22
2.2.4 Програмування.....	22
2.2.5 Використання.....	22
2.3 Таймери.....	23
Висновки за розділом 2.....	25
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ШІМ КОНТРОЛЕРА.....	27
3.1. Рекомендації щодо впровадження і експлуатації.....	27
3.2 Створення схеми в симуляторе SimulIDE.....	28
3.3. Написання коду.....	30
3.3.1 Налаштування таймера.....	30
3.3.2 Налаштування пінів.....	33
Висновки за розділом 3.....	34
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТКИ.....	37

## ВСТУП

Широтно-імпульсна модуляція (ШІМ) є однією з найважливіших технологій, що широко використовується в сучасній електроніці та системах управління. Завдяки своїй здатності точно контролювати подачу енергії, ШІМ знаходить застосування в різних галузях, від промислової автоматизації до споживчої електроніки та автомобільної промисловості. Програмні моделі ШІМ контролерів є особливо важливими, оскільки вони дозволяють реалізовувати гнучкі та ефективні рішення для управління різними пристроями і системами.

**Актуальність роботи.** Дослідження зумовлені потребою в енергоефективних і точних методах управління в контексті зростаючого попиту на автоматизацію та оптимізацію процесів. Зокрема, в умовах глобальних енергетичних викликів використання технологій ШІМ дозволяє істотно знизити енергоспоживання, підвищити продуктивність і забезпечити високий рівень керування в різних системах. Програмні моделі ШІМ-контролерів, на відміну від апаратних, дають можливість швидко адаптуватися до мінливих вимог і оновлювати функціонал без істотних витрат.

**Мета дослідження** моделювання програмної моделі ШІМ контролера для раціонального керування потужністю в електронних системах.

**Об'єкт дослідження** – це процес програмної симуляції технології ШІМ.

**Методи дослідження:** аналіз літературних джерел, експериментальне моделювання, аналіз результатів експериментів.

**Предмет дослідження** – симуляція програмної моделі ШІМ з використанням мікроконтролера.

### **Завдання дослідження**

1. Огляд літератури: опис програмної моделі ШІМ контролера, основні принципи роботи та застосування в електротехніці.
2. Розробка математичної моделі: визначення математичних виразів для опису динаміки системи, яка використовує ШІМ контролера

3. Моделювання: використання програмного забезпечення для симуляції роботи програмної моделі ШІМ контролера з різними параметрами роботи

4. Аналіз результатів: оцінка ефективності та стабільності роботи ШІМ контролера.

## РОЗДІЛ 1

### ШИРОТНО-ІМПУЛЬСНА МОДУЛЯЦІЯ

#### 1.1 Загальна інформація о ШІМ

Широтно-імпульсна модуляція (ШІМ) - це інноваційний метод управління електричними пристроями і системами, що відкриває широкі можливості для ефективного регулювання їх параметрів. Одним із ключових принципів ШІМ є зміна ширини імпульсу сигналу залежно від значення аналогового вхідного сигналу. Цей метод дозволяє точно контролювати потужність, швидкість та інші характеристики електронних пристроїв і систем, що критично для багатьох сучасних застосувань.

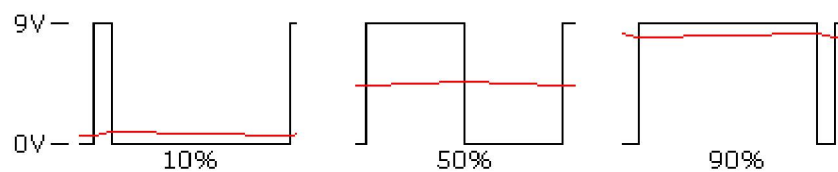


Рисунок 1.1 – Робочий цикл ШІМ [7]

Сигнал ШІМ складається з послідовності імпульсів фіксованої частоти, але змінної тривалості. Ця змінна тривалість імпульсу визначається робочим циклом, що виражається у відсотках загального періоду сигналу. Робочий цикл відображає середнє значення напруги або струму, що подається на навантаження, і дозволяє точно налаштувати вихідні параметри системи.

Однією з важливих переваг ШІМ є її ефективність порівняно з аналоговими системами керування. Аналогові пристрої, як правило, виділяють багато тепла через значний струм, що проходить через них, тоді як цифрові системи майже не виділяють тепла. Це стало можливим завдяки тому, що в цифрових системах більшість теплового випромінювання виникає під час переходів між станами, коли пристрій перебуває між двома станами. Використання ШІМ дозволяє мінімізувати споживання енергії, так як

потужність пропорційна добутку напруги і струму, і якщо один з цих параметрів наближається до нуля, то потужність буде мінімальною.

Широтно-імпульсна модуляція зберігає багато характеристик аналогових систем управління, оскільки цифровий сигнал може приймати будь-які значення в межах допустимого діапазону. Це робить ШІМ універсальним методом управління, який може успішно застосовуватись у різних галузях: від промислового до побутового застосування. Навіть у випадках, коли необхідно зберегти дані, контролери високого класу можуть працювати без конфліктів, забезпечуючи надійну та ефективну роботу системи управління.

## 1.2 Основні параметри ШІМ

ШІМ має ряд параметрів, які визначають його характеристики та функціональність:

- Розрядність: цей параметр визначає кількість можливих значень амплітуди або часу у цифровому сигналі ШІМ. У контексті ШІМ-контролера розрядність вказує кількість можливих рівнів ширини імпульсу, наприклад, використання 8-бітного контролера дозволяє отримати 256 рівнів ширини імпульсу. Він визначається за формулою:

$$N=2^n-1,$$

де:

$N$  - кількість можливих рівнів ширини (або амплітуди) імпульсу в сигналі ШІМ;

$n$  - розрядність, тобто кількість біт, що використовуються для кодування амплітуди.

- Шпаруватість: цей параметр вказує пропорцію часу, протягом якого сигнал увімкнено (високий рівень), відносно загального періоду сигналу. У ШІМ-контролері робочий цикл

використовується для визначення часу, протягом якого вихідний сигнал буде увімкнено. Визначається за формулою:

$$D = \frac{T_{on}}{T},$$

де:

$D$  – шпаруватість;

$T_{on}$  – час високого рівня сигналу (час, коли сигнал ШІМ "включено").

$$T_{on} = D * T$$

$T$  – період сигналу, тобто час одного повного циклу.

$$T = \frac{1}{F_{PWM}}$$

- Значення розділювача: Цей параметр використовується для розділення основного генератора часу, наприклад таймера, на певне значення для отримання бажаної частоти для ШІМ. Це значення може бути налаштовано для досягнення необхідної частоти ШІМ сигналу.
- Частота: цей параметр вказує на кількість повторюваних циклів сигналу в одиницю часу. У ШІМ-контролері частота визначає, як часто буде змінюватися ширина імпульсу в сигналі джерела. Визначається за формулою:

$$F_{PWM} = \frac{F_{CPU}}{N * 2^n},$$

де:

$F_{PWM}$  – частота сигналу;

$F_{CPU}$  – частота тактового генератора (16 МГц);

$N$  – Значення піддільника (1, 8, 64, 256 чи 1024);

$n$  – бітність

- Період: час, необхідний для завершення одного циклу сигналу, включаючи час увімкнення та вимкнення. У ШІМ-контролері період визначає, як довго триватиме один цикл широтно-імпульсної модуляції. Він визначається за формулою:

$$T = \frac{1}{F_{PWM}},$$

де,

$T$  – період;

$F_{PWM}$  – частота;

Ці параметри визначають можливості та характеристики ШІМ, дозволяючи налаштувати його для різноманітних застосувань, від промислових систем керування до споживчих електронних пристроїв. Оптимальне використання цих параметрів дозволяє досягти необхідної ефективності і точності управління, що робить ШІМ незамінною технологією в сучасній електроніці.

### 1.3 Переваги і недоліки

Програмні моделі ШІМ-контролерів мають важливі переваги, які роблять їх незамінними в сучасних технічних рішеннях:

1. Енергоефективність: ШІМ зменшує втрати тепла порівняно з аналоговими методами керування потужністю, оскільки ключові компоненти працюють у режимі ввімкнення/вимкнення, зменшуючи розсіювання тепла.
2. Точне керування: ШІМ забезпечує високоточне керування параметрами (такими як швидкість двигуна або яскравість світлодіода) завдяки точному контролю ширини імпульсу.
3. Гнучкість: моделі програмного забезпечення ШІМ-контролерів легко налаштовуються та адаптуються до різних умов і вимог без необхідності змінювати апаратне забезпечення.

4. Сумісність: ШІМ може бути реалізований на багатьох різних мікроконтролерах та процесорах цифрових сигналів, що забезпечує універсальність застосування.
5. Масштабованість: можливість реалізації ШІМ на різних рівнях потужності - від низького (для невеликих пристроїв) до високого (для промислових застосувань).

