

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

А.О. Новацький

Електроніка та мікропроцесорна техніка

Курсовий проєкт

*Затверджено Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів освітньої програми «Інтегровані
інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»*

Київ
КПІ ім. Ігоря Сікорського
2026

Електроніка та мікропроцесорна техніка: Курсовий проєкт [Електронний ресурс] : навч. посіб. для студ. освітньої програми «Інтегровані інформаційні системи» спеціальності 126 «Інформаційні системи та технології» / А.О. Новацький, КПП ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 6,4 Мбайт). – Київ : КПП ім. Ігоря Сікорського, 2026. – 105с.

Гриф надано Методичною радою КПП ім. Ігоря Сікорського (протокол №4 від 05.02.2026р.) за поданням Вченої ради факультету Інформатики та обчислювальної техніки (протокол № 6 від 26.01.2026р.)

Електронне мережне навчальне видання

Укладач: Новацький Анатолій Олександрович, к.т.н., доцент

ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА

Курсовий проєкт

Відповідальний

редактор:

Полторак В. П., к.т.н., доцент, КПП ім. Ігоря Сікорського, факультет інформатики та обчислювальної техніки, кафедра інформаційних систем та технологій

Рецензент:

Селіванов Віктор Львович, к.т.н., доцент, КПП ім. Ігоря Сікорського, факультет інформатики та обчислювальної техніки, кафедра обчислювальної техніки

Навчальний посібник охоплює теоретичний матеріал та практичні завдання, які необхідні для виконання курсового проєкту з дисципліни «Електроніка та мікропроцесорна техніка».

В посібнику наводяться методичні рекомендації по розробці типових мікропроцесорних пристроїв: пристрою обміну інформації та пристрою керування двигуном постійного струму.

В роботі наводяться приклади розрахунку, програмування та моделювання цих пристроїв. Наведено робочі моделі в пакеті PROTEUS, схеми алгоритмів роботи та керуючі програми.

В роботі також наводяться рекомендації по оформленню текстової та графічної документації. Тематика посібника відповідає «сілабусу» дисципліни, яка є обов'язковою у навчальному плані підготовки бакалаврів зі спеціальності 126 «Інформаційні системи та технології». Робота може бути корисною студентам відповідних спеціальностей при вивченні дисциплін, пов'язаних із використанням окремих програмованих модулів мікроконтролерів, а також при виконанні бакалаврських, курсових та магістерських робіт, в яких використовуються відповідні пристрої.

Останнє було враховано при оформленні роботи, яке виконано згідно вимог до конструкторської документації.

ЗМІСТ

1 ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВИХ ПРОЄКТІВ	7
1.1 Мета і завдання курсового проєкту.....	7
1.2 Основна тематика курсового проєктування	7
1.3 Загальні вимоги до обсягу курсових проєктів	8
1.4 Вимоги до змісту та методичні вказівки до написання окремих текстових документів	9
1.5 Вимоги до оформлення текстової документації	12
1.5.1 Загальні вимоги	12
1.5.2 Нумерація формул і оформлення посилань на них	14
1.5.3 Оформлення ілюстрацій (рисуноків)	15
1.5.4 Оформлення таблиць	16
1.5.5 Оформлення додатків до ПЗ	16
1.6 Вимоги до змісту та оформлення графічної документації.....	17
1.6.1 Зміст графічної документації.....	17
1.6.2 Вимоги до оформлення графічної документації	18
1.7 Порядок комплектування документів	19
2 ПРОЄКТУВАННЯ ПРИСТРОЮ ОБМІНУ ІНФОРМАЦІЄЮ МІЖ МОДУЛЕМ УСАПП ТА МІКРОКОНТРОЛЕРОМ.....	20
2.1 Завдання	20
2.2 Розробка та опис робочої моделі в пакеті PROTEUS.....	21
2.2.1 Робоча модель пристрою.....	21
2.2.2 Опис робочої моделі.....	21
2.3 Розрахунок модуля УСАПП	22
2.3.1 Розрахунок швидкості обміну	22
2.3.2 Програмування регістра контролера швидкості передачі UBRR	24
2.3.3 Програмування регістра UCSRA	26

2.3.4 Програмування регістра UCSRC	30
2.3.4 Програмування регістра UCSRB	34
2.3.5 Програмування регістра даних UDR.....	36
2.4 Схема алгоритму роботи пристрою	37
2.5 Керуюча програма мовою Асемблер	38
2.5.1 Лістинг керуючої програми мовою Асемблер	38
2.5.2 Компіляція робочої програми та отримання hex-файлу	42
2.6 Моделювання пристрою в пакеті PROTEUS	42
2.6.1 Налаштування віртуального терміналу	42
2.6.2 Налаштування світлодіодів на цифровий режим	43
2.6.3 Запуск процесу моделювання	43
2.6.4 Встановлення двох точок зупинки під час виконання програми.....	44
2.6.5 Введення символу, який надсилається від віртуального терміналу	44
2.6.6 Введення символу, який надсилається від мікроконтролера через УАПІ	49
2.6.7 Визначення отриманої швидкості передачі.....	51
2.7 Розробка та опис структурної схеми пристрою	52
2.8 Розробка текстових документів до роботи	54
3 ПРОЄКТУВАННЯ ПРИСТРОЮ КЕРУВАННЯ ДВИГУНОМ ПОСТІЙНОГО СТРУМУ	55
3.1 Завдання	55
3.2 Розробка та опис робочої моделі	56
3.2.1 Робоча модель пристрою	56
3.2.2 Опис моделі	57
3.2.3 Опис окремих елементів моделі	63
3.2.3.1 Мікроконтролер.....	63
3.2.3.2 Модуль таймера.....	63
3.2.3.3 Двигун	64
3.2.3.4 Драйвер ШІМ.....	64

3.3 Розрахунок ШІМ-модуля та програмування його мовою Асемблер.....	68
3.4 Схема алгоритму роботи пристрою	71
3.4.1 Схема алгоритму роботи пристрою, що відповідає завданню	71
3.4.2 Особливості схем алгоритмів, які відповідають вихідним даним на виконання проєкту згідно з відповідним варіантом.....	72
3.5 Розробка окремих фрагментів програми мовою Асемблер.....	73
3.5.1 Зупинка таймера.....	73
3.5.2 Завантаження регістра TCCR1A.....	76
3.5.3 Програмування ліній PB5, PB1, PB0 на виведення	77
3.5.4 Завантаження регістра ICR1	80
3.5.5 Завантаження регістра OCR1A.....	80
3.5.6 Програмування $K_{\text{діл}} = N = 8$, режиму роботи номер 8 та запуск таймера T/C1.....	81
3.6 Керуюча програма мовою C.....	82
3.6.1 Керуюча програма мовою C, яка відповідає завданню.....	82
3.6.2 Компіляція робочої програми, отримання hex-файлу та його завантаження у пам'ять комп'ютера.....	84
3.6.3 Особливості керуючої програми, яка відповідає вихідним даним на виконання роботи згідно з відповідним варіантом	84
3.7 Моделювання пристрою в пакеті PROTEUS	88
3.7.1 Моделювання пристрою, який розроблено згідно завданню на КП.....	88
3.7.2 Моделювання пристрою, який відповідає вихідним даним на виконання роботи згідно з відповідним варіантом завдання на КП.....	92
3.8 Розробка та опис структурної схеми пристрою	93
3.9 Розробка текстових документів до роботи.....	95
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96
Додаток А Перелік тем курсових проєктів для реальних пристроїв	98
Додаток Б Приклад оформлення титульного аркуша	100
Додаток В Приклад оформлення завдання на курсовий проєкт	101

Додаток Г Приклад оформлення змісту до ПЗ курсового проєкту.....	103
Додаток Д Приклад оформлення текстового документу	105

1 ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВИХ ПРОЄКТІВ

1.1 Мета і завдання курсового проєкту

Мета курсового проєкту – закріпити теоретичні положення курсу «ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА», навчитися застосовувати отримані знання для практичних розрахунків, придбати навички проєктування окремих електронних вузлів інформаційних систем. Виконання курсового проєкту є підготовкою до дипломної бакалаврської роботи. При цьому студент на основі теоретичних знань, отриманих в результаті вивчення курсу та виконання лабораторних робіт, повинен проаналізувати один із запропонованих пристроїв, розрахувати основні елементи, а також розробити електричну структурну схему, схему алгоритму роботи, робочу програму та схему моделювання пристрою, який проєктується.

Робота над курсовим проєктом повинна бути завершена протягом шістнадцяти тижнів від дня видачі завдання.

Студент захищає проєкт у встановлені кафедрою терміни перед комісією, до складу якої входять викладачі кафедри. При захисті студент повинен продемонструвати необхідні знання, здатність орієнтуватися в теоретичному курсі «ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА» і суміжних дисциплінах, вміння логічно мислити, повинен переконливо обґрунтовувати свою точку зору.

Захищений проєкт зберігається на кафедрі протягом двох років з моменту захисту.

1.2 Основна тематика курсового проєктування

Темою курсового проєкту може бути в залежності від завдання, отриманого від керівника на початку семестру, розробка одного з двох типових пристроїв:

а) проектування пристрою обміну інформацією між модулем УСАПП та мікроконтролером;

б) проектування пристрою керування двигуном постійного струму.

Крім того, останнім часом збільшується число реальних курсових проєктів, виконуваних студентами всіх форм навчання за їх бажанням. Темою цих проєктів мають бути розробки і дослідження реальних електронних пристроїв інформаційних систем, що застосовуються в науково-дослідних розробках кафедри і на промислових підприємствах (додаток А). При виконанні таких проєктів може бути виготовлено макет пристрою та проведено його дослідження. З огляду на практичне застосування цих розробок, бажано досліджувати проєктований пристрій на патентну чистоту і знайти оригінальні рішення, здатні служити предметом заявки на винахід. Виконання таких робіт буде враховано при отриманні оцінки.

1.3 Загальні вимоги до обсягу курсових проєктів

Курсовий проєкт містить текстову та графічну документацію, яка повинна бути виконана у відповідності до вимог державних стандартів і єдиної системи конструкторської документації (ЄСКД).

Текстова документація складається з титульного листа, завдання на курсовий проєкт та пояснювальної записки (ПЗ).

Обсяг текстової документації повинен становити 25-30 сторінок тексту формату А4, якщо вона набрана на комп'ютері 14-м шрифтом з полуторним міжрядковим інтервалом, в тому числі:

- а) титульний аркуш – 1 аркуш (додаток Б);
- б) завдання на курсовий проєкт – 1-2 аркуші (додаток В);
- в) пояснювальна записка – 20-25 аркушів.

Графічна документація визначається завданням на КП і може містити структурну та функціональну електричні схеми (якщо вони не наведені в

пояснювальній записці); схему електричну принципову об'єкта проєктування; перелік елементів, якщо він не поміщений на листі з принциповою схемою (для реальних проєктів); ілюстраційний матеріал (часові діаграми, графіки, експериментальні дані, результати моделювання).

1.4 Вимоги до змісту та методичні вказівки до написання окремих текстових документів

Титульний аркуш

Титульний аркуш оформлюється згідно прикладу, який наведено в додатку Б.

Завдання на курсовий проєкт

Завдання на курсовий проєкт містить вихідні дані до проєкту. Його підписують керівник проєкту і студент. Завдання оформляють у відповідності з вимогами, загальними для НТУУ «КПІ». Приклад оформлення завдання на курсовий проєкт наведено у додатку В.

Пояснювальна записка

Пояснювальна записка (ПЗ) до курсового проєкту повинна починатися змістом, в якому наводяться найменування основних розділів і підрозділів з зазначенням номера аркуша (не нумерується). Приклад змісту наведено в додатку Г.

В загальному випадку ПЗ повинна містити наступні основні розділи:

- а) вступ (не нумерується). Коротко обґрунтувати актуальність теми проєкту, поставити завдання розробки та вказати, на підставі яких документів розроблено проєкт;

- б) призначення та область застосування пристрою, що проєктується. Сформулювати призначення пристрою і стисло описати область його застосування;
- в) технічна характеристика. Перелічити основні параметри та характеристики виробу;
- г) огляд існуючих рішень і обґрунтування вибору обраного принципу побудови пристрою. Привести коротку характеристику існуючих методів, проаналізувати ці методи і зробити відповідні висновки, що дозволяють сформулювати принципи побудови виробу; коротко охарактеризувати аналогічні пристрої, розроблені в Україні і за кордоном; провести порівняльний аналіз технічних і економічних характеристик (переваги і недоліки). На підставі цього аналізу показати, які елементи відомих аналогічних пристроїв можуть бути використані при проєктуванні заданого виробу і з яких елементів слід знайти більш ефективні рішення. Чітко сформулювати прийняті в роботі принципи побудови пристрою. Особливу увагу при цьому звернути на показники економічності, надійності, їх відповідність рівню міжнародних стандартів і т. ін.;
- д) розробка та опис структурної та функціональної (не обов'язково) електричних схем виробу. На підставі результатів попередніх розділів скласти структурну і функціональну електричні схеми, а також коротко описати роботу пристрою за структурною та функціональною схемами;
- е) вибір і обґрунтування окремих вузлів та елементів. Здійснюється вибір окремих вузлів та елементів пристрою з урахуванням їх електричних, механічних та економічних показників, вирішуючи завдання вибору з максимальним використанням уніфікованих і стандартних елементів;
- ж) розрахунки та експериментальні матеріали, що підтверджують працездатність і надійність виробу. Обсяг і характер розрахункової частини роботи визначається в кожному конкретному випадку стосовно до теми проєкту. В залежності від тематики проєкту розрахункова

частина може містити оцінку основних показників ефективності об'єкту проектування; синтез та мінімізацію логічної структури пристрою; варіантне опрацювання об'єкту проектування з оптимізацією (пошуком оптимального варіанта) при наявності обмежень з використанням різних методів оптимізації; електричний розрахунок окремих вузлів і елементів принципової схеми виробу,и та інші розрахунки;

и) розробка та опис принципової електричної схеми об'єкта проектування.