Незважаючи на численні переваги, широтно-імпульсна модуляція має певні недоліки, які слід враховувати при її використанні. Розглянемо основні недоліки:

1. Електромагнітні перешкоди (ЕМІ): ШІМ створює високочастотні імпульси, які можуть створювати електромагнітні перешкоди (ЕМІ). Це може негативно вплинути на роботу чутливих електронних пристроїв, розташованих поблизу.
2. Чутливість навантаження: продуктивність ШІМ може знизитися під час роботи з індуктивними або ємнісними навантаженнями. Індуктивні навантаження, такі як електродвигуни, можуть викликати додаткові стрибки струму та напруги, які ускладнюють керування.
3. Потрібна висока частота: для забезпечення плавного та тихого керування часто потрібні високі частоти модуляції. Це може вимагати використання мікроконтролерів або інших високошвидкісних апаратних рішень, що збільшує складність системи та вартість. Крім того, робота на високій частоті може спричинити додаткові втрати тепла в комутаційних компонентах.
4. Складність реалізації: реалізація ефективної системи ШІМ потребує детального розуміння як апаратних, і програмних аспектів. Схематичне проектування, розрахунок параметрів та налаштування програмного забезпечення потребують високого рівня технічних знань. Для новачків у цій галузі це може стати

проблемою, що призводить до додаткових витрат часу та ресурсів на навчання та налагодження.

5. Акустичний шум: ШІМ може створювати акустичний шум, особливо при використанні в системах, що працюють у діапазоні звукових частот. Наприклад, вентилятори або двигуни можуть створювати шум, який може бути неприємним для користувачів. Особливо це стосується побутової техніки або пристроїв, що працюють у безпосередній близькості від людей.
6. Нелінійність керування. У деяких додатках нелінійність сигналу ШІМ може бути проблематичною. Досягнення точного управління може вимагати складного калібрування або використання додаткових алгоритмів компенсації, що збільшує складність розробки та впровадження системи.
7. Обмеження навантаження: ШІМ-контролери можуть мати обмеження на максимальну потужність навантаження. Високі струми і напруги можуть вимагати використання спеціальних транзисторів і систем охолодження, що збільшує розмір і вартість кінцевого продукту.

#### **1.4 Застосування ШІМ контролерів**

Широтно-імпульсні модулятори широко використовуються в різних галузях промисловості завдяки своїм універсальним можливостям регулювання потужності та високому ККД. Ось деякі з найпоширеніших застосувань ШІМ-контролерів:

##### **1. Промисловість**

У промислових системах ШІМ-контролери використовуються для управління різними пристроями і механізмами.:

- Електродвигуни: ШІМ дозволяє точно контролювати швидкість і крутний момент електродвигунів, що має вирішальне значення для багатьох промислових процесів.

- Насоси та компресори: контролери ШІМ забезпечують точне керування потоком рідини чи газу, забезпечуючи ефективну роботу систем опалення, вентиляції та кондиціонування.
- Приводи та клапани: використовуються для керування рухом механізмів та регулювання потоку рідини або газу в системах промислової автоматизації.

## 2. Енергетика

В енергетиці ШІМ-контролери використовуються для управління електрикою і різними електронними пристроями:

- Інвертори: використовуються для перетворення постійного струму на змінний і подачі його на електричні пристрої, такі як двигуни, енергосистеми і т.д.
- Стабілізатори напруги: ШІМ дозволяє регулювати вихідну напругу і стабілізувати її на певному рівні, що дуже важливо для надійного живлення різних електронних пристроїв.
- Сонячні панелі та вітрогенератори: використовуються для оптимального збирання та перетворення енергії з альтернативних джерел, допомагаючи знизити залежність від традиційних джерел енергії.

## 3. Споживча електроніка

У побутовій та споживчій електроніці ШІМ контролери застосовуються для управління різними пристроями та системами:

- Освітлення: використовується для регулювання яскравості світлодіодних ламп, диммерних ламп тощо.
- Кліматичні системи: дозволяють точно контролювати температуру, вологість та інші параметри середовища в приміщенні.

- Побутова техніка: використовується для керування роботою вентиляторів, насосів, пилососів та інших побутових пристроїв.

### 1.5 Способи реалізації ШІМ

Існує кілька методів реалізації широтно-імпульсної модуляції (ШІМ), які можна розділити на апаратні та програмні. Кожен має свої переваги та недоліки, а також підходить для різних типів застосувань.

- За допомогою технічних рішень і схем:
  1. ШІМ на основі компаратора: У цій схемі аналоговий сигнал порівнюється з певним пороговим значенням, і результат порівняння використовується для визначення стану вихідного сигналу. Наприклад, якщо аналоговий сигнал перевищує порогове значення, вихідний сигнал буде "високим", а якщо менше - "низьким". Цей метод забезпечує простий та ефективний спосіб реалізації ШІМ.
  2. ШІМ на основі таймера/лічильника мікроконтролера: У більш складних системах ШІМ може бути реалізований за допомогою вбудованих таймерів або лічильників мікроконтролера. Мікроконтролер генерує цифровий ШІМ сигнал, використовуючи програмне забезпечення для розрахунку і керування тривалістю імпульсів та їх частотою.
  3. ШІМ на основі оперативного підсилювача: Вхідний аналоговий сигнал подається на вхід оперативного підсилювача, а ширина імпульсів визначається відносною тривалістю часу, протягом якого вихідний сигнал перебуває в високому стані порівняно з періодом, протягом якого він перебуває в низькому стані.
- За допомогою програмованих цифрових пристроїв:

1. Мікроконтролери: є одними з найпоширеніших програмованих пристроїв для управління електронікою. Вони мають вбудовані таймери та лічильники, які можуть бути програмовані для генерації ШІМ сигналів. При цьому використовується внутрішній таймер мікроконтролера для визначення ширини імпульсів ШІМ.
2. FPGA (Field-Programmable Gate Array): це програмовані вентильні матриці, які можуть бути програмно налаштовані для виконання будь-яких логічних функцій. Завдяки своїй гнучкості та високій швидкості обробки, FPGA можуть бути використані для ефективної реалізації ШІМ у високопродуктивних системах.
3. DSP (Digital Signal Processor): це спеціалізовані мікропроцесори, призначені для обробки цифрових сигналів в реальному часі. Вони зазвичай мають вбудовані функції для генерації ШІМ сигналів та можуть бути програмовані для реалізації різних алгоритмів управління.
4. CPLD (Complex Programmable Logic Device): це ще один тип програмованих цифрових пристроїв, які можуть бути використані для реалізації ШІМ. Вони зазвичай мають більшу кількість логічних елементів порівняно з FPGA, але меншу гнучкість у програмуванні.

### **Висновки за розділом 1**

Проаналізувавши всю цю інформацію, можна зробити кілька висновків:

1. ШІМ є важливим інструментом сучасної електроніки, який дозволяє досягти високої ефективності, точності та надійності управління електричними пристроями та системами. Це зумовлено можливістю точно регулювати ширину імпульсів, що

дозволяє оптимізувати енергоспоживання і покращити продуктивність різних електронних компонентів.

2. ШІМ має велику кількість переваг, таких як висока ефективність, гнучкість у налаштуваннях, можливість зменшення енергоспоживання та зниження теплових втрат. Проте, вона не бездоганна і має низку недоліків. До них можна віднести складність у реалізації для високочастотних сигналів, можливість виникнення електромагнітних завад, а також потребу в детальному налаштуванні та калібруванні для досягнення оптимальної роботи.
3. Завдяки гнучкості та можливості точного регулювання параметрів сигналу, ШІМ технологія знаходить широке застосування у різних галузях, таких як промисловість, енергетика, споживча електроніка та автомобільна промисловість. У промисловості вона використовується для автоматизації та точного контролю процесів, в енергетиці - для керування електродвигунами та інверторами, у споживчій електроніці - для регулювання яскравості світлодіодів та швидкості вентиляторів, а в автомобільній промисловості - для управління системами освітлення та двигунами.
4. ШІМ технологія продовжує розвиватися з удосконаленням апаратних та програмних компонентів, що розширює її можливості та сфери застосування. Новітні розробки спрямовані на підвищення частоти роботи, зниження рівня електромагнітних завад та поліпшення точності регулювання, що робить ШІМ ще більш ефективним інструментом у сучасних технологіях.
5. Впровадження ШІМ контролерів у різні системи дозволяє значно підвищити їхню енергоефективність, що є важливим фактором у контексті глобальних тенденцій до зменшення енергоспоживання та покращення екологічних показників. Таким чином, ШІМ не

лише покращує технічні характеристики пристроїв, але й сприяє екологічній сталості.

Отже, ШІМ є ключовою технологією, яка значно впливає на сучасну електроніку та промислові процеси, забезпечуючи високий рівень ефективності, точності та гнучкості в управлінні електричними системами.

## РОЗДІЛ 2

### ПІДГОТОВКА АПАРАТНОЇ ЧАСТИНИ

#### 2.1 Вибір платформи

Як платформу для реалізації ШПМ я вибрав мікроконтролери. Оскільки вони мають ряд переваг, які роблять їх найбільш популярним вибором для таких програм:

1. Гнучкість програмування: Мікроконтролери дозволяють легко програмувати алгоритми ШПМ та налаштовувати їх відповідно до конкретних потреб проекту.
2. Вбудовані функції: Багато мікроконтролерів, включаючи ATmega328, мають вбудовані таймери та PWM виходи, що спрощує реалізацію ШПМ без необхідності додаткових зовнішніх пристроїв.
3. Низька вартість: Мікроконтролери, зокрема ATmega328, доступні за помірну ціну і підходять для різних застосувань в різних сферах, від проектів DIY до промислових систем.
4. Низьке споживання енергії: Багато мікроконтролерів мають оптимізовану енергоефективність, що робить їх ідеальними для пристроїв, що працюють від акумуляторних джерел живлення або в енергозберігаючих системах.