Принципову електричну схему розробляють на підставі даних попередніх розділів. При цьому необхідно прагнути до використання елементів, що серійно випускаються. Опис роботи пристрою за принциповою схемою бажано супроводжувати часовими діаграмами;

і) розробка та опис схеми алгоритму роботи пристрою та керуючої програми;

к) моделювання розробленого пристрою в одному з сучасних пакетів, наприклад, PROTEUS;

л) висновок (не нумерується). Коротко викласти основні результати виконаного проєкту; вказати практичні рекомендації щодо подальшого вдосконалення виробу (якщо це можливо), його впровадження та застосування у промисловості; дати оцінку техніко-економічної ефективності, яка може бути отримана при використанні результатів даної роботи; відзначити відповідність параметрів виробу технічному завданню на проєктування (при невідповідності, обґрунтувати і пояснити причину його виникнення);

м) список літератури, складений в порядку появи посилань на неї в тексті ПЗ, який має містити повний перелік літератури, використаної при розробці об'єкту проєктування. Список повинен бути оформлений відповідно до вимог ГОСТ 7.1-84.

Крім зазначених розділів ПЗ містить додатки, в які включають:

— результати моделювання;

- програми і результати розрахунків на ЕОМ;
- ілюстраційні матеріали.

1.5 Вимоги до оформлення текстової документації

1.5.1 Загальні вимоги

Текстову документацію оформлюють на листах формату А4, з розрахунку не більше 30 рядків на сторінці з форматуванням тексту за шириною (шрифт TIMES NEW ROMAN - 14; інтервал — полуторний). Порядок побудови розділів і підрозділів, правила оформлення тексту, розрахунків, а також оформлення рисунків, написання формул, цифр, та побудови таблиць повинні відповідати вимогам ДСТУ 3008:2015.

Відстань від верхнього або нижнього рядка тексту до верхньої або нижньої межі формату повинна бути не менше 20 мм.

Абзаци в тексті починають з відступом, рівним 12,5 мм.

Текст документа при необхідності розділяють на розділи та підрозділи. Розділи повинні мати порядкові номери в межах всього документа (частини), позначені арабськими цифрами без крапки і записані з абзацу. Підрозділи повинні мати нумерацію в межах кожного розділу. Номери підрозділів складаються з номерів розділу і підрозділу, розділених крапкою. В кінці номера підрозділу крапка не ставиться.

Розділи, як і підрозділи, можуть складатися з декількох пунктів. Якщо документ не має підрозділів, то нумерація пунктів в ньому повинна бути в межах кожного розділу, та номер пункту повинен складатися з номерів розділу і пункту, розділених крапкою. Наприкінці номера пункту крапка не ставиться. Якщо документ має підрозділи, то нумерація пунктів повинна складатися з номерів розділу, підрозділу і пункту, розділених крапками. Якщо розділ складається з одного пункту, він також нумерується. Якщо текст документа поділяється тільки на пункти, то вони нумеруються порядковими номерами у межах документа.

Пункти, при необхідності можуть бути розбиті на підпункти, які повинні мати порядкову нумерацію в межах кожного пункту, наприклад: 4.2.1.1, 4.2.1.2, 4.2.1.3 і т. ін.

В середині пунктів або підпунктів можуть бути приведені перерахування. Перед кожною позицією перерахування варто ставити дефіс або, при необхідності посилання в тексті документа на одне з перерахувань, маленьку літеру, після якої ставиться дужка. Для подальшої деталізації перерахувань необхідно використовувати арабські цифри, після яких ставиться дужка.

Кожен пункт, підпункт та перерахування записують з абзацу.

Розділи та підрозділи повинні мати заголовки. Пункти, як правило, заголовків не мають. Заголовки повинні чітко і коротко відображати зміст розділів, підрозділів. Заголовки слід друкувати з прописної літери, без крапки в кінці, не підкреслюючи. Перенесення слів у заголовках не допускаються. Якщо заголовок складається з двох речень, їх розділяють крапкою.

Відстань між заголовком і текстом повинна бути – один рядок. Відстань між заголовками розділу і підрозділу – 12 інтервалів.

Кожен розділ текстового документа рекомендується починати з нового листа.

Приклад оформлення текстового документа наведено в додатку Д.

В документі великого обсягу (пояснювальній записці) на першому листі поміщають зміст, що включає в себе номери та назви розділів і підрозділів з зазначенням номерів аркушів. Слово «ЗМІСТ» записують у вигляді заголовка (симетрично тексту) прописними літерами (додаток Г). При цьому порядковий номер змісту не приписують і крапку в кінці не ставлять. Найменування, включені у зміст, записують малими буквами, починаючи з прописної літери.

Текст документа повинен бути коротким, чітким, і не допускати різних тлумачень.

При викладенні обов'язкових вимог в тексті повинні вживатися слова «повинен», «слід», «необхідно» і похідні від них.

В документах повинні застосовуватися науково-технічні терміни, позначення та визначення, встановлені відповідними стандартами, а при їх відсутності загальноприйняті в науково-технічній літературі. При цьому в тексті документа повинні бути приведені відповідні посилання.

Якщо в документі прийнята специфічна термінологія, то в кінці його (перед списком літератури) повинен бути наведений перелік прийнятих термінів з відповідними роз'ясненнями. При цьому в тексті документа повинні бути приведені відповідні посилання.

В тексті документа не допускаються: застосовувати для одного і того ж поняття різні науково-технічні терміни, близькі за змістом (синоніми), а також іноземні слова і терміни за наявності рівнозначних слів і термінів в українській мові; скорочувати позначення одиниць фізичних величин, якщо вони вживаються без цифр за винятком одиниць фізичних величин у заголовках таблиць і в розшифруваннях літерних позначень, що входять до формули; застосовувати скорочення слів, крім встановлених правилами української орфографії, а також відповідними державними стандартами.

1.5.2 Нумерація формул і оформлення посилань на них

В формулах слід застосовувати символи та позначення, встановлені відповідними державними стандартами. Пояснення символів і числових коефіцієнтів, що входять у формулу, якщо вони не пояснені раніше в тексті, повинні бути наведені безпосередньо під формулою. Пояснення кожного символу слід давати з нового рядка в тій послідовності, в якій символи наведені у формулі. Перший рядок пояснення має починатися зі слова «де» без двокрапки після нього. Всі формули, якщо їх в документі більше однієї, нумерують арабськими цифрами в межах розділу. Номер формули складається з номера розділу і порядкового номера формули, розділених крапкою. Номер вказують з правого боку аркуша на рівні формули в круглих дужках.

Приклад – Щільність кожного зразка ρ , кг/м³ обчислюється за формулою

$$\rho = \frac{m}{V}, \quad (4.1)$$

де m – маса зразка кг; V – об'єм зразка, м³.

Формули, що йдуть одна за одною і не розділені текстом, розділяють комою.

Посилання в тексті на номер формули дають у дужках, наприклад: «... у формулі (4.1) ...». Допускається нумерація формул в межах всього документа.

1.5.3 Оформлення ілюстрацій (рисуноків)

Кількість ілюстрацій має бути достатньою для пояснення тексту, що викладається. Ілюстрації повинні розташовуватися як по тексту документа (як можна ближче до відповідних розділів тексту), так і в кінці його або подані в додатку. Ілюстрації повинні бути виконані у відповідності з вимогами стандартів. Ілюстрації, за винятком ілюстрацій, поміщених в додатки, слід нумерувати арабськими цифрами наскрізною нумерацією. Якщо рисунок один, то він позначається «Рисунок 1».

Ілюстрації кожного додатка позначають окремою нумерацією арабськими цифрами з додаванням перед цифрою позначення додатка. Наприклад, – Рисунок А.3.

Допускається нумерувати ілюстрації в межах розділу. В цьому випадку номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Наприклад – Рисунок 1.1.

При посиланнях на ілюстрації при наскрізній нумерації слід писати « у відповідності з рисунком 2 ... » і при нумерації в межах розділу слід писати – «.. у відповідності з рисунком 1.2 ... ».

Ілюстрації, при необхідності, можуть мати найменування та пояснювальні дані (підрисуночний текст). Слово «Рисунок» та назву розміщують після пояснювальних даних і поміщають наступним чином: Рисунок 1 – Деталі приладу.

1.5.4 Оформлення таблиць

Цифровий матеріал, як правило, оформляють у вигляді таблиць, які застосовують для кращої наочності та зручності порівняння показників. Таблиця може мати заголовок, який слід виконувати малими літерами (крім першої прописної) і розміщувати над таблицею. Заголовок повинен бути коротким і повністю відображати зміст таблиці. Заголовки граф таблиці починають з прописних літер.

Всі таблиці, якщо їх більше однієї, нумерують в межах всього документа арабськими цифрами. Допускається нумерація таблиць у межах одного розділу. У цьому разі номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою.

Над лівим верхнім кутом таблиці розміщують напис «Таблиця» з зазначенням номера таблиці, наприклад «Таблиця 2.1».

Якщо таблиця має назву (заголовок), то вона наводиться над таблицею і відокремлюється від напису «Таблиця» дефісом, наприклад: Таблиця 2.3 – Дані для розрахунку надійності.

Таблицю, в залежності від її розміру, поміщають під текстом, в якому вперше дано посилання на неї, або на наступній сторінці, а, при необхідності, у додатку до документа. Допускається розміщувати таблицю уздовж довгої сторони документа.

Слово «Таблиця» вказують один раз ліворуч над першою частиною таблиці.

Якщо у документі одна таблиця, то вона повинна бути позначена, наприклад: «Таблиця 1» або «Таблиця В.1», якщо вона наведена у додатку В.

На всі таблиці документа повинні бути наведені посилання в тексті документа. При посиланні слід писати слово «Таблиця» із вказівкою номера.

1.5.5 Оформлення додатків до ПЗ

Ілюстраційний матеріал, таблиці або текст допоміжного характеру допускається давати у вигляді додатків. Додаток оформляється як продовження даного документа на наступних його листах або випускають у вигляді самостійного документа.

Додатки поміщають в кінці пояснювальної записки після «ВИСНОВКУ».

Додатки позначають великими літерами українського алфавіту, починаючи з А, за винятком літер І, О, Ї, Й. Щ. Після слова «Додаток» слідує літера, що позначає його послідовність.

Допускається позначення додатків літерами латинського алфавіту, за винятком букв І і О.

В разі повного використання літер українського і латинського алфавітів допускається позначати додатки арабськими цифрами.

Якщо в документі один додаток, то він позначається «Додаток А».

Додаток повинен мати заголовок, який записується симетрично тексту з прописної літери окремим рядком.

Нумерація аркушів документа і додатків, що входять до складу документа, повинна бути наскрізна.

Ілюстрації і таблиці в додатках нумерують в межах кожного додатка.

Якщо в документі є додатки, то на них роблять посилання в основному тексті документа, а у змісті перераховують всі додатки із зазначенням номерів і заголовків.

1.6 Вимоги до змісту та оформлення графічної документації

1.6.1 Зміст графічної документації

Зміст і обсяг графічної документації до курсового проєкту залежать від характеру та тематики виконуваної роботи.

Графічна документація, в загальному випадку, включає: креслення, електричні схеми та ілюстраційний матеріал. Креслення повинні відображати

конструкцію проєктованого пристрою. В якості креслень можуть виконуватися складальне і габаритне креслення, а також креслення друкованої плати. В якості електричних схем можуть виконуватися структурна, функціональна і принципова схеми. Структурну та функціональну схеми допускається зображувати в ПЗ у вигляді рисунків. В якості ілюстраційного матеріалу можуть приводитися схеми алгоритмів, часові діаграми роботи пристрою, графіки, осцилограми і т. ін.

Якщо студент представляє на захист виготовлений ним або за його участю макет проєктованого пристрою, за узгодженням з керівником обсяг документації проєкту може бути скорочений.

1.6.2 Вимоги до оформлення графічної документації

Всі схеми і креслення курсового проєкту виконують відповідно до вимог державних стандартів і ЄСКД. Схеми і креслення виконують на ПК з використанням відповідних засобів розробки з наступною роздруківкою на принтері. Підписи в основному написі виконують лише чорними чорнилами.

Якщо можливо, креслення виконують в масштабі 1:1 на білому папері формату А3 або А2. Якщо на цьому форматі паперу розташовується кілька креслень або схем меншого формату, то розрізати папір на окремі формати не слід.

Крім основних форматів (А4, А3, А2, А1 і А0), допускається застосування додаткових форматів, які утворюються збільшенням сторін основних форматів на величину, кратну розмірам формату А4. При цьому коефіцієнт кратності повинен бути цілим числом. Вид формату вибирається автором роботи, який повинен прагнути вибрати мінімальні розміри листа при збереженні наочності і зручності при читанні документів.

Креслення та схеми слід забезпечувати необхідними поясненнями, що дають можливість читати їх без звернення до тексту ПЗ.

В залежності від основного призначення схеми підрозділяють на наступні типи: електричні структурні, електричні функціональні, електричні принципові, з'єднань (монтажні), підключення, розташування, загальні.

1.7 Порядок комплектування документів

Перед комплектацією всі текстові та графічні документи повинні підписати виконавець, суміжні виконавці та керівник проєкту.

Підпис керівника про допуск до захисту ставиться після остаточного оформлення альбому, що містить документацію по курсовому проєкту.

Документація проєкту в загальному випадку комплектується в такій послідовності:

- титульний аркуш;
- завдання на курсовий проєкт;
- пояснювальна записка (оформити у форматі PDF);
- структурна схема (якщо вона виконується як графічний документ), оформити у Visio 2010;
- функціональна схема (якщо вона виконується як графічний документ), оформити у Visio 2010;
- принципова схема (може бути в реальних роботах), оформити у Visio 2010;
- перелік елементів до принципової схеми (якщо він не зображений на схемі);
- ілюстративний матеріал (якщо він виконується як графічний документ).

Всі документи, які будуть друкуватись, приводять до формату А4 згідно з ГОСТ 2.501-88 «Правила обліку та зберігання» та брошурують в одну папку або альбом.

Якщо текстові документи представляються в електронному виді, то їх треба об'єднати у одну папку та в упакованому виді переслати викладачу для перевірки.

2 ПРОЄКТУВАННЯ ПРИСТРОЮ ОБМІНУ ІНФОРМАЦІЄЮ МІЖ МОДУЛЕМ УСАПП ТА МІКРОКОНТРОЛЕРОМ

2.1 Завдання

Виконати проєктування пристрою обміну інформацією між модулем УСАПП мікроконтролера AT mega 8515 та віртуальним терміналом:

- а) розробити, навести та описати робочу модель в пакеті PROTEUS;
- б) зробити розрахунок модуля УСАПП;
- в) розробити схему алгоритму роботи пристрою;
- г) розробити керуючу програму мовою Асемблер;
- д) виконати моделювання пристрою в пакеті PROTEUS:
 - навести «скріншоти», які пояснюють роботу моделі та підтверджують правильність виконаних розрахунків, схеми алгоритму та робочої керуючої програми;
 - обґрунтувати, що отримана швидкість обміну відповідає завданню;
 - навести ASCII-код символу, який отримано від віртуального терміналу – Virtual Terminal;
- е) розробити та описати структурну схему пристрою;
- ж) розробити пояснювальну записку та інші текстові документи до курсового проєкту у Word згідно переліку, який наведено у підрозділі 1.3 методички до виконання КП.

Вихідні дані:

- швидкість передачі: $V_{\text{пд}} = 9600$ бод;
- тактова частота кварцового резонатора: $f_{\text{CLK}} = 8\text{МГц}$;
- для обміну використовувати модуль USART;
- режим передачі – асинхронний;
- розмір слова даних – 8 розрядів;
- використовувати перевірку на непарність;

- кількість стоп-бітів – 1;
- байт для передачі знаходиться у регістрі R22 та дорівнює: 10110101b;
- символ для прийому від віртуального термінулу – 3.

2.2 Розробка та опис робочої моделі в пакеті PROTEUS

2.2.1 Робоча модель пристрою

Робочу модель дослідження універсального синхронно/асинхронного приймача-передавача (УСАПП) в пакеті Proteus наведено на рисунку 2.1.

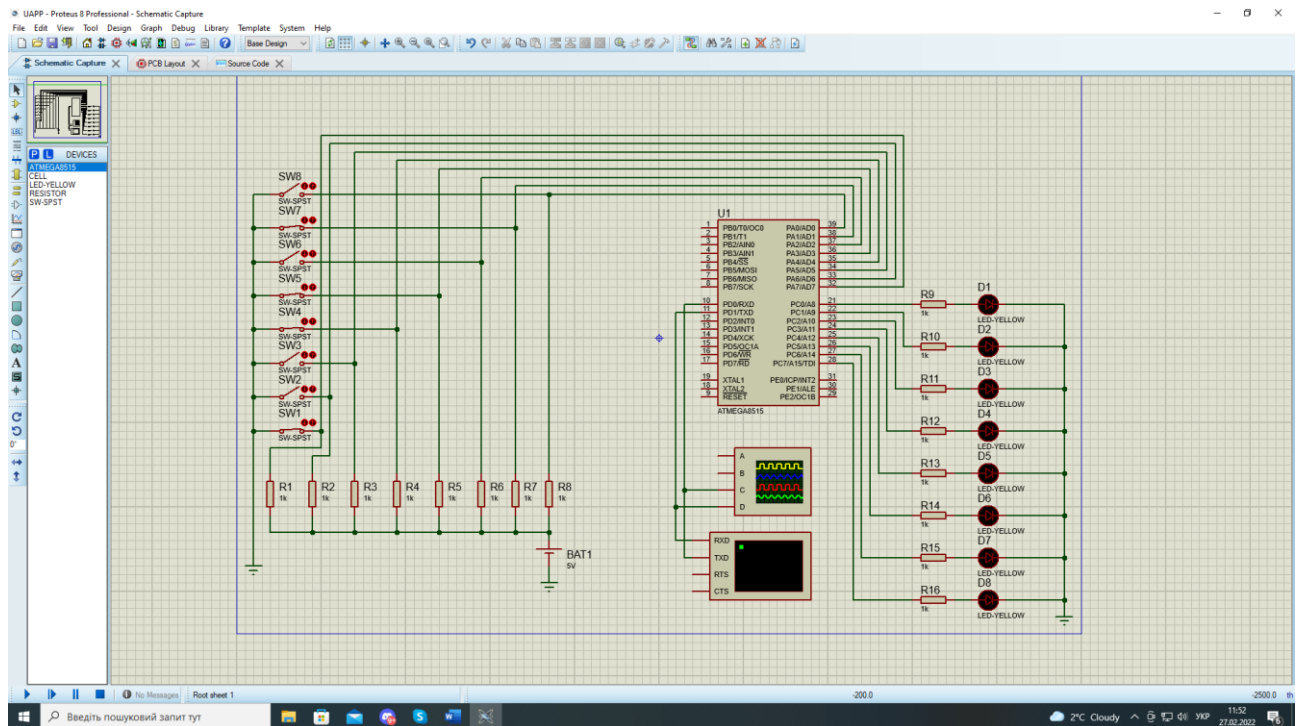


Рисунок 2.1 – Схема робочої моделі

2.2.2 Опис робочої моделі

Зліва знаходяться вісім кнопок, за допомогою яких задається байт (8 біт) даних, який передається через модуль УС АПП.

Кнопки підключено до восьми ліній порту А мікроконтролера: PA.0, PA.1, ... , PA.7. Якщо якась із кнопок відпущена, то через резистори R1...R8 на відповідну лінію порту А подається логічна 1. Якщо кнопка нажата, то вводиться логічний нуль. Праворуч зображено вісім світлодіодів, які підключено до ліній порту C: PC.0, PC.1, ... , PC.7. Катоди світлодіодів підключено до спільного дроту (землі), а на аноди через резистори R9...R16, які обмежують струм, із виходів порту C подаються логічні одиниці, коли треба засвітити відповідний світлодіод.

До ліній TxD – вихід передавача УСАПП та RxD – вхід приймача підключено осцилограф, та віртуальний термінал «Virtual Terminal», який вибрано з віртуальних інструментів програми Proteus. На них відображаються повідомлення, які вводяться/виводяться через модуль УСАПП в/з мікроконтролера.

До виводів XTAL1 та XTAL2 в реальних схемах підключають кварцовий резонатор частотою, яка визначає тактову частоту генератора мікроконтролера: f_{BQ} . В нашій роботі вона дорівнює 8МГц. Також підключають конденсатори C1 та C2, ємністю 30пФ, які призначені для підвищення стабільності роботи системного генератора. Резонатор та конденсатори потрібні в практичній схемі. В модель Proteus їх можна не вводити. Для задання частоти треба двічі лівою кнопкою миші клацнути на мікроконтролері та в опції Clock Frequency вказати значення частоти.

В якості мікроконтролера в моделі використано AVR-мікроконтролер ATmega 8515.

2.3 Розрахунок модуля УСАПП

2.3.1 Розрахунок швидкості обміну

При роботі модуля в *асинхронному режимі* швидкість обміну визначається вмістом регістра UBRR та станом розряду U2X (U2Xn) регістра UCSRA [2]. Якщо цей розряд встановлено в «1», коефіцієнт ділення попереднього дільника зменшується у два рази, а швидкість обміну відповідно подвоюється. При роботі у синхронному режимі цей розряд має бути скинуто.

Швидкість [бод]	8.0000 МГц				11.0592 МГц				14.7456 МГц				16.0000 МГц				18.4320 МГц				20.0000 МГц			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]	UBRR	Похибка [%]
2400	207	0.2	416	-0.1	287	0.0	575	0.0	383	0.0	767	0.0	416	-0.1	832	0.0	479	0.0	959	0.0	520	0.0	1041	0.0
4800	103	0.2	207	0.2	143	0.0	287	0.0	191	0.0	383	0.0	207	0.2	416	-0.1	239	0.0	479	0.0	259	0.2	520	0.0
9600	51	0.2	103	0.2	71	0.0	143	0.0	95	0.0	191	0.0	103	0.2	207	0.2	119	0.0	239	0.0	129	0.2	259	0.2
14400	34	-0.8	68	0.6	47	0.0	95	0.0	63	0.0	127	0.0	68	0.6	138	-0.1	79	0.0	159	0.0	86	-0.2	173	-0.2
19200	25	0.2	51	0.2	35	0.0	71	0.0	47	0.0	95	0.0	51	0.2	103	0.2	59	0.0	119	0.0	64	0.2	129	0.2
28800	16	2.1	34	-0.8	23	0.0	47	0.0	31	0.0	63	0.0	34	-0.8	68	0.6	39	0.0	79	0.0	42	0.9	86	-0.2
38400	12	0.2	25	0.2	17	0.0	35	0.0	23	0.0	47	0.0	25	0.2	51	0.2	29	0.0	59	0.0	32	-1.4	64	0.2
57600	8	-3.5	16	2.1	11	0.0	23	0.0	15	0.0	31	0.0	16	2.1	34	0.8	19	0.0	39	0.0	21	-1.4	42	0.9
76800	6	-7.0	12	0.2	8	0.0	17	0.0	11	0.0	23	0.0	12	0.2	25	0.2	14	0.0	29	0.0	15	1.7	32	-1.4
115200	3	8.5	8	-3.5	5	0.0	11	0.0	7	0.0	15	0.0	8	-3.5	16	2.1	9	0.0	19	0.0	10	-1.4	21	-1.4
230400	1	8.5	3	8.5	2	0.0	5	0.0	3	0.0	7	0.0	3	8.5	8	-3.5	4	0.0	9	0.0	4	8.5	10	-1.4
250000	1	0.0	3	0.0	2	-7.8	5	-7.8	3	-7.8	6	5.3	3	0.0	7	0.0	4	-7.8	8	2.4	4	0.0	9	0.0
0.5M	0	0.0	1	0.0	—	—	2	-7.8	1	-7.8	3	-7.8	1	0.0	3	0.0	—	—	4	-7.8	—	—	4	0.0
1.0M	—	—	0	0.0	—	—	—	—	0	-7.8	1	-7.8	0	0.0	1	0.0	—	—	—	—	—	—	—	—
Max ¹⁾	0.5M		1.0M		691.2K		1.3824M		921.6K		1.8432M		1.0M		2.0M		1.152M		2.304M		1.25M		2.5M	
¹⁾ При UBRR = 0 і похибці 0.0%																								

Значення регістра UBRR дозволяє отримати стандартні для асинхронного режиму швидкості передачі при використанні різних резонаторів, а також величини значень похибок відносно стандартних швидкостей, що отримуються у цьому випадку (стандартні для асинхронного режиму швидкості передачі при використанні різних резонаторів позначено жовтим кольором).

Згідно з таблицею 2.1 умовам завдання задовольняють значення UBRR = 51 при U2X = 0, або UBRR = 103 при U2X = 1. В обох випадках похибка дорівнює 0,2%. **Обираємо** UBRR = 51 = \$33 та U2X = 0.

2.3.2 Програмування регістра контролера швидкості передачі UBRR

Для нашого МК-ра регістр UBRR є 12-розрядним та фізично розміщений у двох регістрах введення/виведення (таблиця 2.2).

Таблиця 2.2 – Розміщення регістрів контролера швидкості передачі

Модель	Регістри	Адреса
ATmega8515x/8535x ¹⁾	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
ATmega8x/16x/32x ¹⁾	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
ATmega64x/128x	UBRR0H:UBRR0L	(\$90):\$09 (\$29)
	UBRR1H:UBRR1L	(\$98):(\$99)
ATmega48x/88x/168x	UBRR0H:UBRR0L	(\$C5):(\$C4)
ATmega162x ¹⁾	UBRR0H:UBRR0L	\$20 (\$40):\$09 (\$29)
	UBRR1H:UBRR1L	\$3C (\$5C):\$00 (\$20)
ATmega164x/324x/644x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
ATmega165x/325x/3250x/645x/6450x	UBRR0H:UBRR0L	(\$C5):(\$C4)
¹⁾ У цих моделях регістр UBRRH (UBRRnH) розміщується за тією ж адресою, що й регістр UCSRC (UCSRnC)		
ATmega640x/1280x/2560x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
	UBRR2H:UBRR2L	(\$D5):(\$D4)
	UBRR3H:UBRR3L	(\$135):(\$134)
ATmega1281x/2561x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)

Ці **регістри мають адреси**: $UBRRH = \$0040$; $UBRRL = \$0029$. **Біт URSEL** регістра UCSR0C після скидання дорівнює нулю.

Згідно з розрахунком, наведеним вище, $UBRR = 51 = \$0033$.

Тоді робоча програма має вигляд:

LDI R18, \$00; R18←\$00;

LDI R31, \$00; R31←\$00;

LDI R30, \$40; R30←\$40; R31, R30 (Z)←\$0040 (адреса UBRRH);

ST Z, R18; UBRRH ←00 (старший байт UBRR).

LDI R19, \$33; R19←\$33;

LDI R27, \$00; R27←\$00;

LDI R26, \$29; R26←\$29; R27, R26 (X)←\$0029 (адреса UBRRL);

ST X, R19; UBRRL = \$33 (молодший байт UBRR).

2.3.3 Програмування регістра UCSRA

Формат регістра UCSRA наведено на рисунку 2.2, а опис його окремих розрядів – у таблиці 2.3.

	7	6	5	4	3	2	1	0
	RXC _n	TXC _n	UDRE _n	FE _n	DOR _n	UPEn	U2X _n	MPCM _n
Зчитування(R)/Запис(W)	R	R/W	R	R	R	R	R/W	R/W
Початкове значення	0	0	1	0	0	0	0	0

Рисунок 2.2 – Формат регістрів UCSRA (UCSRnA)

Таблиця 2.3 – Опис розрядів регістрів UCSRA, UCSR0A та UCSR1A

Розряд	Назва	Опис
7	RXC (RXCn)	Прапорець завершення прийому Прапорець встановлюється в «1» при пересиланні прийнятого слова з регістра зсуву приймача в регістр даних UDR і утримується в одиниці при наявності непрочитаних даних у буфері приймача (регістр даних UDR). Скидається прапорець апаратно після очищення буфера (у UART – після читання регістра даних). Якщо розряд RXCIE (RXCIE _n) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується запит на переривання «прийом завершено».
6	TXC (TXCn)	Прапорець завершення передачі Прапорець встановлюється в «1» після передачі усіх розрядів пакету із регістра зсуву передавача, за умови, що у регістр даних UDR не було завантажено нового значення. Якщо розряд TXCIE (TXCIE _n) регістра UCSRB (UCSRnB) встановлено, то при встановленні прапорця генерується переривання «передачу завершено». Прапорець скидається апаратно при виконанні підпрограми обробки переривання чи програмно, записуючи у нього логічну 1.
5	UDRE (UDRE _n)	Прапорець очищення регістра даних Даний прапорець встановлюється в «1» при пустому буфері передавача (після пересилання байта із регістра даних UDR у регістр зсуву передавача). Встановлений прапорець означає, що у регістр даних можна завантажити нове значення. Якщо розряд UDRIE регістра UCR (UCSRB) встановлено, генерується запит на переривання «регістр даних пустий». Прапорець скидається апаратно, при записі у регістр даних.