На вибір є декілька мікроконтролерів:

1. ATmega328 (Microchip):
  - Вбудований PWM модуль: спрощує реалізацію ШПМ без потреби в додаткових пристроях..
  - Низька вартість і широка популярність: Це один з найпоширеніших мікроконтролерів, що робить його доступним та підходящим для багатьох проектів.

- Сумісність з Arduino IDE: Це важливий фактор для тих, хто використовує Arduino-сумісне програмне забезпечення для розробки.
- Ідеальний для простих до середньої складності проектів: ATmega328 чудово підходить для невеликих або середньо-розмірних проектів, де потрібна простота та ефективність.

## 2. STM32 серії (STMicroelectronics):

- Різноманітні моделі: STM32 пропонує широкий вибір моделей з різними характеристиками, що дозволяє підібрати оптимальний контролер для конкретного завдання.
- Висока продуктивність: STM32 має високу продуктивність та швидку реакцію на зміни вхідного сигналу, що робить його ідеальним для вимогливих додатків.
- Підтримка широкого спектру протоколів зв'язку: Завдяки вбудованій підтримці протоколів, таких як USB, UART, SPI, I2C, STM32 може бути дуже універсальним рішенням для багатьох застосувань.

## 3. PIC серії (Microchip):

- Надійність та стабільність : PIC має репутацію надійних та стабільних мікроконтролерів, що робить їх популярними у багатьох галузях.
- Широкий вибір моделей: Як і STM32, PIC пропонує широкий вибір моделей з різними функціональними можливостями, що дозволяє підібрати контролер для конкретних вимог проекту.
- Вбудовані апаратні PWM модулі: Наявність вбудованих PWM модулів спрощує реалізацію ШІМ і забезпечує стабільну роботу.

Я вибрав ATmega328, оскільки він найкраще підходить для поставленого завдання. Він має вбудований ШІМ-модуль і коштує досить дешево.

## 2.2 ATmega328

ATmega328 - це мікроконтролер, який широко використовується для створення різноманітних електронних пристроїв, таких як Arduino.

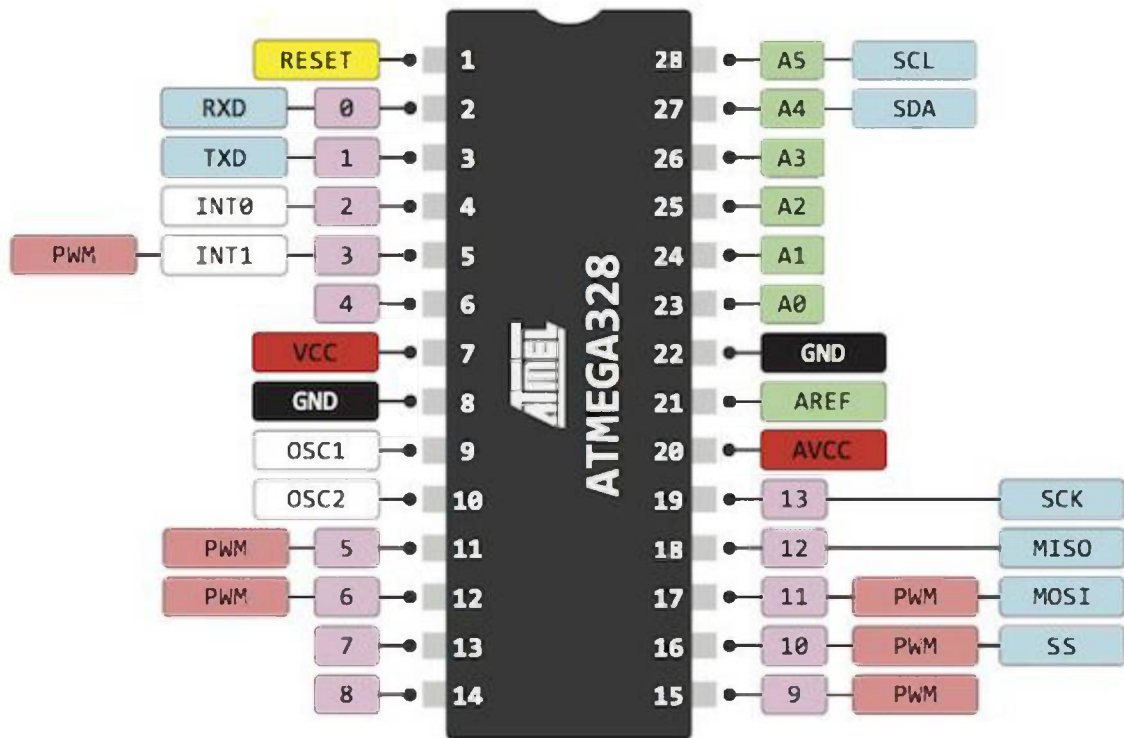


Рисунок 2.1 – Схема ATmega328 [6]

### 2.2.1 Архітектура:

ATmega328 належить до сімейства мікроконтролерів AVR, вироблених Microchip Technology (раніше Atmel). Цей мікроконтролер використовує архітектуру RISC (Reduced Instruction Set Computing) архітектуру.

### 2.2.2 Характеристики

#### 1. Процесор:

- Архітектура: 8-бітний AVR
- Тактова частота: до 20МГц
- Підтримка RISC-інструкцій: 131 інструкція

#### 2. Пам'ять:

- Флеш-пам'ять: 32 кілобайти
- SRAM: 2 кілобайти
- EEPROM: 1 кілобайт

### 3. Живлення:

- Напруга живлення: 1.8V - 5.5V
- Споживана потужність: залежить від режиму роботи, може бути в діапазоні від кількох міліампер до кількох мікроампер у режимах сну.

### 4. Інтерфейси введення/виведення:

- Цифрові входи/виходи: 23
- Аналогові входи: 6 (10-бітний АЦП)
- ШІМ (PWM) виходи: 6

### 5. Таймери:

- Один 16-бітний таймер
- Два 8-бітні таймери

### 6. Комунікаційні інтерфейси:

- UART: 1
- SPI: 1
- I2C: 1

### 7. Режими енергозбереження:

- Idle
- ADC Noise Reduction
- Power-save
- Power-down
- Standby
- Extended Standby

### 2.2.3 Особливості та можливості

1. Вбудований Watchdog Timer: Забезпечує безперервний моніторинг роботи мікроконтролера і може виконувати перезавантаження системи в разі збою.
2. Широкий діапазон робочих температур: ATmega328 може працювати при температурах від  $-40^{\circ}\text{C}$  до  $+85^{\circ}\text{C}$ , що робить його придатним для використання в різних умовах.
3. Вбудовані апаратні модулі: Підтримка апаратного множення, що підвищує швидкість виконання складних математичних операцій.
4. Самопрограмування: мікроконтролер може оновлювати власну прошивку за допомогою спеціальних завантажувачів.

### 2.2.4 Програмування

- Arduino IDE: Найпоширеніша платформа для програмування ATmega328. Платформа забезпечує простий інтерфейс та бібліотеки для роботи з різними датчиками та модулями.
- Atmel Studio: Потужний інструмент для професійної розробки, який підтримує налагодження та симуляцію.
- SimulIDE: Проста, але потужна платформа для моделювання електронних схем і мікроконтролерів. Він призначений для студентів, інженерів і техніків з електроніки, які хочуть швидко й ефективно тестувати свої проекти.

### 2.2.5 Використання

ATmega328 широко використовується в різних проектах:

- Arduino Uno: Найвідоміша платформа розробки, побудована на основі ATmega328.
- Робототехніка: Використовується для створення роботів, автоматичних систем управління.

- Сенсорні системи: Завдяки вбудованим АЦП, підходить для зчитування аналогових сигналів від датчиків.
- Автоматизація домашніх систем: Управління освітленням, системами безпеки та іншими домашніми пристроями.

### 2.3 Таймери

Мікроконтролер ATmega328 має три таймери: Timer0, Timer1 та Timer2. Ці таймери є важливими компонентами для створення різноманітних функцій, таких як генерація ШІМ, відлік часу, керування подіями тощо. Розглянемо кожен таймер детальніше:

#### 1. Timer0:

- Розрядність: 8 біт
- Режими роботи: Нормальний, ШІМ (Fast PWM, Phase Correct PWM), СТС (Clear Timer on Compare Match)
- Джерела тактового сигналу: Внутрішній крок (з визначником), зовнішній крок.
- Особливості: Часто використовується для функцій затримки, таких як `delay()` в середовищі Arduino.

#### 2. Timer1:

- Розрядність: 16 біт
- Режими роботи: Нормальний, ШІМ (Fast PWM, Phase and Frequency Correct PWM), СТС, режим захоплення
- Джерела тактового сигналу: Внутрішній крок (з визначником), зовнішній крок
- Особливості: Завдяки своїй розрядності підходить для більш точних вимірювань часу і генерації довгих імпульсів.

#### 3. Timer2:

- Розрядність: 8 біт
- Режими роботи: Нормальний, ШІМ (Fast PWM, Phase Correct PWM), СТС

- Джерела тактового сигналу: Внутрішній крок (з визначником), зовнішній крок, асинхронний режим з зовнішнім кварцом на 32.768 kHz
- Особливості: Може працювати в асинхронному режимі з зовнішнім низькочастотним кварцом, що робить його корисним для реалізації реального часу (RTC).

Основні режими роботи таймерів:

- Нормальний режим: Лічильник збільшується від 0 до максимально можливого значення, після чого скидається і генерує переповнення.
- CTC (Clear Timer on Compare Match): Лічильник збільшується до значення, встановленого в регістрі порівняння (OCRnA), після чого скидається і може генерувати переривання.
- ШІМ (PWM): Використовується для генерації широтно-імпульсної модуляції. Існують різні режими ШІМ, такі як Fast PWM та Phase Correct PWM, які дозволяють генерувати стабільні вихідні сигнали для керування потужністю або яскравістю світлодіодів.

Регістра для налаштування таймерів

- TCCRnA та TCCRnB (Timer/Counter Control Registers): Використовуються для налаштування режиму роботи та джерела тактового сигналу.
- TCNTn (Timer/Counter Register): Містить поточне значення лічильника таймера.
- OCRnA та OCRnB (Output Compare Registers): Використовуються для встановлення значень порівняння в режимах CTC та ШІМ.
- TIMSKn (Timer Interrupt Mask Register): Дозволяє переривання по подіям таймера.
- TIFRn (Timer Interrupt Flag Register): Флаги переривань таймера.

### Застосування таймерів

- Генерація ШІМ: Використовується для керування швидкістю двигунів, яскравістю світлодіодів та іншими додатками, де необхідно регулювати потужність.
- Відлік часу: Використовується для створення затримок, таймерів реального часу та годинників.
- Події: Використовується для відстеження певних подій через переривання, таких як обробка імпульсів з датчиків.

### Висновки за розділом 2

Проаналізувавши всі аспекти, пов'язані з вибором мікроконтролера для створення програмної моделі ШІМ, я можу зробити наступні висновки:

ATmega328 має ряд переваг, які роблять його оптимальним вибором для цього завдання:

- Гнучкість програмування: AVR-архітектура та підтримка мови C дозволяють легко налаштувати алгоритми ШІМ та адаптувати їх до конкретних потреб проекту.
- Вбудовані функції: Наявність вбудованого PWM-модуля та таймерів полегшує реалізацію ШІМ без необхідності додаткових зовнішніх компонентів.
- Низька вартість: ATmega328 доступний за помірною ціною, що робить його економічно вигідним рішенням для різноманітних проектів.
- Низьке споживання енергії: Енергоефективність мікроконтролера робить його ідеальним для пристроїв, що працюють від акумуляторів або в енергозберігаючих системах.
- Широке співтовариство користувачів: Завдяки своїй популярності ATmega328 має велике співтовариство користувачів та розробників, що дає доступ до безлічі ресурсів та підтримки.

Окрім переваг, ATmega328 має й деякі обмеження, які слід враховувати:

- 8-бітна архітектура: Ця архітектура може обмежувати продуктивність для складних проектів з високими вимогами до точності та швидкості ШІМ.
- Обмежена пам'ять: 32 КБ Flash-пам'яті та 2 КБ SRAM можуть бути недостатньо для програмних моделей ШІМ з великою кількістю даних або алгоритмів.
- Відсутність вбудованих аналогових модулів: Для складних аналогових обчислень, необхідних у деяких ШІМ-системах, можуть знадобитися додаткові зовнішні компоненти.

Незважаючи на ці обмеження, ATmega328 залишається чудовим вибором для створення програмної моделі ШІМ завдяки своїй простоті, гнучкості, доступності та широкому спектру функцій.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ ШІМ КОНТРОЛЕРА

#### 3.1 Рекомендації щодо впровадження і експлуатації

Рекомендації щодо впровадження і експлуатації програмної моделі ШІМ контролера включають:

1. Початкове налаштування та калібрування:
  - Провести ретельне калібрування контролера відповідно до специфікацій системи та навантаження.
  - Використовувати тестові сценарії для перевірки точності та стабільності роботи контролера перед введенням в експлуатацію.
2. Моніторинг та діагностика:
  - Впровадити системи моніторингу для постійного відстеження роботи контролера та виявлення можливих збоїв.
  - Регулярно проводити діагностику для виявлення та усунення проблем на ранніх стадіях.
3. Умови експлуатації:
  - Забезпечити роботу контролера в рекомендованих температурно-вологісних умовах.
  - Захищати обладнання від механічних впливів та електромагнітних завад.
4. Технічне обслуговування:
  - Проводити регулярне технічне обслуговування, включаючи очищення, перевірку з'єднань та оновлення програмного забезпечення.
  - Здійснювати планові перевірки системи відповідно до графіка обслуговування.
5. Оптимізація та оновлення:

- Періодично аналізувати ефективність роботи контролера та за необхідності вносити коригування для оптимізації роботи.
- Впроваджувати оновлення програмного забезпечення для покращення функціональності та безпеки контролера.

### 3.2 Створення схеми в симуляторе SimulIDE

Для створення схеми яка буде керувати ШІМ необхідно врахувати декілька параметрів:

1. Потрібно використовувати Timer1. Так як потрібно змінювати розрядність від 2 і до 16, а Timer0 і Timer2 мають максимальну розрядність 8-біт.
2. Потрібно надати можливість змінювати розрядність.
3. Потрібно надати можливість змінювати шпаруватість.
4. Потрібно надати можливість змінювати значення розділювача.

Так як ми обрали Timer1, то для виводу сигналу можна використовувати тільки піни 9 та 10 (в межах ATmega328 вони називаються V1 та V2). Для зручності побудови схеми ми будемо використовуват пін 9 (V1).

Далі потрібно можливість керування, для цього будемо використовувати звичайні кнопки.

Нехай піни D0-D4 будуть для зміни значення розділювача:

- D0 – розділювач = 1;
- D1 – розділювач = 8;
- D2 – розділювач = 64;
- D3 – розділювач = 256;
- D4 – розділювач = 1024.

Піни C0-C1 будуть для змінення шпаруватості:

- C0 – зменшує шпаруватість;
- C1 – збільшує шпаруватість.

Піни C2-C3 будуть для змінення розрядності:

- C2 – зменшує розрядність;
- C3 – збільшує розрядність.

Для показу працездатності до пiна В1 пiдключемо осцилограф i свiтлодiод iз колекторним двигуном через транзистор.

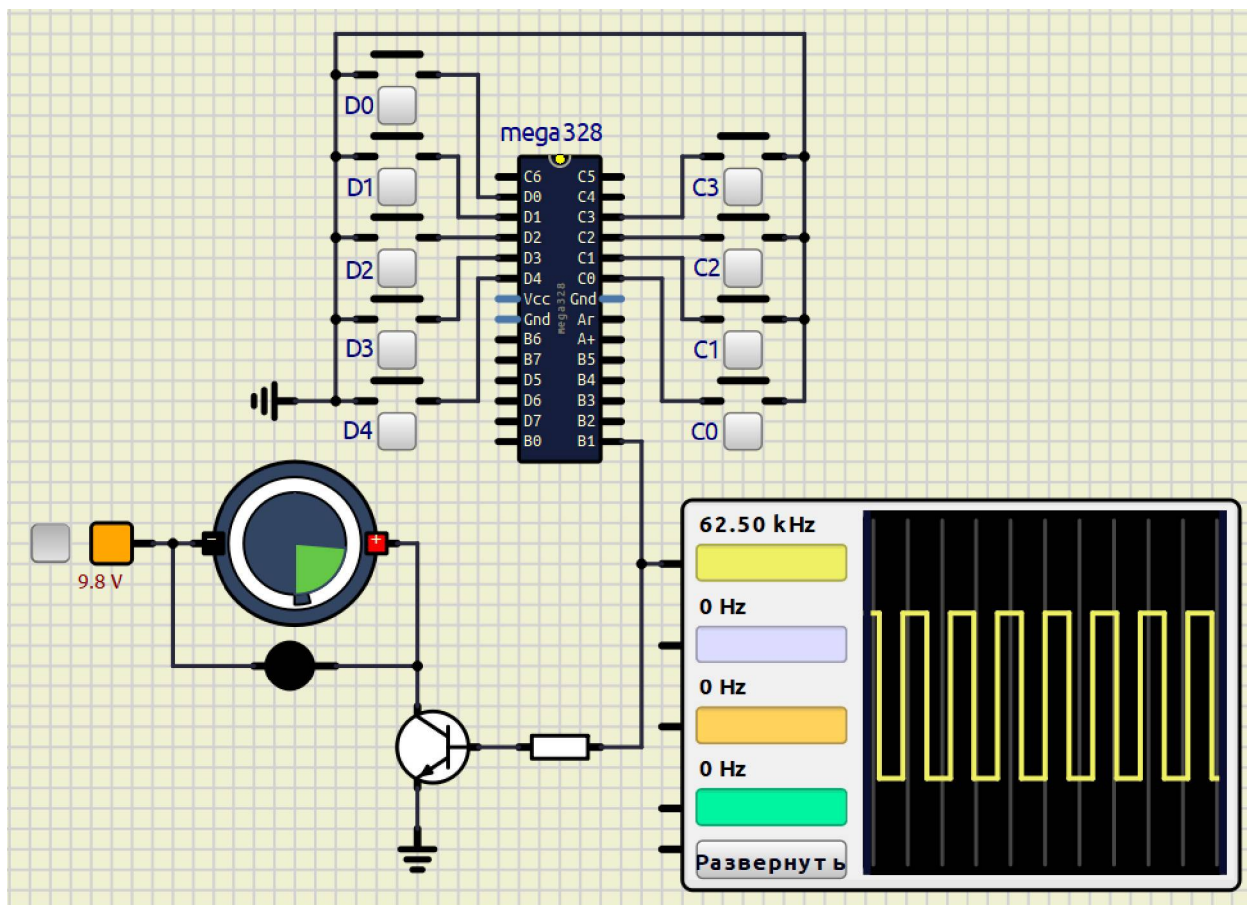


Рисунок 3.1 – Схема ШІМ контролера

Щоб не захаращувати схему зайвими резисторами використовуватиму внутрішній резистор мікроконтролера, тому підключаю всі кнопки до землі.