4	FE (FEn)	Прапорець помилки кадровання Прапорець встановлюється в «1» при виявленні помилки кадровання, тобто якщо перший стоп-біт прийнятої послідовності дорівнює «0». Прапорець скидається при прийомі стоп-біта, який дорівнює «1»
3	OR (ORn)	Прапорець переповнення В USART прапорець встановлюється в «1», якщо у момент виявлення нового старт-біта в регістрі зсуву приймача знаходиться останнє прийняте слово, а буфер приймача заповнений (два значення). Прапорець скидається при пересиланні прийнятих даних із регістра зсуву приймача у буфер.
	DOR (DORn)	
2	UPE (UPEn)	Прапорець помилки контролю біта парності Прапорець встановлюється в «1», якщо в даних, що знаходяться в буфері приймача, виявлена помилка контролю парності. При вимкненому контролі парності цей розряд постійно скинуто в «0»
1	U2X (U2Xn)	Подвоєння швидкості обміну Якщо цей розряд встановлено в «1», коефіцієнт ділення попереднього дільника контролера швидкості передачі зменшується з 16 до 8, подвоюючи тим самим швидкість асинхронного обміну за послідовним каналом. В USART розряд U2X (U2Xn) використовується тільки при асинхронному режимі роботи. У синхронному режимі він має бути скинутий
0	MPCM (MPCMn)	Режим мультипроцесорного обміну Розряд MPCM використовується у режимі мультипроцесорного обміну. Якщо він встановлений в «1», ведений мікроконтролер очікує прийому кадру, що містить

	(\$C8)						•		•	•	
UCSR1B	\$01(\$21)					•					Регістр В керування USART1
	(\$9A)			•							
	(\$C9)						•		•	•	
UCSR1C	\$3C (\$5C)					•					Регістр С керування USART1
	(\$9D)			•							
	(\$CA)						•		•	•	
UCSR2A	(\$D0)								•		Регістр А керування USART2
UCSR2B	(\$D1)								•		Регістр В керування USART2
UCSR2C	(\$D2)								•		Регістр С керування USART2
UCSR3A	(\$130)								•		Регістр А керування USART3
UCSR3B	(\$131)								•		Регістр В керування USART3
UCSR3C	(\$132)								•		Регістр С керування USART3

Тоді програма має вигляд:

LDI R16, \$20; R16←KC1 = \$20,

LDI R27, \$00; R27←\$00,

LDI R26, \$2B; R26←\$2B; R27, R26 (X) ←\$002B,

ST X, R16; UCSR0A ←R16 = KC1 = \$20.

2.3.4 Програмування регістра UCSRC

Формат регістра UCSRC наведено на рисунку 2.3, а опис його окремих розрядів – у таблиці 2.5.

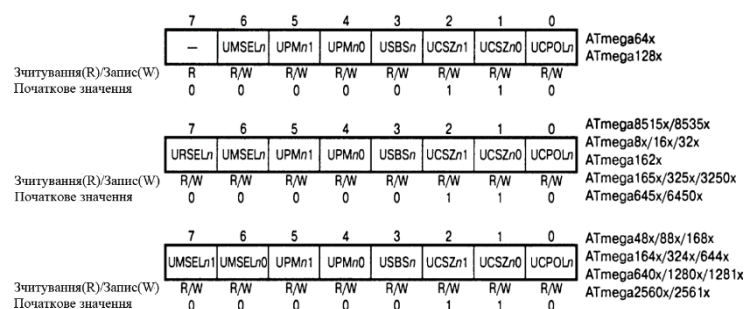


Рисунок 2.3 – Формат регістрів UCSRC (UCSRnC)

Таблиця 2.5 – Опис розрядів регістрів UCSRC (UCSRnC)

Розряд	Назва	Опис
7	UMSELn1	Режим роботи USART (моделі 48х/88х/168х, 164х/324х/644х і 640х/1280х/1281х/2560х/2561х) Разом з розрядом UMSELn0 визначає режим роботи модуля USART
	URSEL* (URSELn)	Вибір регістра Цей розряд визначає, у який з регістрів модуля здійснюється запис. Якщо розряд встановлено в «1», звернення здійснюється до регістра UCSRC (UCSRnC). Якщо ж розряд скинуто в «0», звернення здійснюється до регістра UBRRH (UBRRnH)
6	UMSEL (UMSELn)	Режим роботи USART Якщо розряд скинуто в «0», модуль USART працює в асинхронному режимі. Якщо розряд встановлено в «1», то модуль USART працює у синхронному режимі
	UMSELn0	Режим роботи USART (моделі 48х/88х/168х, 164х/324х/644х і 640х/1280х/1281х/2560х/2561х) Разом з розрядом UMSELn1 визначає режим роботи модуля USART
5	UPM1 (UPMn1)	Режим роботи схеми контролю та формування парності 5-й та 4-й розряди визначають функціонування схем контролю та формування парності (таблиця 2.6)
4	UPM0 (UPMn0)	
3	USBS (USBSn)	Кількість стоп-бітів Цей розряд визначає кількість стоп-бітів, що пересилаються передавачем. Якщо розряд скинуто в «0», передавач посилає 1 стоп-біт, якщо встановлено в «1», то 2 стоп-біти. Для приймача вміст цього розряду не має значення
2	UCSZ1 (UCSZn1)	Формат кадру Разом з розрядом UCSZ2 (UCSZn2) ці розряди визначають кількість розрядів даних та посилок (розмір слова) (таблиця 2.7)
1	UCSZ0 (UCSZn0)	

0	UCPOL (UCPOL _n)	Полярність тактового сигналу Значення цього розряду визначає момент видачі та зчитування даних на виводах модуля. Розряд використовується тільки при роботі у синхронному режимі. При роботі в асинхронному режимі він має бути скинутий у «0»		
		UCPOL (UCPOL _n)	Видача даних на вивід TXD (TXD _n)	Зчитування даних з виводу RXD (RXD _n)
		0	Наростаючий фронт ХСК (ХСК _n)	Спадаючий фронт ХСК (ХСК _n)
		1	Спадаючий фронт ХСК (ХСК _n)	Наростаючий фронт ХСК (ХСК _n)
* – зарезервований у моделях АТmega 64х/128х				

Примітка: n = 0, 1, 2 чи 3.

Для МК-ра AT mega 8515 згідно з таблицею 2.5 розряд UMSEL програмує режим роботи модуля USART. Оскільки у завданні задано асинхронний режим, то цей біт повинен мати нульове значення.

Біти UPM1 та UPM0 згідно з таблицею 2.6 керують контролем парності.

Таблиця 2.6 – Керування контролем парності

UPM1 (UPM _{n1})	UPM0 (UPM _{n0})	Режим роботи
0	0	Виключений
0	1	Зарезервовано
1	0	Включений, перевірка на парність (even parity)
1	1	Включений, перевірка на непарність (odd parity)

Примітка: n = 0, 1, 2 чи 3.

В нашому випадку необхідно запрограмувати: UPM1 = 1; UPM0 = 1 (перевірка на непарність).

Значення біта парності отримується шляхом виконання операції «виключне АБО» над усіма розрядами слова даних, яке передається. Якщо використовується перевірка на парність (even parity), то отриманий результат інвертується:

$$\begin{aligned} P_{EVEN} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1, \\ P_{ODD} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0. \end{aligned} \quad (2.1)$$

Біт **USBS** скидається в **нуль**, тому що ми повинні передавати **1 стоп-біт**.

Біти **UCSZ1, UCSZ0** разом з бітом **UCSZ2** регістра **UCSRB** програмують розмір слова даних (таблиця 2.7).

Таблиця 2.7 – Визначення розміру слова даних у модулях USART

UCSZ2 (UCSZn2)	UCSZ1 (UCSZn1)	UCSZ0 (UCSZn0)	Розмір слова даних
0	0	0	5 розрядів
0	0	1	6 розрядів
0	1	0	7 розрядів
0	1	1	8 розрядів
1	0	0	Зарезервовано
1	0	1	Зарезервовано
1	1	0	Зарезервовано
1	1	1	9 розрядів

Примітка: n = 0, 1, 2 чи 3.

Для нашого прикладу треба запрограмувати: **UCSZ1=1; UCSZ0=1, а UCSZ2=0** (передаються 8 розрядів).

Розряд **UCPOL** в асинхронному режимі не використовуються, тому записуємо в нього **нуль**.

Біт **URSEL=1**, тому, що запис виконується в регістр **UCSRC**.

Тоді керуюче слово **KC2** має вигляд:

	7p	6p	5p	4p	3p	2p	1p	0p	
UCSRC	1	0	1	1	0	1	1	0	KC2=\$B6

Згідно з таблицею 2.4 регістр UCSRC для заданого МК-ра має адресу: \$0040.

Тоді програма має вигляд:

LDI R17, \$B6; R17← \$B6,

LDI R29, \$00; R29←\$00,

LDI R28, \$40; R28←\$C2, R29, R28 (Y) ←\$0040,

ST Y , R17; UCSR0C ←R17 = KC2 = \$B6.

2.3.4 Програмування регістра UCSRB

Формат регістра UCSRB наведено на рисунку 2.4, а опис його розрядів – у таблиці 2.8.

	7	6	5	4	3	2	1	0
	RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB8 _n	TXB8 _n
Зчитування(R)/Запис(W)	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Початкове значення	0	0	0	0	0	0	1	0

Рисунок 2.4 – Формат регістрів UCSRB (UCSRnB)

Таблиця 2.8 – Опис розрядів регістрів UCSRB (UCSRnB)

Розряд	Назва	Опис
7	RXCIE (RXCIE _n)	Дозвіл переривання після завершення прийому Якщо даний розряд в «1», то при встановленні прапорця RXC (RXC _n) регістра UCSRA (UCSRnA) генерується переривання «прийом завершено», якщо прапорець I регістра SREG встановлено в «1»
6	TXCIE (TXCIE _n)	Дозвіл переривання після завершення передачі Якщо даний розряд в «1», то при встановленні прапорця TXC (TXC _n) регістра UCSRA (UCSRnA) генерується

		переривання «передачу завершено, якщо прапорець I регістра SREG встановлено в «1»
5	UDRIE (UDRIEn)	Дозвіл переривання при очищенні регістра даних UART Якщо даний розряд встановлено в «1», то при встановленні прапорця UDRE у регістрі UCSRA (UCSRnA) генерується переривання «регістр даних пустий», якщо прапорець I регістра SREG встановлено в «1»
4	RXEN (RXENn)	Дозвіл прийому Якщо даний розряд встановлюється в одиницю, то дозволяється робота приймача та перевизначається функціонування виводу RXD (RXDn). При скидані розряду RXEN (RXENn) робота приймача забороняється, а його буфер очищується. Значення прапорців RXC (RXCn), DOR/OR, FE (FEN) при цьому стають недійсними
3	TXEN (TXENn)	Дозвіл передачі При встановленні цього розряду в «1» дозволяється робота передавача UART/USART та перевизначається функціонування виводу TXD (TXDn). Якщо розряд скидається в «0» під час передачі, відключення передавача відбудеться тільки після закінчення передачі даних, що знаходяться у регістрі зсуву та буфері передавача
2	CHR9 (CHR9n) UCSZ2 (UCSZn2)	Формат пакетів Цей розряд використовується для задання розміру слів даних, що передаються за послідовним каналом. В модулях USART він використовується разом з розрядами UCSZ1:0 (UCSZn1:0) регістра UCSRC (UCSRnC). У модулях UART, якщо розряд CHR9 (CHR9n) встановлено в «1», здійснюється передача та прийом 9-розрядних даних, якщо скинутий – 8-розрядних
1	RXB8 (RXB8n)	8-й розряд даних, що приймаються При використанні 9-розрядних слів даних цей розряд містить значення старшого розряду прийнятого слова. У випадку USART вміст цього розряду має бути зчитано до читання регістра даних UDR
0	TXB8 (TXB8n)	8-й розряд даних, що передаються При використанні 9-розрядних слів даних, вміст цього розряду є старшим розрядом слова, що передається.

		Необхідне значення має бути занесене у цей розряд до завантаження байта даних у регістр UDR
--	--	---

Примітка: $n = 0, 1, 2$ або 3 .

Для нашого прикладу необхідно запрограмувати:

$RXCIE = TXCIE = UDRIE = 0$; $UCSZ02 = 0$ (таблиця 2.7); та $TXEN = RXEN = 1$.

Інші біти не використовуються, тому пишемо в них нулі.

Тоді керуюче слово KC3 має вигляд:

	7p	6p	5p	4p	3p	2p	1p	0p	
UCSRB	0	0	0	1	1	0	0	0	KC3=\$18

Згідно з таблицею 2.4 регістр UCSRБ має адресу: \$002A.

Тоді програма має вигляд:

LDI R19, \$18; R19 ← KC3=\$18,

LDI R29, \$00; R29 ← \$00,

LDI R28, \$2A; R28 ← \$2A; R29, R28 (Y) ← \$002A,

ST Y, R19; UCSRБ ← R19 = KC3 = \$18.

2.3.5 Програмування регістра даних UDR

Згідно з таблицею 2.9 регістр даних UDR нашою МК-ра має адресу \$000C.

Тоді програма має вигляд:

LDI R31, \$00; R31 ← \$00;

LDI R30, \$0C; R30 ← \$0C; R31, R30 (Z) ← \$000C;

ST Z, R22; UDR ← R22 (завантаження байта для передачі).

Після завантаження байта з регістра R22 в регістр UDR (буфер передавача) інформація у послідовному двійковому коді у старт-стопному форматі передається на лінію TXD (для МК-ра AT mega 8515 – PD1).

Таблиця 2.9 – Розміщення регістрів даних модулів USART/UART деяких мікроконтролерів сімейства AVR

Регістр	Адреса	ATmega8515x/8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48/x88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/350x, ATmega645x/6450x	ATmega640x/1280x/1281x	ATmega2560x/2561x	Опис
UDR	\$0C	•	•								Регістр даних USART
	(\$C6)							•			
UDR0	\$0C			•		•					Регістр даних USART0
	(\$C6)				•		•		•	•	
UDR1	\$03					•					Регістр даних USART1
	(\$9C)			•							
	(\$CE)								•	•	
UDR2	(\$D6)								•		Регістр даних
UDR3	(\$136)								•		Регістр даних

2.4 Схема алгоритму роботи пристрою

Схему алгоритму роботи пристрою наведено на рисунку 2.5.