### 3.3. Написання коду

#### 3.3.1 Налаштування таймера

Мікроконтролер може створювати декілька варіантів ШІМ сингала: звичайний ШІМ, швидкий ШІМ та їх інвертовані варіації. Я буду використовувати швидкий не інвертований ШІМ, так як більш продуктивний.

Для цього потрібно налаштувати мікроконтролер на створення швидкого ШІМ сигналу на основі Timer1. Тому потрібно використовувати регістри TCCR1A і TCCR1B вони дозволять налаштувати Timer1.

Для початку налаштуємо регістр TCCR1A:

Bit	7	6	5	4	3	2	1	0	
(0x8D)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рисунок 3.2 – Структура регістра TCCR1A [8]

Для початку необхідно 7-4 біти, відштовхуватися буду від таблиць взятих із Datasheet (Табл. 3.1).

Таблиця 3.1

#### Режим виведення, швидка ШІМ [8]

COM1A1/COM1B1	COM1A1/COM1B1	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Виходячи з таблиці виходить що 7 та 5 біт = 1, а 6 та 4 біт = 0.

Третій та другий біт залишу нулями так як вони не впливають ні на що.

Останні два біта буду налаштовувати відштовхувались від табл.3.2.

Таблиця 3.2

**Режим генерації сигналу опис біта [8]**

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8- bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9- bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10- bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Режими які налаштовують таймер на швидкий ШІМ це: 5, 6, 7, 14 та 15.  
Із них я виберу 14 режим так як, 5, 6 та 7 не дозволяють самому налаштовувати

верхню межу. Я не обрав 15 режим тому що хочу щоб за верхню межу відповідав регістр ICR1.

Тоді виходить що регістр **TCCR1A = 0b10100010**.

Тепер необхідно налаштувати регістр TCCR1B

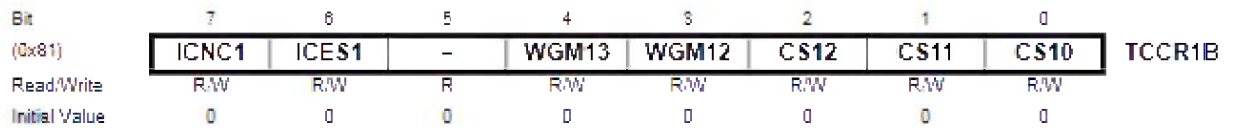


Рисунок 3.3 – Структура регістра TCCR1B [8]

Біт 7 – відповідає за прут Capture Noise Canceler, він не потрібен, ставимо в 0.

Біт 6 – відповідає за Input Capture Edge Select, він не потрібен, ставимо в 0.

Біт 5 – залишаємо в 0.

Біти 3 та 4 – дивимось в таблиці 3.2. Виходить що Біт 3 та 4 ставлю в 1.

Біти 0-2 – відповідають за розділювач. Їх значення дивимось в табл. 3.3

*Таблиця 3.3*

**Таблиця дільників[8]**

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	1
0	1	0	8
0	1	1	64
1	0	0	256
1	0	1	1024
1	1	0	External clock source on T1 pin. Clock on falling edge
1	1	1	External clock source on T1 pin. Clock on rising edge

Установим їх в початкове значення: біт 2 = 0, біт 1 = 0, біт 0 = 1.

Тоді виходить що регістр **TCCR1B = 0b00011001**

Ще потрібно налаштувати ICR1 та OCR1A.

OCR1A буде відповідати за шпаруватість, початкове значення поставлю в 0.

ICR1 відповідає верхньому значенню. Кількість бітів може бути від 2 і до 16, візьму середнє значення – 8. Тоді  $ICR1 = 2^8 - 1 = 255$ .

**BITNESS = 8;**

**ICR1 = (1<<BITNESS)-1;**

**OCR1A = 0;**

### 3.3.2 Налаштування пінів

Потрібно налаштувати пін B1 на вихід, а піни C0 – C3 і D0 – D4 на вихід:

$DDRB |= (1<< DDB1);$

$DDRC \&= \sim(1<<DDC0);$

$DDRC \&= \sim(1<<DDC1);$

$DDRC \&= \sim(1<<DDC2);$

$DDRC \&= \sim(1<<DDC3);$

$DDRD \&= \sim(1<<DDD0);$

$DDRD \&= \sim(1<<DDD1);$

$DDRD \&= \sim(1<<DDD2);$

$DDRD \&= \sim(1<<DDD3);$

$DDRD \&= \sim(1<<DDD4);$

Ще необхідно включити підтягуючий резистор:

$PORTC |= (1<<PORTC0);$

$PORTC |= (1<<PORTC1);$

$PORTC |= (1<<PORTC2);$

$PORTC |= (1<<PORTC3);$

```
PORTD |= (1<<PORTC0);
```

```
PORTD |= (1<<PORTC1);
```

```
PORTD |= (1<<PORTC2);
```

```
PORTD |= (1<<PORTC3);
```

```
PORTD |= (1<<PORTC4);
```

Вводи будуть інвертовані, тому потрібно буде інвертувати при порівнянні в функціях, наприклад :

```
if (!(PIND&(1<<PIND0)))
```

### **Висновки за розділом 3**

Для створення програмної моделі ШІМ контролера потрібно мати не тільки базові знання про створення схем з участю мікроконтролерів, а й глибоке розуміння їх функціонування та правил використання. Це включає знання про апаратні можливості конкретного мікроконтролера, такі як доступні таймери, піни та їх функціональні можливості. Крім того, важливо мати навички налаштування цих компонентів з урахуванням конкретних цілей програми, зокрема, правильно встановлювати параметри таймерів для реалізації ШІМ та налаштовувати піни для взаємодії зі зовнішніми пристроями.

## ВИСНОВКИ

Дослідження демонструє, що розробка та моделювання програмної моделі ШІМ контролера є ефективним підходом для управління потужністю в різноманітних системах. Аналіз отриманих результатів підтверджує, що впровадження цієї моделі дозволяє значно покращити ефективність роботи систем за рахунок точнішого та гнучкішого регулювання потужності.

Однією з ключових переваг програмної моделі ШІМ контролера є можливість адаптації до змінних умов експлуатації та різноманітних типів навантажень. Це забезпечує підвищену стабільність системи та зменшує втрати енергії, що є критичним фактором у сучасних умовах енергозбереження. Оптиміальні стратегії регулювання, розроблені на основі цієї моделі, дозволяють досягти значного зниження енергоспоживання без втрати продуктивності, що сприяє підвищенню загальної ефективності системи.

Додатково, дослідження підкреслює важливість інтеграції програмних методів управління з апаратними компонентами для досягнення максимального синергетичного ефекту. Це забезпечує не тільки підвищення ефективності, але й покращує здатність діагностувати та запобігати несправностям, що важливо для тривалої роботи систем.

Загалом, це дослідження демонструє перспективність використання програмно керованих ШІМ контролерів у широкому спектрі застосувань, від побутової електроніки до промислових систем. Високий потенціал цієї технології для підвищення енергоефективності та надійності систем робить її важливим напрямком для подальших досліджень та розвитку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Блум, Джереми - "Дослідження Arduino: Інструменти та техніки для інженерних чарів"
- 2) Вільямс, Елліот - "Make: Програмування AVR: Навчання написанню програмного забезпечення для апаратного забезпечення"
- 3) Еванс, Брайан - "Основи програмування на Arduino"
- 4) Крафт, Брок - "Проекти на Arduino для чайників"
- 5) Марголіс, Майкл - "Книга рецептів Arduino" Про
- 6) Ардуино и не только [Електронний ресурс]. – режим доступу: URL: <https://tsibrov.blogspot.com/2019/02/atmega328p.html> (дата звернення – 19.05.2024)
- 7) Широтно-імпульсна модуляція [Електронний ресурс]. – режим доступу: URL: [https://ukrayinska.libretexts.org/Робоча\\_сила/Технологія\\_електроніки/Книга%3A\\_Електричні\\_схеми\\_III\\_-\\_Напівпровідники\\_\(Kuphaldt\)/11%3A\\_Приводи\\_двигуна\\_постійного\\_струму/11.01%3A\\_Широтно-імпульсна\\_модуляція](https://ukrayinska.libretexts.org/Робоча_сила/Технологія_електроніки/Книга%3A_Електричні_схеми_III_-_Напівпровідники_(Kuphaldt)/11%3A_Приводи_двигуна_постійного_струму/11.01%3A_Широтно-імпульсна_модуляція) (дата звернення – 20.05.2024)
- 8) Офіційна документація від Microchip (колишній Atmel) [Електронний ресурс]. – режим доступу: URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1424998/MICROCHIP/ATMEGA328P-ANR.html> (дата звернення – 10.05.2024)
- 9) ATmega328P Fast PWM mode Programming Examples [Електронний ресурс]. – режим доступу: URL: <https://www.ee-diary.com/2021/07/atmega328p-fast-pwm-mode-programming.html> (дата звернення – 12.05.2024)

**ДОДАТКИ****Додаток А**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**  
Галузь знань: **12 – Інформаційні технології**  
Спеціальність: **123 – Комп'ютерна інженерія.**

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної  
та прикладної системотехніки

\_\_\_\_\_ д.т.н., проф. Шматков С. І.