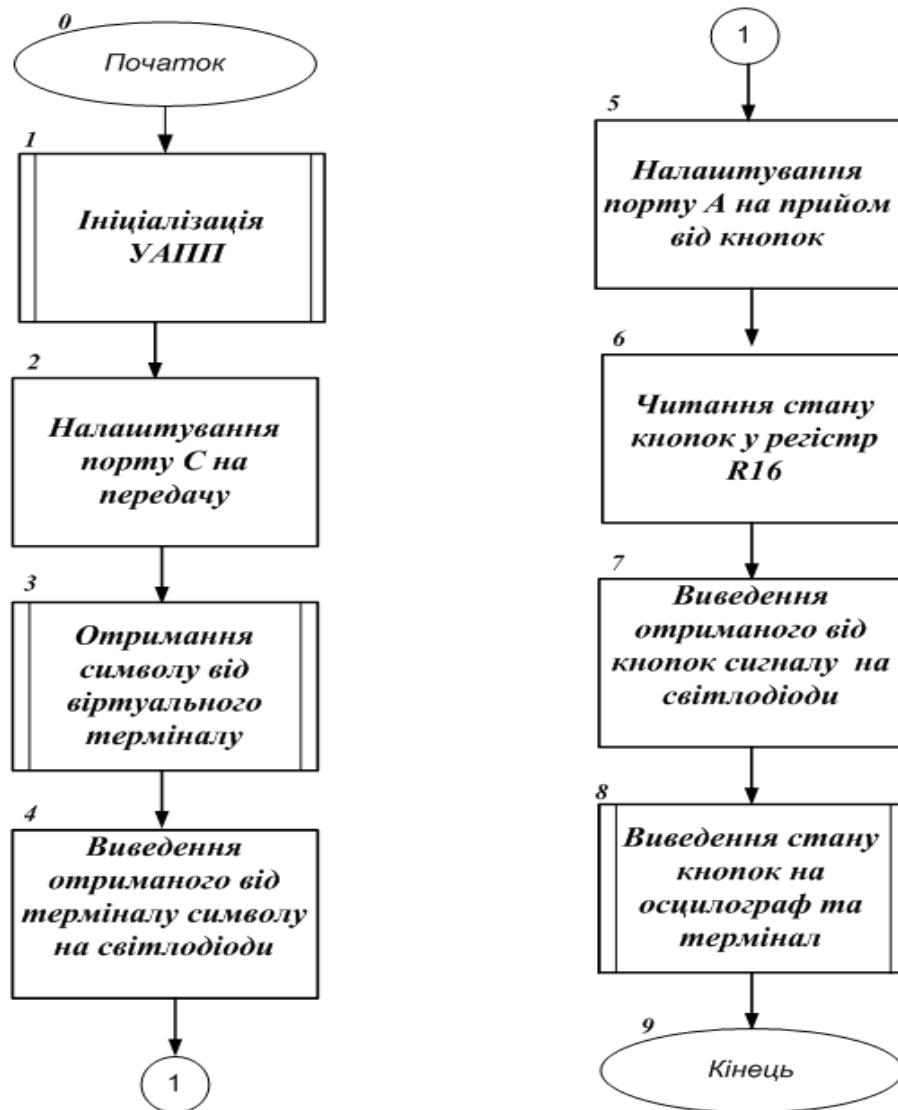


Рисунок 2.5 – Схема алгоритму роботи пристрою

2.5 Керуюча програма мовою Асемблер

2.5.1 Лістинг керуючої програми мовою Асемблер

.nolist ; директива відключає генерацію коду в лістинг,

; тобто далі у файлі *.lss не буде фіксуватися асемблерний код

.include "m8515def.inc" ; підключення стандартного заголовочного

; файлу для ATmega8515

```

.list    ; директива включає генерацію коду в лістинг,
        ; тобто далі у файлі *.lss буде фіксуватися асемблерний код

.equ fCK = 8000000    ; частота в герцах

.equ BAUD = 9600    ; швидкість для УСАПП у бодах

.equ UBRR_value = (fCK/(BAUD*16))-1; розрахунок значення для
        ;регістра UBRR

.cseg    ;директива визначення, що далі іде код програми

.org 0    ;директива визначення, що код програми
        ; записано у FLASH-пам'яті та буде розміщено з нульової адреси

; КОД ОСНОВНОЇ ПРОГРАМИ

Start: ; (блок 0)

; ініціалізації з мітки init_USART

rcall init_USART    ; (блок 1) відносний виклик підпрограми ініціалізації
        ; налаштування порту C на передачу

LDI R16, 0xFF    ; (блок 2) R16 ← 0xFF

OUT DDRC, R16    ; (блок 2) DDRC ← R16

; отримання того, що відправив віртуальний термінал (блок 3)

rcall USART_recieve    ; (блок 3) відносний виклик підпрограми з мітки
        ; USART_recieve

; (блок 4) виведення отриманого від терміналу символу на світлодіоди

OUT PORTC, R16    ; (блок 4) PORTC ← R16

;(блок 5) налаштування порту A на прийом від кнопок

```

```

LDI R16, 0x00          ; (блок 5) R16←0x00

OUT DDRA, R16          ; (блок 5) DDRA← R16

;(блок 6) читання стану кнопок в регістр R16

in R16, PINA           ; (блок 6) R16← PINA

;(блок 7) виведення отриманого від кнопок на світлодіоди

OUT PORTC, R16         ; (блок 7) PORTC←R16

;(блок 8) виведення стану кнопок на осцилограф та термінал

;(підпрограма з міткою USART_send)

rcall USART_send       ; (блок 8) відносний виклик підпрограми з мітки

                        ; USART_send

; (блок 1) ПІДПРОГРАМА ІНІЦІАЛІЗАЦІЇ МОДУЛЯ USART init_USART:

; програмування швидкості обміну 9600бод

ldi R16, high(UBRR_value) ; (блок 1) R16←старший байт UBRR_value

out UBRRH, R16          ; (блок 1) UBRRH←R16

ldi R16, low(UBRR_value) ; (блок 1) R16←молодший байт UBRR_value

out UBRRL, R16          ; (блок 1) UBRRL←R16

; дозвіл роботи приймача-передавача модуля USART

ldi R16, (1<<RXEN)|(1<<TXEN)|(0<<RXCIE)|(0<<TXCIE)|(0<<UDRIE)

out UCSRB, R16          ; (блок 1) UCSRB←R16

; програмування передачі восьми біт, з перевіркою на непарність,

; в асинхронному режимі, з одним стоп-бітом

ldi R16, (1<<URSEL)|(1<<UPM1)|(1<<UPM0)|(1<<UCSZ1)|(1<<UCSZ0)

```

```

out UCSRC, R16          ; (блок 1) UCSRC ← R16

    ret                ; (блок 1) повернення з підпрограми ініціалізації

; передача через модуль USART

USART_send:

out UDR, R16            ; (блок 8) регістр даних UDR ← R16

sending:

; чекання завершення передачі

sbis UCSRA, TXC         ; (блок 8) якщо біт TXC = 1, то наступна ;команда
                        ; пропускається, інакше — наступна команда

rjmp sending           ; (блок 8) безумовний перехід на мітку sending

ret                    ; (блок 8) повернення з підпрограми

USART_recieve:

sbis UCSRA, RXC         ; (блок 3) якщо біт RXC = 1 (в буфері ППМ є
; отриманий байт) , то пропускається наступна команда,
; інакше наступна команда

rjmp USART_recieve     ; (блок 3) безумовний перехід на мітку
                        ; USART_recieve

in R16, UDR            ; (блок 3) R16 ← UDR (завантаження в
                        ; регістр R16 отриманого байта від термінала

ret                    ; (блок 3) повернення з підпрограми

END Start              ; (блок 9)

```

2.5.2 Компіляція робочої програми та отримання hex-файлу

Для програмування мікроконтролерів AVR може використовуватися, наприклад, Microchip Studio 7; Atmel Studio 7 – інтегроване середовище розробки для програмування та відлагодження програм для мікроконтролерів AVR та AVR32 в операційних системах Windows. Після шостої версії середовище може працювати як і з AVR-мікроконтролерами, так і з системами з ARM-архітектурою. У [3] описано послідовність створення робочого проєкту в цьому середовищі, компіляцію програми, створення hex-файлу та його *завантаження в пам'ять мікроконтролера*.

2.6 Моделювання пристрою в пакеті PROTEUS

2.6.1 Налаштування віртуального термінала

Для налаштування віртуального термінала (Terminal) треба двічі натиснути на нього лівою кнопкою миші, та зробити налаштування згідно завдання. Нижче на рисунку 2.6 наведено приклад для випадку, коли зі швидкістю 9600 біт/сек потрібно передавати наступну інформацію: 8 біт даних та один стоп-біт з перевіркою на непарність.

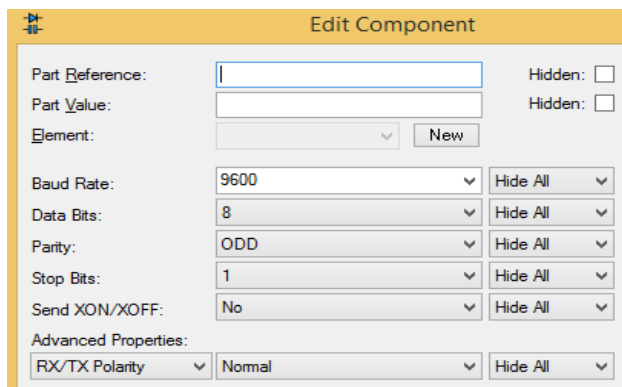


Рисунок 2.6 – Налаштування віртуального термінала

Для налаштування перевірки на парність або непарність треба використати опцію *Parity*.

2.6.2 Налаштування світлодіодів на цифровий режим

Для налаштувати світлодіодів на цифровий режим треба двічі клікнути на них лівою кнопкою миші та обрати необхідний цифровий режим (рисунок 2.7).

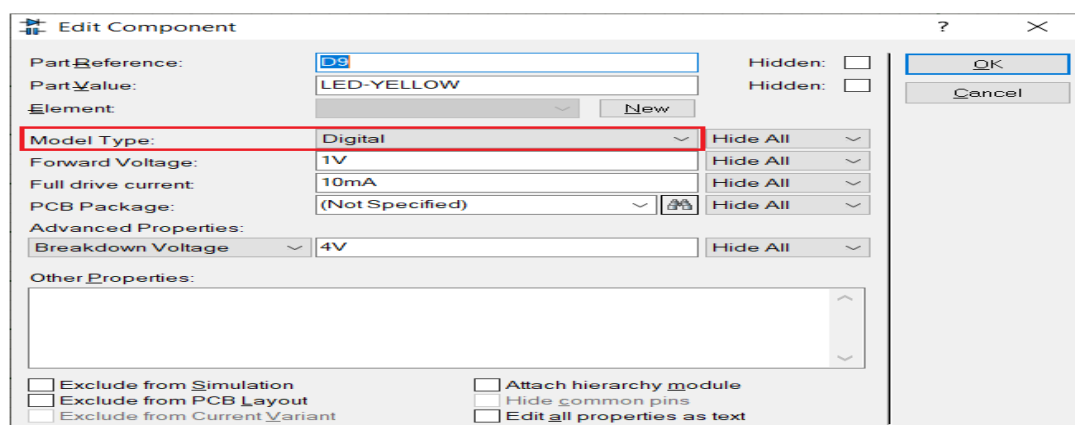


Рисунок 2.7 – Налаштування світлодіодів на цифровий режим

2.6.3 Запуск процесу моделювання

Для запуску процесу моделювання треба натиснути «Start VSM Debugging» (рисунок 2.8).

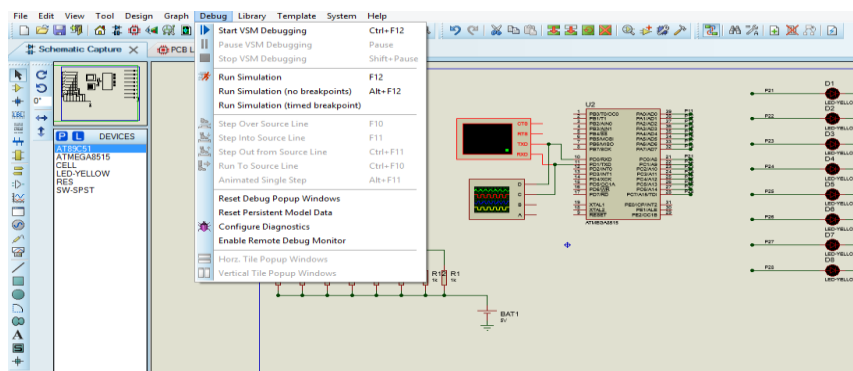


Рисунок 2.8 – Запуск процесу моделювання

2.6.4 Встановлення двох точок зупинки під час виконання програми

Для встановлення двох точок зупинки під час виконання програми натискається кнопка «Pause VSM Debugging». У вкладці «Source Code» встановлюються дві точки зупинки. Для цього потрібно лівою кнопкою миші двічі клацнути з лівої сторони двох команд, як показано на рисунку 2.9.