«21» грудня 2024 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**Циба Богдан Олегович**

\_\_\_\_\_ (прізвище, ім'я, по батькові студента)

**1. Тема роботи «ПРОГРАМНА МОДЕЛЬ ШІМ КОНТРОЛЕРА»**

керівник роботи Котвицький Альберт Тадеушевич, к.т.н., доц., доцент  
кафедри ТПС

затверджені наказом по університету від "03" Травня 2024 року №4101-5/909

**2. Строк подання студентом роботи 31 Травня 2024**

**3. Перелік питань, які потрібно розробити**

- 1) Принцип роботи ШІМ.
- 2) Схема ШІМ контролера.
- 3) Програмна модель ШІМ контролера при мінімальних та максимальних значеннях .

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз научної літератури	21.12.2023 - 25.11.2024
2	Моделювання ШІМ контролера	19.12.2023 - 2.01.2024
3	Створення схеми на симуляторі	2.01.2024 - 2.02.2024
4	Тестування та апробація комп'ютерної моделі.	3.02.2024 - 30.03.2024
5	Підготовка наукової статті	3.03.2024 - 30.04.2024
6	Розробка пояснювальної записки.	31.03.2024 - 31.05.2024

5. Дата видачі завдання \_\_\_\_\_ 21.12.2023 \_\_\_\_\_

Студент

Б. О. Циба  
ініціали, прізвище

  
підпис

Керівник роботи

А. Т. Котвицький  
ініціали, прізвище

  
підпис

## Додаток Б

Затверджую

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

Технічне завдання  
на розробку програмного виробу «ПРОГРАМНА МОДЕЛЬ ШІМ  
КОНРОЛЕРА»

1.	Введення	1.1. Назва: Програмна модель ШІМ контролера. 1.2. Галузь застосування: комп'ютерні технології, побутова електроніка, енергетика, телекомунікація.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія. 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/909 від «03» Травня 2024 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: аналіз та моделювання програмної моделі ШІМ контролера для ефективного управління потужністю в електронних системах, з метою розробки оптимальних стратегій регулювання та підвищення їхньої ефективності. 3.2. Призначення розробки полягає в підвищенні ефективності управління потужністю в різних системах шляхом використання програмної моделі ШІМ контролера для оптимізації регулювання та зниження енергоспоживання. 3.3. Вихідні дані розробки: технічні специфікації системи, параметри навантаження, вимоги до енергоефективності, початкові налаштування контролера та обмеження щодо експлуатаційних умов.
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: високу точність і стабільність регулювання потужності, адаптивність до змінних навантажень, низьке енергоспоживання та сумісність з різними типами систем. 4.2. Вимоги до надійності: немає. 4.3. Вимоги до умов експлуатації: немає. 4.4. Вимоги до складу і параметрів технічних засобів: звичайне обчислювальне обладнання, ПК, тощо. 4.5. Вимоги до інформаційної та програмної сумісності: немає.

		<p>4.6. Вимоги до маркування та упаковки: немає.</p> <p>4.7. Вимоги до транспортування і зберігання: на звичайних носіях інформації.</p> <p>4.8. Спеціальні вимоги: немає.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «ПРОГРАМНА МОДЕЛЬ ШІМ КОНТРОЛЕРА» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до дипломної роботи).</p> <p>3) Опис програми (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>В даному розділі можуть бути представлені:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		до 25 грудня 2023	Аналіз наукової літератури.
		від 25 грудня 2023 до 16 січня 2024	Розробка концептуальної моделі
		від 2 січня 2024 до 2 лютого 2024	Моделювання ШІМ контролера.
		від 3 січня 2024 до 30 березня 2024	Оптимізація стратегії управління
		від 11 лютого 2024 до 27 травня 2024	Створення схеми в симуляторі
		від 31 березня 2024 до 27 квітня 2024	Тестування та валідація комп'ютерної моделі
		від 1 травня 2024 до 28 травня 2024	

		30 травня 2024	Оформлення результатів. Написання пояснювальної записки.  Представлення кваліфікаційного проекту керівнику кваліфікаційної роботи та рецензенту.
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> <li>1. Перевірку ходу розробки програми виконувати раз в 3 тижні.</li> <li>2. Захист розробленої моделі провести на засіданні Атестаційної комісії.</li> <li>3. Пояснювальну записку подати в електронному вигляді в 1 примірнику.</li> </ol>	

Виконавець  
студент групи КІ- 41  
Циба Б. О.




---

Замовник  
д. т. н., проф.  
Котвицький А. Т.




---

## Програма і методика випробувань програмного виробу

### «ПРОГРАМНА МОДЕЛЬ ШІМ КОНТРОЛЕРА»

#### 1. Об'єкт випробувань

1. Назва програмного виробу : «ШІМ контролер зроблений на симуляторе SimulIDE с використанням мови програмування С»
2. Галузь застосування : комп'ютерні технології, побутова електроніка, енергетика, телекомунікація
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

#### 2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

#### 3. Загальні положення

##### 1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

##### 2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

##### 3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

##### 4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

#### 4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

1. працювати на основі мікроконтролера: ATmega328 ;

2. вимоги до надійності;
3. передбачити захист від некоректних дій користувача;
4. сумісність з іншими програмними продуктами;
5. зменшити об'єм програмного коду;
6. вимоги до складу і параметрів технічних засобів;
7. вимоги до маркування та упаковки (не висуваються);
8. вимоги до транспортування і зберігання (не висуваються).

Спеціальні вимоги (не пред'являються).

## **5. Вимоги до програмної документації**

Програмною документацією до виробу «ШІМ контролер зроблений на симуляторе SimulIDE с використанням мови програмування С» вважати:

1. справжнє технічне завдання на розробку програми (представити як Додаток Б до пояснювальної записки до кваліфікаційної роботи);
2. Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи);
3. рекомендацій щодо застосування створеної програмної стандартизації у проектах (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
4. Текст програми(представити як Додаток Г до пояснювальної записки до кваліфікаційної роботи).

## **6. Засоби і порядок випробувань**

### **6.1 Засоби випробувань**

Для проведення випробувань необхідний симулятор який підтримує симуляцію мікроконтролера ATmega328.

### **6.2 Порядок проведення випробувань**

Як правило, випробування проводяться в два етапи:

-ознайомчий (1-й етап);

-випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. перевірку відповідності технічних характеристик програми вимогам технічного завдання;
2. перевірку ступеня виконання функціональних вимог до програми;
3. методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

1. Програма працює відповідно до умов експлуатації мікроконтролера ATmega328.
2. Для роботи необхідний компілятор мови програмування C.
3. Порядок проведення випробувань:
  - 3.1. Запустити симуляцію за допомогою кнопки «пуск»
  - 3.2. Після запуску програми необхідно встановити необхідні нам значення змінних. (це можна зробити і через код, але це необхідно робити перед запуском і перекомпілювати)
  - 3.3. Після запуску ми побачимо що видаватиме нам осцилограф і візуалізацію продуктивності через світлодіод та двигун.

Для проведення випробувань пропонується тест 1, тест 2, тест 3 та тест 4.

#### Тест 1

1. Перевірка виконання програми;
2. Самі розрахуємо частоту та період, для значень: розрядність = 2, а значення роздільника = 1

$$\text{Частота} = \frac{16 * 10^6}{2^2} = 4 * 10^6 = 4 \text{ МГц}$$

$$T = \frac{1}{4 * 10^6} = 0,25 * 10^{-6} = 0,25 \mu\text{s}$$

3. Перевіримо правильність.

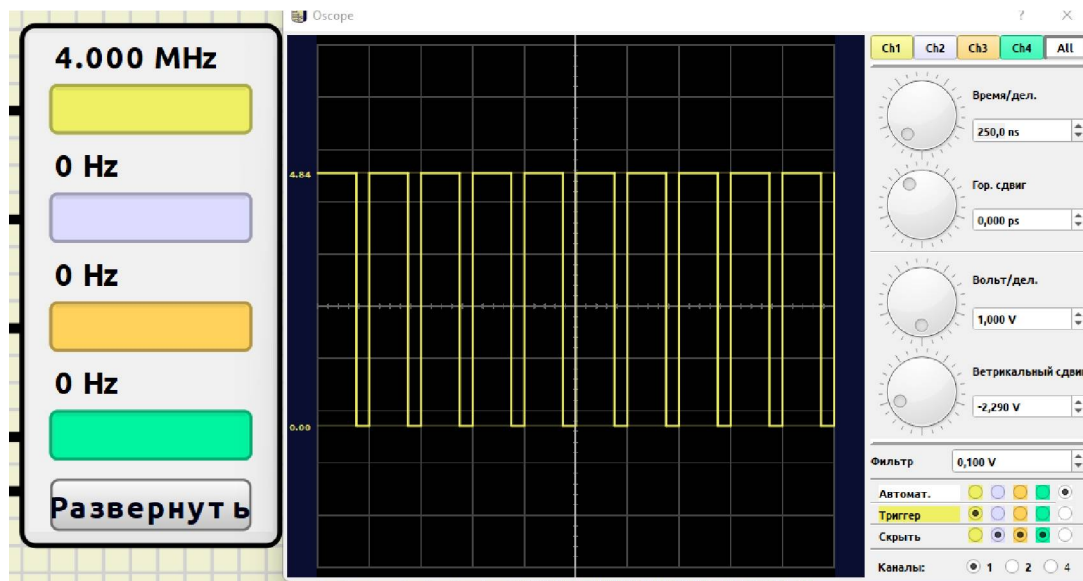


Рисунок В Тест 1

При таких значеннях маємо малу кількість рівнів сигналу, та дуже велику частоту

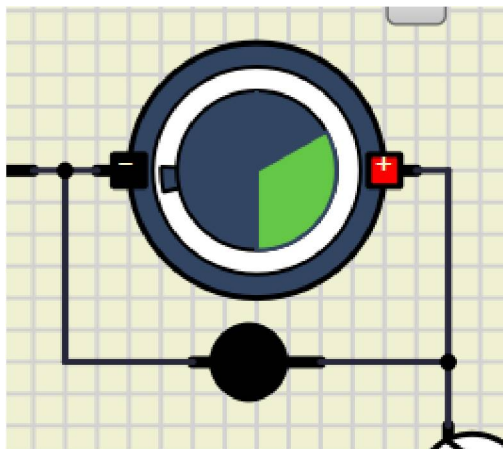


Рисунок В Тест 1 зображення двигуна

## Тест 2

1. Перевірка виконання програми;
2. Самі розрахуємо частоту та період, для значень: розрядність = 2, а значення роздільника = 1024

$$\text{Частота} = \frac{16 \cdot 10^6}{1024 \cdot 2^2} = 3,906 \cdot 10^3 = 3,906 \text{кГц}$$

$$T = \frac{1}{3,906 \cdot 10^3} = 0,25 \cdot 10^{-6} = 256 \mu\text{s}$$

3. Перевіримо правильність.

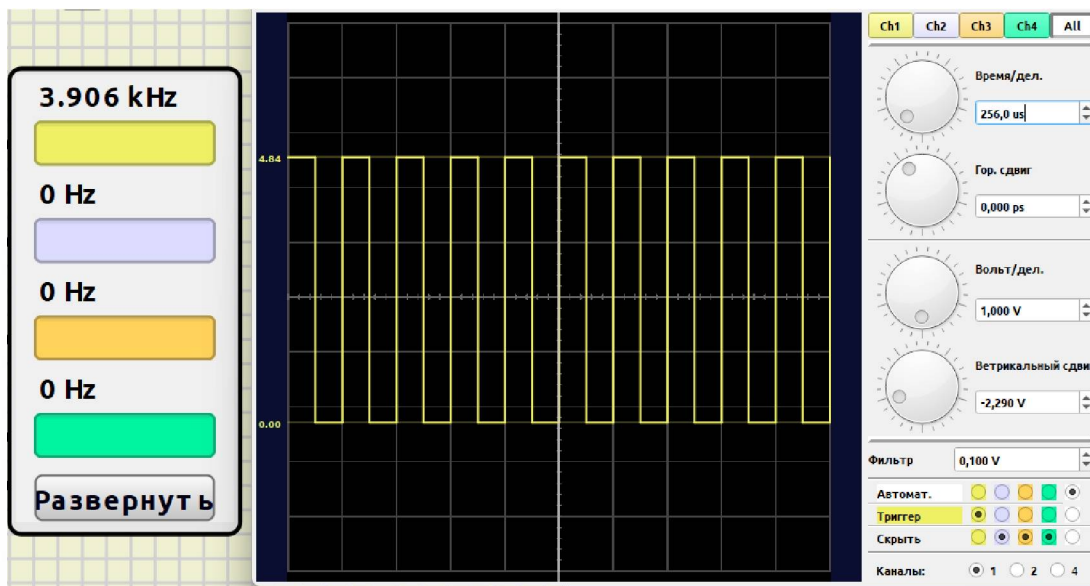


Рисунок В Тест 2

При таких значеннях маємо малу кількість рівнів сигналу, та частоту зменшену більш ніж в 1000 разів. Якість сигналу візуально не змінилась, двигун працює так само. Але споживання електроенергії зменшилось.

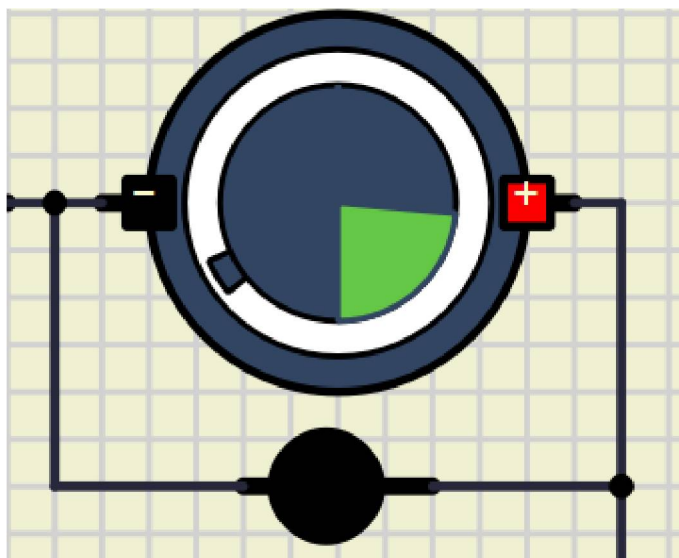


Рисунок В Тест 2 зображення двигуна

Тест 3

1. Перевірка виконання програми;

2. Самі розрахуємо частоту та період, для значень: розрядність = 16, а значення роздільника = 1

$$\text{Частота} = \frac{16 * 10^6}{2^{16}} = 244,14 \text{ Гц}$$

$$T = \frac{1}{244,14} = 4,096 \text{ ms}$$

3. Перевіримо правильність .

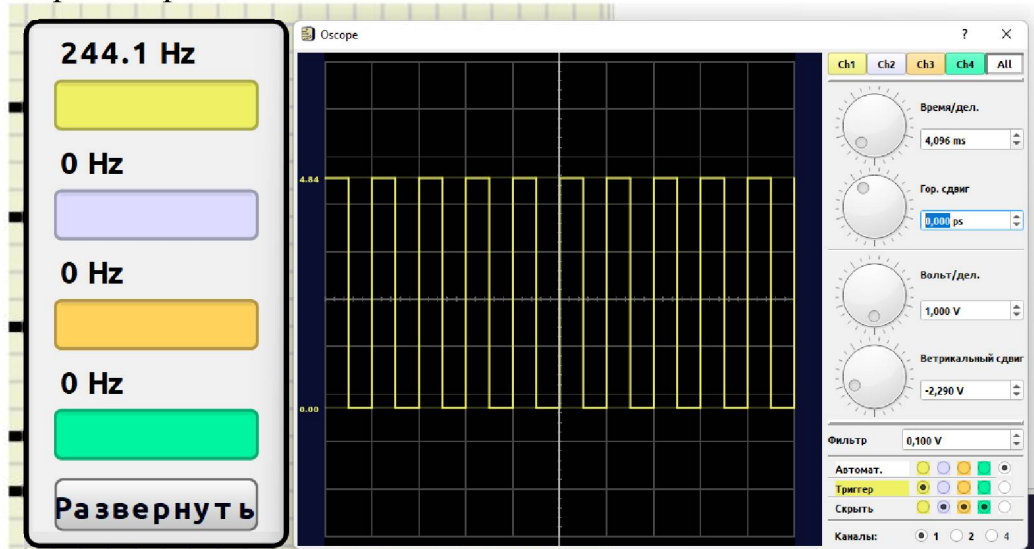


Рисунок В Тест 3

При таких значеннях маємо велику кількість рівнів сигналу, але малу частоту, що спричиняє уривчасту роботу двигуна.

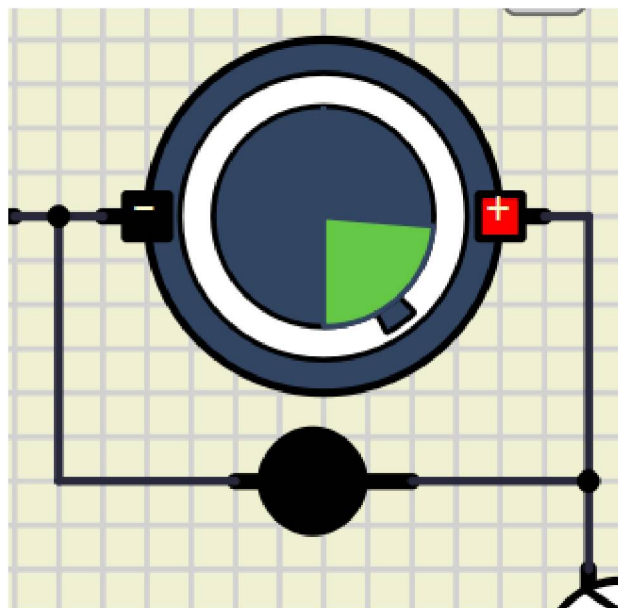


Рисунок В Тест 3 зображення двигуна

## Тест 4

1. Перевірка виконання програми;
2. Самі розрахуємо частоту та період, для значень: розрядність = 16, а значення роздільника = 1024

$$\text{Частота} = \frac{16 * 10^6}{1024 * 2^{16}} = 238,41 \text{ мГц}$$

$$T = \frac{1}{0,23841} = 4,194 \text{ s}$$

3. Перевіримо правильність .