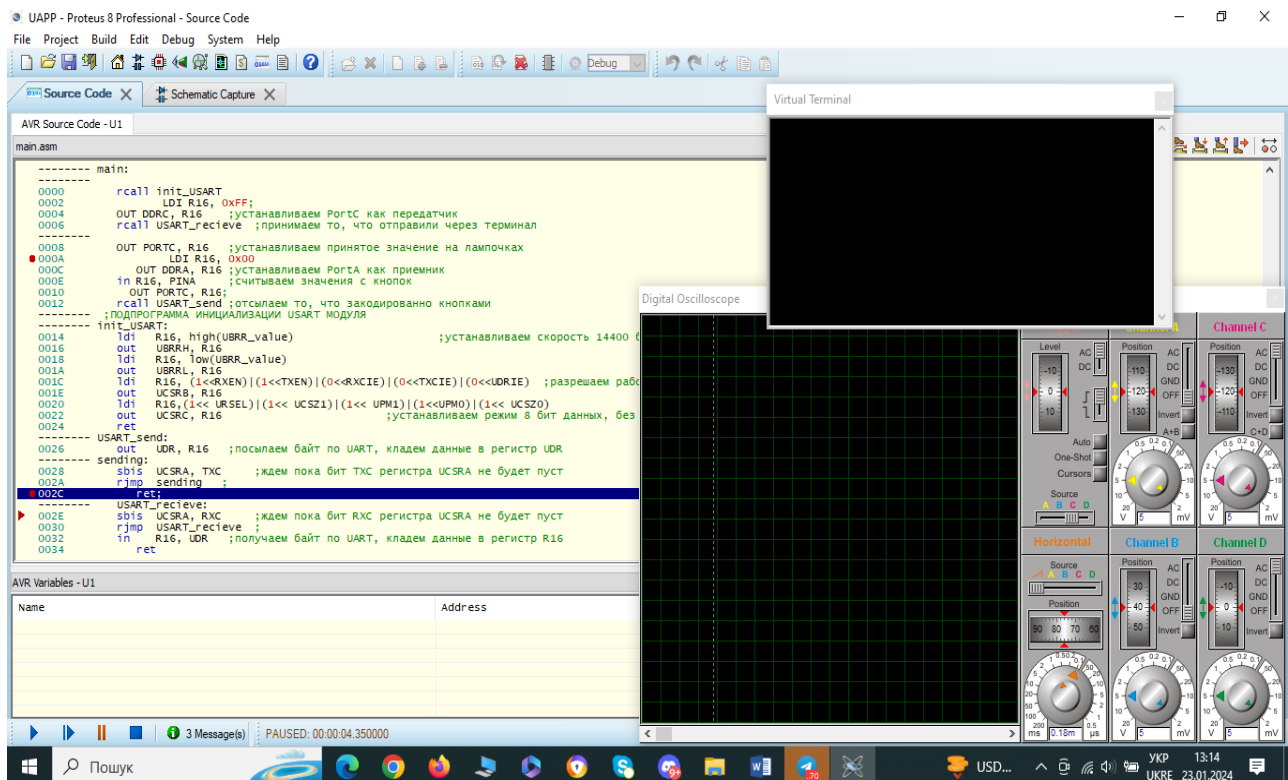


Рисунок 2.9 – Встановлення двох точок зупинки

2.6.5 Введення символу, який надсилається від віртуального терміналу

У разі введення символу від віртуального терміналу потрібно виконати наступне:

- 1) у вікні «Virtual Terminal» натисканням правої кнопки миші обираються: «Hex Display Mod» та «Echo Typed Characters» (рисунок 2.10). Якщо вікно Virtual Terminal закрито, то його можна відкрити із вкладки Debug.

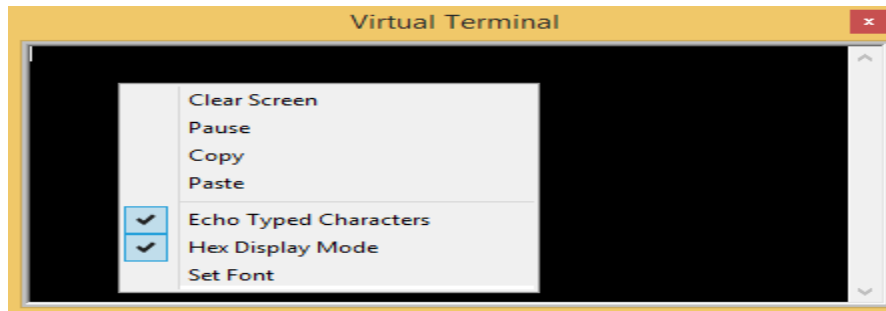


Рисунок 2.10 – Встановлення опцій віртуального термінала

2) виконати налаштування осцилографа згідно рисунком 2.11.

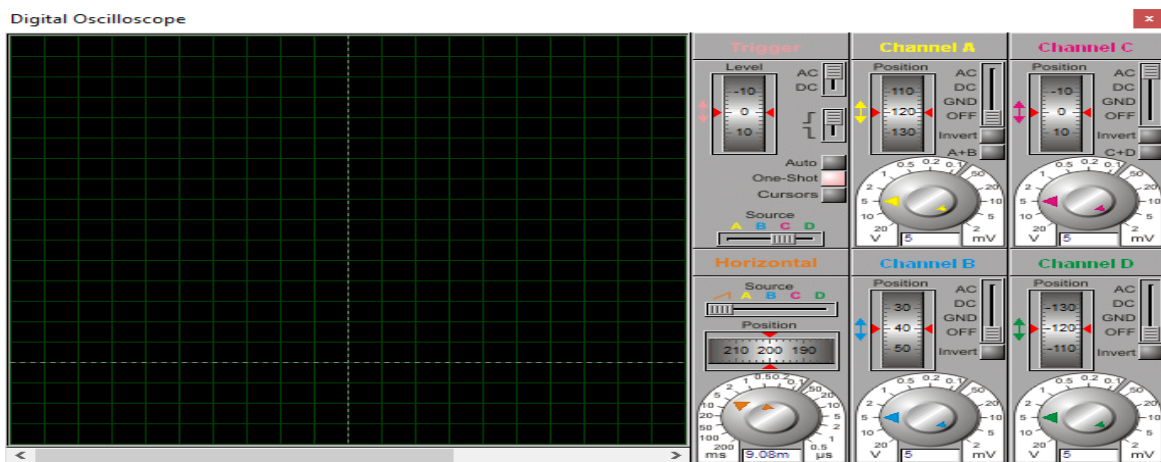


Рисунок 2.11 – Налаштування осцилографа

3) натиснути «Run Simulation» (F12).

4) перед введенням символу у вікні термінала повинен стояти курсор.

5) ввести символ, який надсилається від віртуального термінала до мікроконтролера через УАПП. Символ, який вводиться з клавіатури, віртуальний термінал перетворює в ASCII-код згідно таблиці 2.10.

Наприклад, вводиться цифра 7, ASCII-код якої 37 (hex) = 0011 0111 (bin), в у таблиці 2.10 виділено жовтим кольором.

Таблиця 2.10 – Код ASCII

DEC	HEX		CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	0			53	35	5	106	6A	j	159	9F	я
1	1			54	36	6	107	6B	k	160	A0	a
2	2			55	37	7	108	6C	l	161	A1	б

3	3			56	38	8	109	6D	m	162	A2	в
4	4			57	39	9	110	6E	n	163	A3	г
5	5			58	3A	:	111	6F	o	164	A4	д
6	6			59	3B	;	112	70	p	165	A5	е
7	7			60	3C	<	113	71	q	166	A6	ж
8	8			61	3D	=	114	72	r	167	A7	з
9	9			62	3E	>	115	73	s	168	A8	и
10	A			63	3F	?	116	74	t	169	A9	й
DEC	HEX		CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
11	B			64	40	@	117	75	u	170	AA	к
12	C			65	41	A	118	76	v	171	AB	л
13	D			66	42	B	119	77	w	172	AC	м
14	E			67	43	C	120	78	x	173	AD	н
15	F			68	44	D	121	79	y	174	AE	о
16	10			69	45	E	122	7A	z	175	AF	п
17	11			70	46	F	123	7B	{	176	B0	▒
18	12			71	47	G	124	7C		177	B1	▓
19	13			72	48	H	125	7D	}	178	B2	░
20	14			73	49	I	126	7E	~	179	B3	
21	15			74	4A	J	127	7F		180	B4	┌
22	16			75	4B	K	128	80	A	181	B5	┐
23	17			76	4C	L	129	81	Б	182	B6	└
24	18			77	4D	M	130	82	В	183	B7	┘
25	19			78	4E	N	131	83	Г	184	B8	┌
26	1A			79	4F	O	132	84	Д	185	B9	┐
27	1B			80	50	P	133	85	Е	186	BA	└
28	1C		81	51	Q	134	86	Ж	187	BB	┘	
29	1D		82	52	R	135	87	З	188	BC	┌	
30	1E		83	53	S	136	88	И	189	BD	┐	
31	1F		84	54	T	137	89	Й	190	BE	└	

32	20		85	55	U	138	8A	K	191	BF	ᵏ	
33	21	!	86	56	V	139	8B	Л	192	C0	ℓ	
34	22	“	87	57	W	140	8C	M	193	C1	⊥	
35	23	#	88	58	X	141	8D	H	194	C2	⊤	
36	24	\$	89	59	Y	142	8E	O	195	C3	⊢	
37	25	%	90	5A	Z	143	8F	Π	196	C4	–	
38	26	&	91	5B	[	144	90	P	197	C5	⊥	
39	27	'	92	5C	\	145	91	C	198	C6	⊢	
DEC	HEX		CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
40	28	(	93	5D	]	146	92	T	199	C7	⊢	
41	29	)	94	5E	^	147	93	Y	200	C8	ℓ	
42	2A	*	95	5F	–	148	94	Φ	201	C9	ℓ	
43	2B	+	96	60	'	149	95	X	202	CA	ℓ	
44	2C	'	97	61	a	150	96	Ц	203	CB	⊤	
45	2D	D	98	62	b	151	97	Ч	204	CC	⊢	
46	2E	.	99	63	c	152	98	Ш	205	CD	=	
47	2F	/	100	64	d	153	99	Щ	206	CE	⊢	
48	30	0	101	65	e	154	9A	Ъ	207	CF	⊤	
49	31	1	102	66	f	155	9B	Ы	208	D0	ℓ	
50	32	2	103	67	g	156	9C	Ь	209	D1	⊤	
51	33	3	104	68	h	157	9D	Э	210	D2	π	
212	D4	ℓ	224	E0	p	236	EC	ь	248	F8	ï	
213	D5	ƒ	225	E1	c	237	ED	э	249	F9	ï	
214	D6	π	226	E2	τ	238	EE	ю	250	FA	ÿ	
215	D7	⊢	227	E3	γ	239	EF	я	251	FB	ÿ	
216	D8	⊢	228	E4	φ	240	F0	Ё	252	FC	ⁿ	
217	D9	┘	229	E5	x	241	F1	ё	253	FD	²	
218	DA	г	230	E6	ц	242	F2	Г	254	FE	■	
219	DB	■	231	E7	ч	243	F3	г	255	FF		
220	DC	■	232	E8	ш	244	F4	€				

221	DD	█	233	E9	Щ	245	F5	є				
222	DE	█	234	EA	Ъ	246	F6	І				
223	DF	█	235	EB	Ы	247	F7	і				

Програма перейде на рядок, де встановлено першу точку зупину.

У вкладці зі схемою *Schematic Capture* на світлодіодах та осцилографі відображається символ, який прийнято від терміналу через виходи *порту C* (рисунок 2.12).

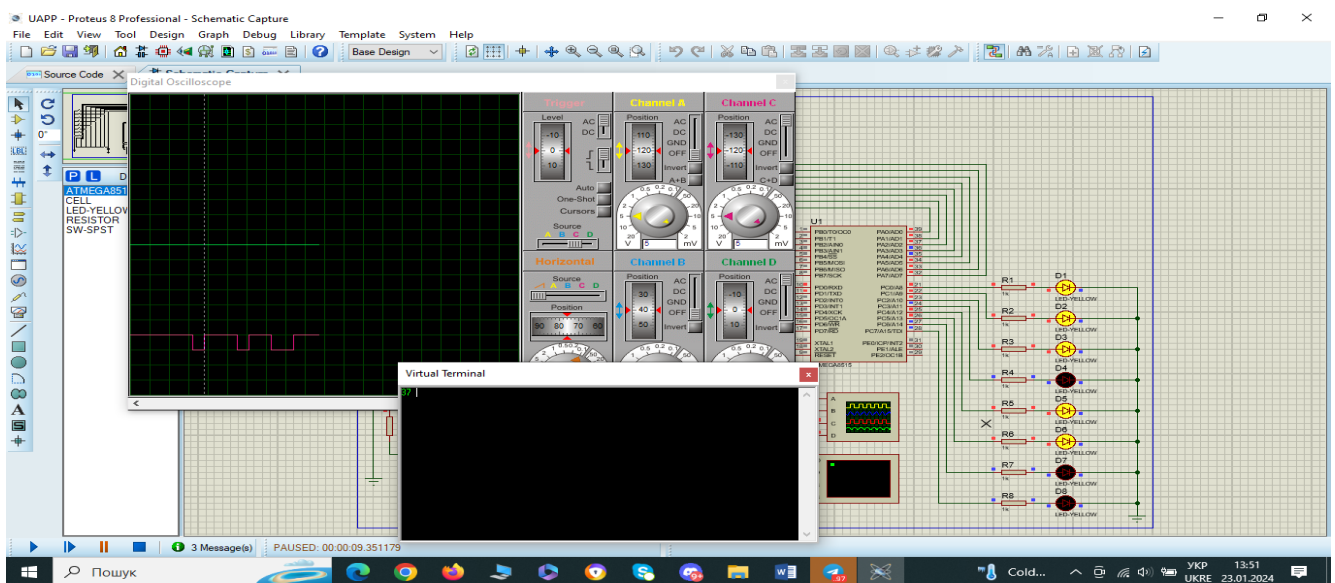


Рисунок 2.12 – Відображення на світлодіодах та осцилографі символу 7, який прийнято від терміналу

Світлодіоди відображають бінарний код символу, який ми прийняли від терміналу (якщо дивитися *знизу уверх*, а бінарне число читати *зліва на право*), нуль – світлодіод вимкнено, одиниця – світлодіод увімкнено (прийнято символ 00110111_b = 37_h). Нижній (червоний) луч осцилографа також вірно відображає прийнятий символ, який було передано від терміналу в асинхронному старт-стопному режимі. Після останнього другого нульового біта перед одиничним стоп-бітом іде одиничний біт, тому що сума одиниць у символі непарна та згідно (2.1) – $P_{ODD} = 1$. Якщо ми, наприклад, від терміналу передаємо цифру 9, ASCII-код якої 39 (hex) = 00111001 (bin), то після другого нульового біта перед одиничним

стоп-бітом іде третій нульовий біт, тому що сума одиниць у символі парна та згідно (2.1) – $P_{0DD} = 0$ (рисунок 2.13).