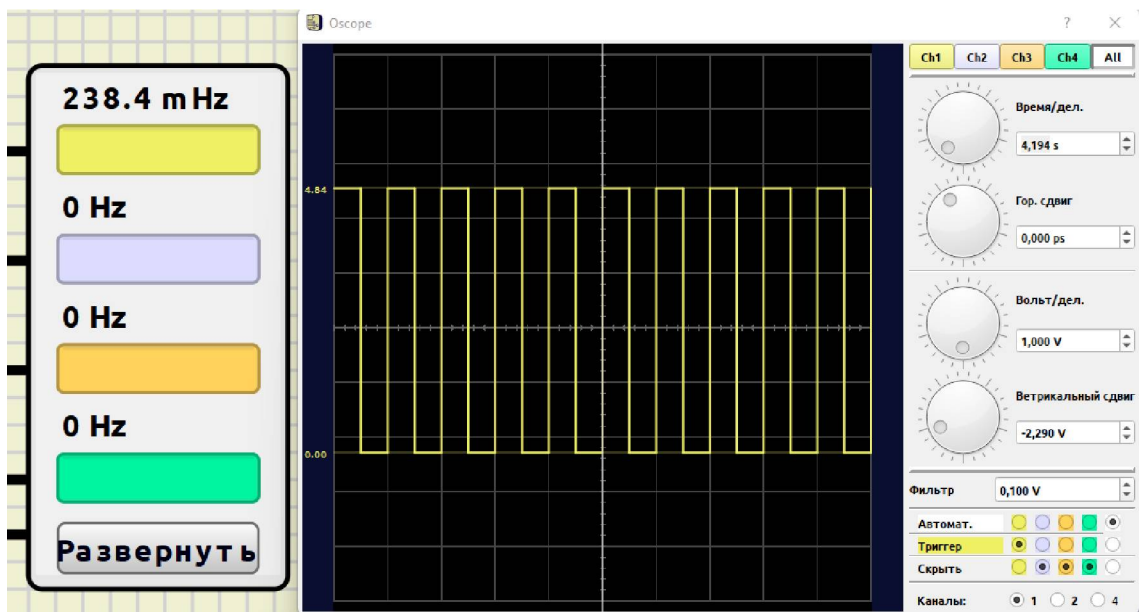


Рисунок В Тест 4

При таких значеннях маємо велику кількість рівнів сигналу, та дуже малу частоту, із-за чого двигун працює періодами. В даному випадку 2.097 с. він працює і 2.097 с. не працює

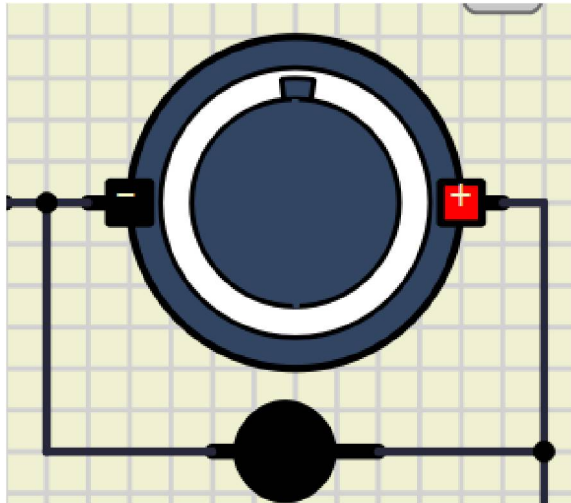


Рисунок В Тест 4 зображення двигуна в перший проміжок часу

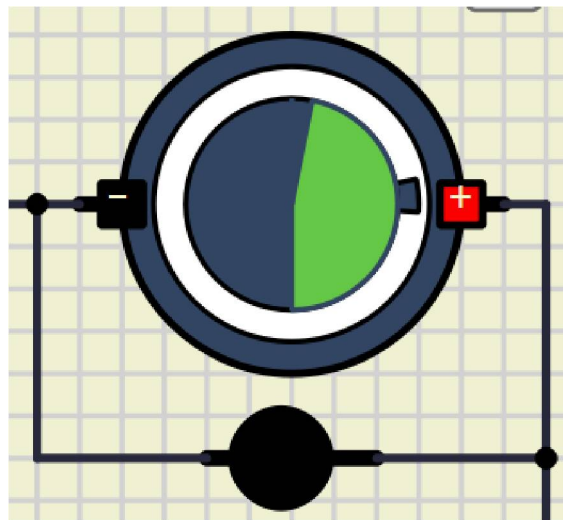


Рисунок В Тест 4 двигуна в другий проміжок часу

**Висновки:** Можна зробити такі висновки:

- **Висока розрядність** дає більшу кількість рівнів сигналу, що може покращити точність управління, але це супроводжується зменшенням частоти, що може спричинити небажані ефекти, такі як миготіння та уривчаста робота пристроїв.
- **Низька розрядність** дає меншу кількість рівнів сигналу, але дозволяє досягти високої частоти, що може бути корисним для швидких процесів, але обмежує плавність управління.
- **Значення роздільника** істотно впливає на частоту. Велике значення роздільника значно зменшує частоту, що може зменшити енергоспоживання, але водночас призводить до великих періодів роботи/відпочинку пристроїв.

- **Оптимальний вибір** параметрів залежить від конкретних вимог до системи, таких як необхідна плавність управління, частота оновлення сигналу та енергоспоживання.

Випробування пройшло успішно.

Виконавець: студент групи КІ-41, Циба Б.О.

Виконавець  
студент групи КІ- 41  
Циба Б. О.



---

### Код програми керування ШІМ контролером

```

#define F_CPU 16000000UL // Визначаємо частоту процесора 16 МГц
#include <avr/io.h> // Включаємо бібліотеку для вводу/виводу
#include <Util/delay.h> // Включаємо бібліотеку для затримок

int BITNESS = 8; // Ініціалізуємо початкову бітність

// Функція для зміни частоти
void frequencyChange()
{
    if (!(PIND & (1 << PIND0)))
    {
        TCCR1B = 0b00011001;
    }
    if (!(PIND & (1 << PIND1)))
    {
        TCCR1B = 0b00011010;
    }
    if (!(PIND & (1 << PIND2)))
    {
        TCCR1B = 0b00011011;
    }
    if (!(PIND & (1 << PIND3)))
    {
        TCCR1B = 0b00011100;
    }
    if (!(PIND & (1 << PIND4)))
    {
        TCCR1B = 0b00011101;
    }
}

uint8_t OLDBUTTONVALUE = 0; // Змінна для збереження попереднього стану
кнопки
uint8_t NEWBUTTONVALUE = 0; // Змінна для збереження нового стану кнопки

// Функція для зміни бітності
void bitChange()
{
    NEWBUTTONVALUE = PINC & 0b00001100;

    if (OLDBUTTONVALUE != NEWBUTTONVALUE)
    {
        if (!(PINC & (1 << PINC3)) && BITNESS < 16)
        {
            BITNESS += 1;
        }
    }
}

```

```

        if (!(PINC & (1 << PINC2)) && BITNESS > 2)
        {
            BITNESS -= 1;
        }

        ICR1 = (1 << BITNESS) - 1;

        if (OCR1A > ICR1)
        {
            OCR1A = ICR1;
        }
    }

    OLDBUTTONVALUE = NEWBUTTONVALUE;
}

// Функція для зміни значення OCR1A
void valueChange()
{
    if (!(PINC & (1 << PINC1)) && OCR1A < ICR1)
    {
        OCR1A += 1;
    }
    if (!(PINC & (1 << PINC0)) && OCR1A > 0)
    {
        OCR1A -= 1;
    }
}

int main(void)
{
    DDRB |= (1 << DDB1); // Встановлюємо порт B1 як вихід

    // Встановлюємо порти C0 - C3 як вхід
    DDRC &= ~(1 << DDC0);
    DDRC &= ~(1 << DDC1);
    DDRC &= ~(1 << DDC2);
    DDRC &= ~(1 << DDC3);

    // Встановлюємо порти D0 - D4 як вхід
    DDRD &= ~(1 << DDD0);
    DDRD &= ~(1 << DDD1);
    DDRD &= ~(1 << DDD2);
    DDRD &= ~(1 << DDD3);
    DDRD &= ~(1 << DDD4);

    // Включаємо підтягувальні резистори на порти C0 - C3
    PORTC |= (1 << PORTC0);
    PORTC |= (1 << PORTC1);
    PORTC |= (1 << PORTC2);

```

```
PORTC |= (1<<PORTC3);

// Включаємо підтягувальні резистори на порти D0 - D4
PORTD |= (1<<PORTC0);
PORTD |= (1<<PORTC1);
PORTD |= (1<<PORTC2);
PORTD |= (1<<PORTC3);
PORTD |= (1<<PORTC4);

TCCR1A = 0b10100010; // Налаштовуємо регістр TCCR1A для PWM
TCCR1B = 0b00011001; // Налаштовуємо регістр TCCR1B для PWM

ICR1 = (1<<BITNESS)-1; // Встановлюємо початкове значення ICR1

OCR1A = 32768; // Встановлюємо початкове значення OCR1A

while(1)
{
    valueChange();
    bitChange();
    frequencyChange();
    _delay_ms(20); // Робимо затримку 20 мс
}
}
```