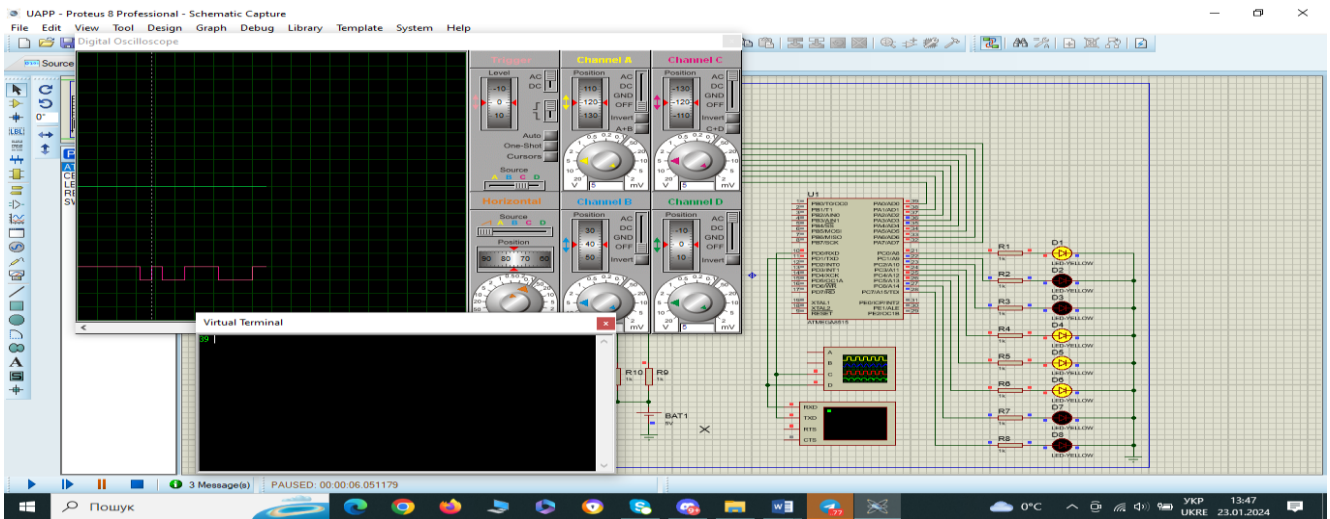


Рисунок 2.13 – Відображення на світлодіодах та осцилографі символу 9, який прийнято від терміналу

2.6.6 Введення символу, який надсилається від мікроконтролера через УАПІ

1) Число для передачі від мікроконтролера через УАПІ кодується кнопками: замкнена кнопка – нуль, а розімкнена – одиниця (рисунок 2.14).

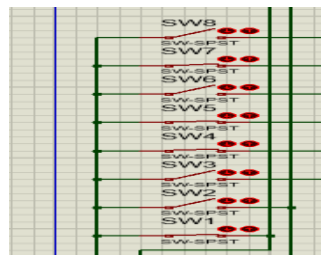


Рисунок 2.14 – Кодування кнопками символу для передачі

2) Щоб побачити переданий сигнал на осцилографі, потрібно натиснути кнопку на опції One-Shot (один кадр) (рисунок 2.11).

3) Для продовження виконання програми знову натискається «Run Simulation». Програма переходить до другої точки зупину.

4) Нижче пояснюються сигнали, що відображаються на осцилографі: осцилограф відображає сигнали за двома променями: верхній (зелений) – сигнал, який передається від кнопок, нижній (червоний) – сигнал, який приймається *від терміналу*: 37(hex) (рисунк 2.15).

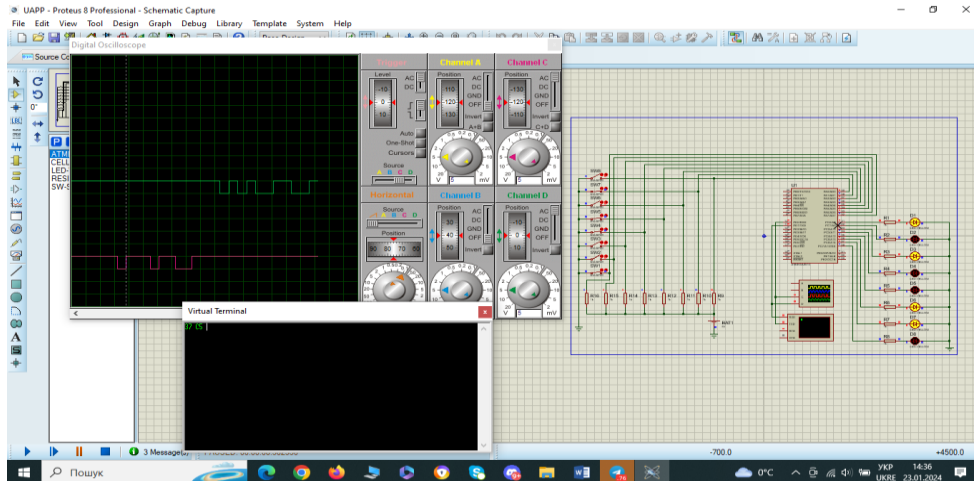


Рисунок 2.15 – Відображення сигналів на осцилографі: від терміналу – червоний; від МК (кнопок) – зелений

Перший перепад сигналу, який передається *від кнопок*, із високого рівня у низький – старт-біт. Потім знаходяться вісім інформаційних бітів, починаючи з молодшого розряду, які відображені на осцилографі, як 01100101 (bin) = 65 (hex).

5) Далі розглянуто ще один приклад, коли на кнопках було закодовано число 67(hex) = 01100111(bin) (рисунк 2.16).

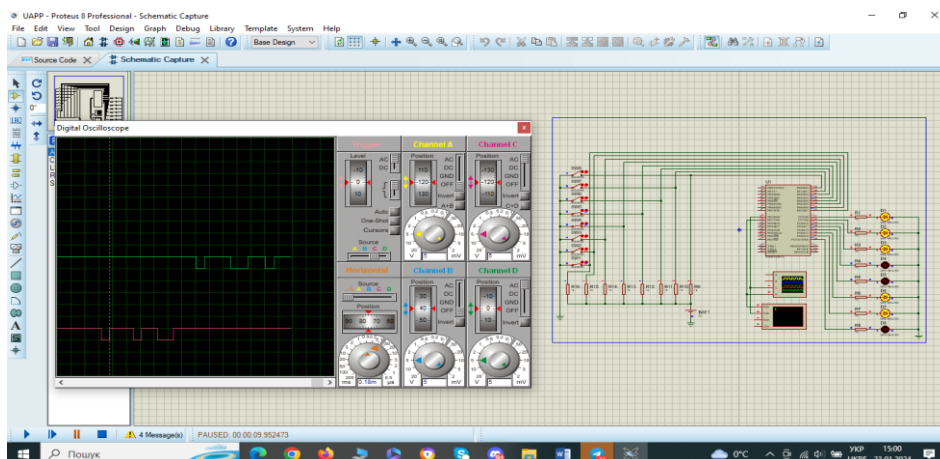


Рисунок 2.16 – Відображення сигналів на осцилографі: від терміналу – червоний, від МК (кнопок) – зелений

На верхньому промені осцилографа відображено: перший нуль – старт-біт, потім три одиниці та один нуль – це число 7 на чотирьох верхніх кнопках, далі йде нуль, дві одиниці та нуль – це число 6 на чотирьох нижніх кнопках, потім біт доповнення до парності – одиниця, та стоп-біт – одиниця (рисунки 2.17).

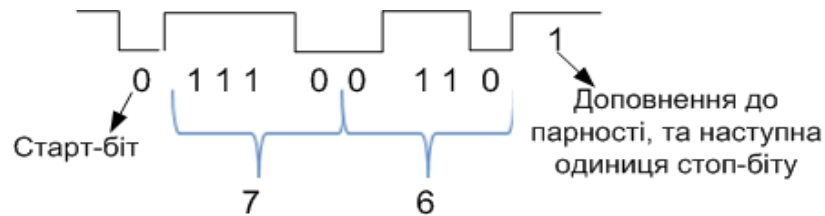


Рисунок 2.17 – Передача числа: 67 (hex) = 01100111 (bin)

2.6.7 Визначення отриманої швидкості передачі

Для визначення отриманої швидкості передачі, яку в даному прикладі було задано – 9600 біт/сек = 9600 пос/сек = 9600 бод, налаштовується така розгортка сигналу на осцилографі, щоб сумістити бічні сторони імпульсу із вертикальними лініями сітки осцилографа (рисунки 2.18).

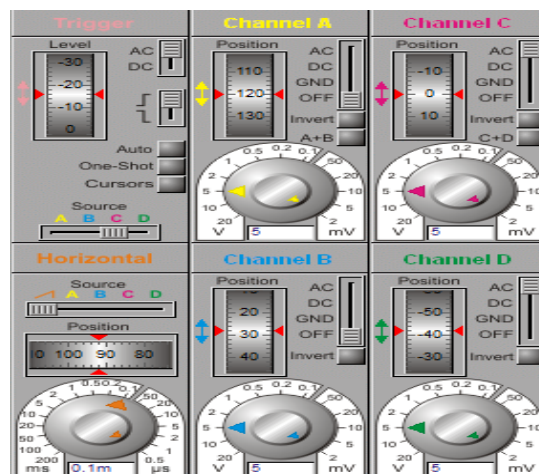


Рисунок 2.18 – Визначення отриманої швидкості передачі на осцилографі

Як видно з рисунку 2.18, тривалість одного імпульсу дорівнює приблизно 0,104 мс. Якщо розділити один біт на цей час, та помножити на 1000, отримується задана швидкість

$$V = \frac{1}{0.104} \cdot 1000 = 9600 \frac{\text{біт}}{\text{сек}}.$$

2.7 Розробка та опис структурної схеми пристрою

Нижче на рисунку 2.19 наведено схему електричну структурну пристрою обміну інформацією між модулем УАПП мікроконтролера ATmega 8515 та віртуальним терміналом. Основним вузлом структури є мікроконтролер, який керує обміном інформацією.

Через паралельний порт введення мікроконтролер отримує сигнали від восьми кнопок, якими кодується байт для передачі. Цей байт через порт виведення передається до світлодіодів. Через модуль УАПП мікроконтролера байт, який закодовано кнопками, з виходу TxD1 передається на вхід RxD1 віртуального терміналу у послідовному асинхронному старт-стопному режимі. Цей байт також відображається на екрані осцилографа.

З виходу TxD2 віртуального терміналу у послідовному асинхронному старт-стопному режимі на вхід RxD2 модуля УАПП подається ASCII-код символу, який вводиться з клавіатури персонального комп'ютера. Цей код також відображається на екрані осцилографа.

Кварцовий резонатор (КР) визначає частоту високочастотного генератора, схему якого вбудовано у мікроконтролер.

Схема скидання (ССКИД) формує сигнал «RESET» (СКИДАННЯ), який запускає виконання робочої програми спочатку. Цей сигнал формується автоматично при включенні живлення, або від зовнішньої кнопки КСКИД.

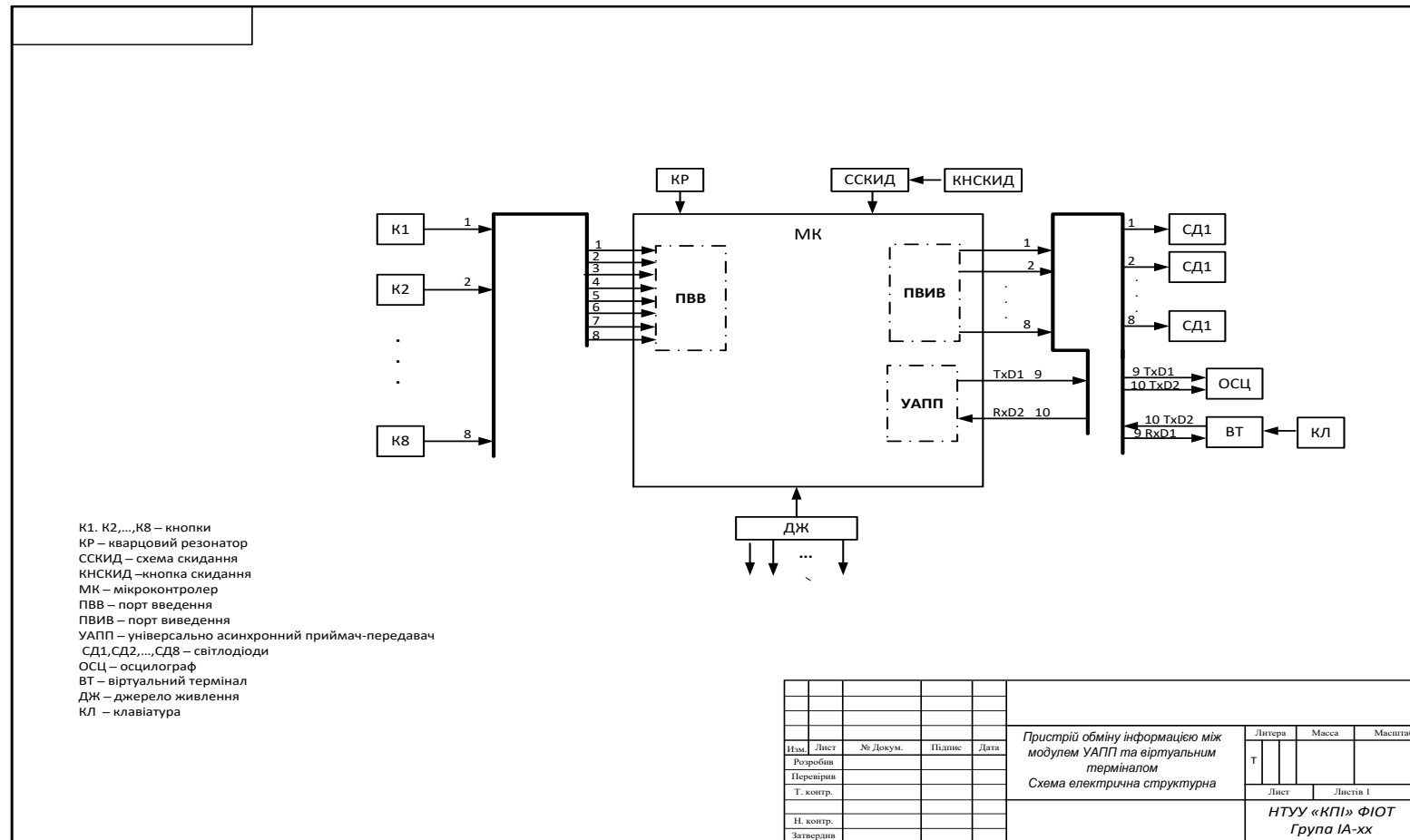


Рисунок 2.19 – Схема електрична структурна пристрою обміну інформацією між модулем УАПП мікроконтролера АТmega 8515 та віртуальним терміналом

2.8 Розробка текстових документів до роботи

Пояснювальна записка та інші текстові документи до курсового проєкту розробляються у Word згідно переліку, який наведено у підрозділі 1.3 методички до виконання КП.

3 ПРОЄКТУВАННЯ ПРИСТРОЮ КЕРУВАННЯ ДВИГУНОМ ПОСТІЙНОГО СТРУМУ

3.1 Завдання

Виконати проєктування пристрою керування двигуном постійного струму:

- 1) розробити, навести та описати робочу модель в пакеті PROTEUS;
- 2) зробити розрахунок ШІМ-модуля;
- 3) розробити схему алгоритму роботи пристрою;
- 4) розробити керуючу програму мовою Асемблер;
- 5) розробити керуючу програму мовою C;
- 6) виконати моделювання пристрою в пакеті PROTEUS:
 - навести «скріншоти», які пояснюють роботу моделі та підтверджують правильність виконаних розрахунків, схеми алгоритму роботи та робочої керуючої програми;
- 7) розробити та описати структурну схему пристрою;
- 8) розробити пояснювальну записку та інші текстові документи до курсового проєкту у Word згідно переліку, який наведено у підрозділі 1.3 цієї методички.

Вихідні дані:

- тип МК-ра: AT mega 128;
- номер таймера: T/C1;
- частота вихідного ШІМ-сигналу: $f_{\text{шім}} = 25 \text{ кГц}$;
- частота тактового сигналу підсистеми введення/виведення: $f_{\text{CLK I/O}} = 16 \text{ МГц}$;
- шпаруватість ШІМ-сигналу: $Q_{\text{шім}} = 2$;
- режим роботи модуля ШІМ: Phase and Frequency Correct PWM;
- лінія порту мікроконтролера, на якій формується ШІМ-сигнал, – PB5;
- вид вихідного ШІМ-сигналу – інвертований.

3.2.2 Опис моделі

З лівого боку моделі наведено кнопки керування: K1, K2, K3, K4, на які через резистори R1, R2, R3, R4 відповідно, подається напруга живлення. У правому нижньому куті зображено двигун постійного струму (DC Motor), а трохи лівіше – мікросхему L293D, через яку мікроконтролер керує двигуном за допомогою ШІМ-сигналу. В якості мікроконтролера обрано мікросхему ATmega128, яка знаходиться у правому куті моделі.

Зовнішні резистори використовуються через те, що в середині мікроконтролера на відповідних входних лініях не підключені внутрішні підтягуючі резистори.

На рисунку 3.2 наведено збільшений вигляд кнопок керування та порядок їх розташування.

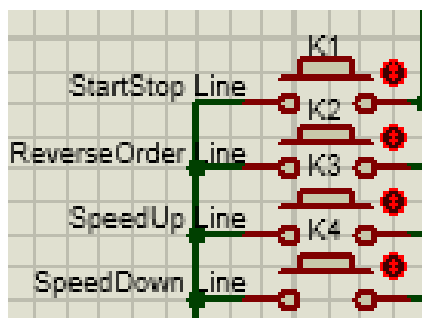


Рисунок 3.2 – Порядок розташування кнопок керування

У вихідному стані кнопки є нормально розімкненими. В цьому випадку на відповідні лінії порту В (PB) мікроконтролера подаються високі рівні напруги, тобто логічні одиниці. Коли кнопки замикаються, на входи мікроконтролера подаються низькі рівні напруги, тобто логічні нулі. Перемикання сигналу з логічного нуля у логічну одиницю від обраної кнопки сприймається мікроконтролером як відповідний сигнал керування.

Кнопка K1 («StartStop Line») відповідає за вмикання та вимикання двигуна. Тобто коли двигун вимкнено, то вона його ввімкне. Коли двигун ввімкнений та обертається, то вона його вимкне. Кнопку K1 підключено до 12-го виводу мікроконтролера – PB2, який повинен бути запрограмований як вхід.

Кнопка K2 («ReverseOrder Line») відповідає за зміну напрямку обертання двигуна. Одне натискання – одна зміна напрямку. Потрібно мати на увазі, що двигун пристрій інерційний. Він не може зупинитися миттєво та одразу почати обертатися в інший бік. Після зміни напрямку програмно, у нього витрачається певний період часу, щоб фізично зупинитись, та почати обертатись у іншу сторону. Кнопку K2 підключено до 13-го виводу ATmega128, а саме до PB3, який попередньо запрограмовано як вхід.

Кнопка K3 («SpeedUp Line») відповідає за збільшення швидкості обертання двигуна шляхом поступового зменшення шпаруватості та збільшення постійної еквівалентної напруги імпульсного ШІМ-сигнала до максимуму, який дорівнює амплітуді імпульса – значенню напруги живлення на виводі VS мікросхеми L293D (рисунок 3.3).

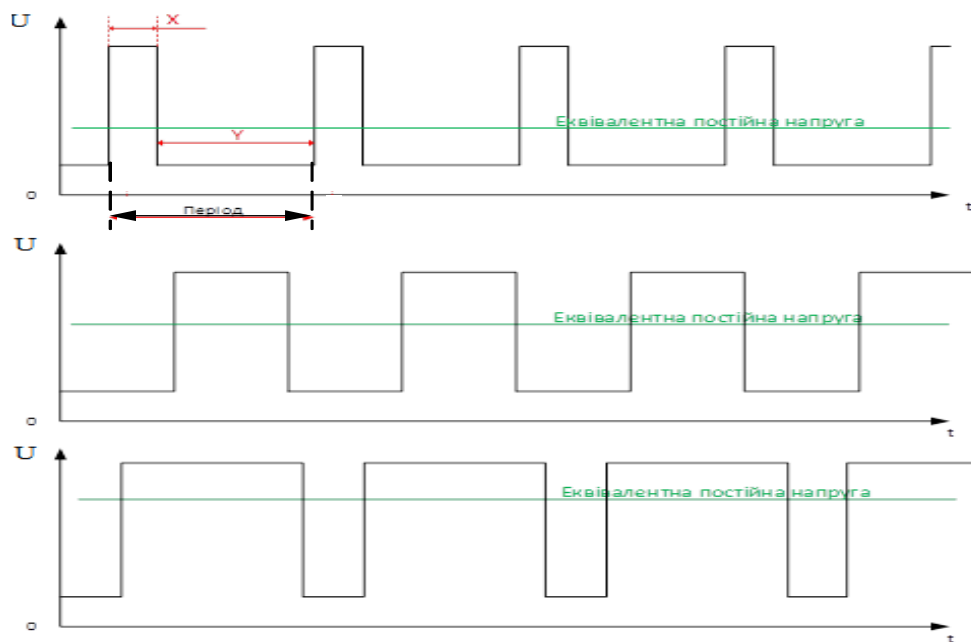


Рисунок 3.3 – Формування імпульсного ШІМ-сигналу

Оскільки двигун інерційний, то швидкість обертання збільшується не стрибкоподібно, а поступово, і потребує певного проміжку часу, щоб вийти на новий програмно встановлений рівень. Кнопку K3 підключено до 14-го виводу мікроконтролера – PB4, який повинен бути запрограмований як вхід.

Кнопка К4 («SpeedDown Line») відповідає за зменшення швидкості обертання двигуна, для чого вона поступово збільшує шпаруватість та зменшує значення постійної еквівалентної напруги імпульсного ШІМ-сигнала (рисунок 3.3).

Через інерційність двигуна швидкість обертання зменшується не стрибками, а поступово і вимагає для свого зменшення певного проміжку часу. Кнопку К4 підключено до 16-го виводу мікроконтролера – PB6, який повинен бути запрограмований як вхід.

На рисунку 3.4 наведено підключення до мікроконтролера мікросхеми L293D (драйвера ШІМ сигналу).

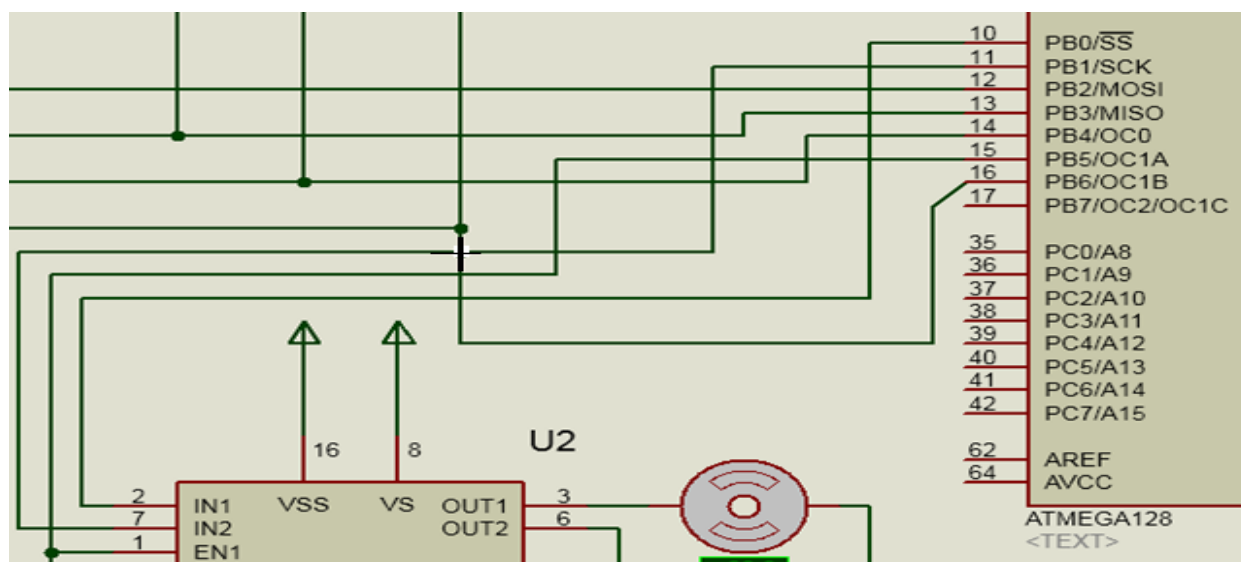


Рисунок 3.4 – Підключення мікросхеми L293D та мікроконтролера ATmega128

До входів IN1 та IN2 цієї мікросхеми підключені виводи 10 та 11 мікроконтролера: PB0 та PB1, які запрограмовані як виходи.

В залежності від комбінації сигналів керування на цих входах – нуль або одиниця, двигун буде або стояти, або обертатись у відповідному напрямку.

15-й вивід мікроконтролера – PB5 запрограмовано як вихід і підключено до входу EN1 мікросхеми L293D. З цього виходу мікроконтролера знімається ШІМ-сигнал. Вхід EN1 відповідає за роботу першої мостової схеми у складі

мікросхеми (див. пп. 3.2.3.4). Коли на ньому високий рівень, на перший міст подається живлення, яке підведено до виводу VS мікросхеми (рисунок 3.4).

Коли на вході EN1 низький рівень, на перший міст напруга живлення не подається. Завдяки ШІМ-сигналу на цьому вході можна змінювати швидкість обертання двигуна.

Два входи GND мікросхеми L293D заземлено. На вхід VSS (рисунок 3.4) подається напруга живлення самої мікросхеми, а на вхід VS – напруга живлення двигуна.

На рисунку 3.5 зображено підключення двигуна до мікросхеми L293D.

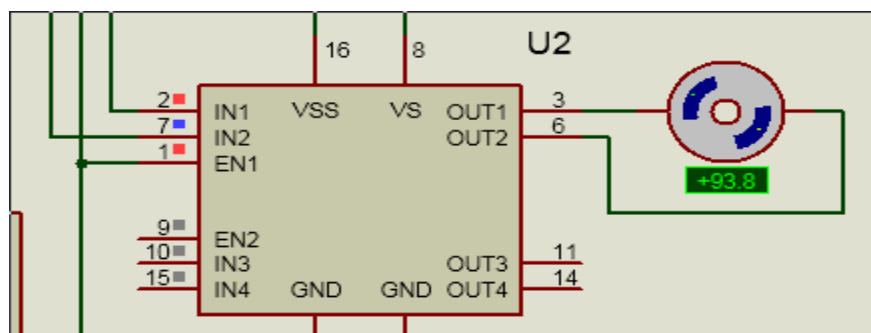


Рисунок 3.5 – Підключення двигуна до мікросхеми L293D

Двигун підключено до 3 та 6 виводів мікросхеми L293D, а саме OUT1 та OUT2. На ці виходи, в залежності від того, чи ввімкнене живлення першої мостової схеми, що керується входом EN1, та в залежності від стану ключів у першій мостовій схемі, які керуються входами IN1 та IN2, або подається напруга живлення, яку ми подали на вхід VS, або напруга не подається.

У випадку, коли на один з виводів – OUT1 чи OUT2 подається напруга живлення, а на інший не подається, виникне різниця потенціалів. Це викличе протікання струму обмоткою збудження двигуна, внаслідок чого він почне обертатися.

Як видно з рисунку 3.5 в моделі сигнал на вході IN1 позначено червоним кольором, тобто від мікроконтролера подано високий рівень (логічну одиницю). Сигнал на вході IN2 позначено синім кольором, тобто подається низький рівень – логічний нуль. За такою комбінацією на входах IN1 та IN2 двигун починає

обертатися за годинниковою стрілкою. Під час протилежної комбінації двигун буде обертатися проти годинникової стрілки. За однакових сигналах на входах IN1 та IN2 двигун обертатися не буде.

На рисунках 3.6...3.9 наведено деякі фрагменти, які демонструють роботу моделі.

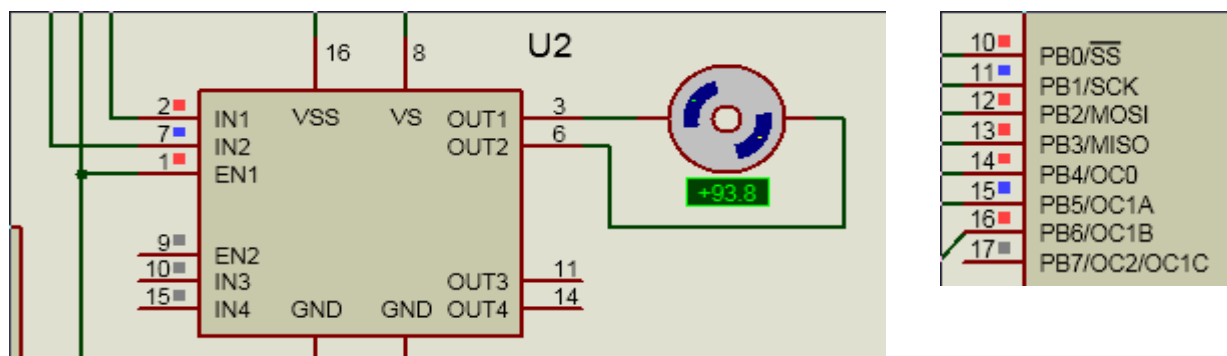


Рисунок 3.6 – Стан моделі після натискання кнопки «старт/стоп»

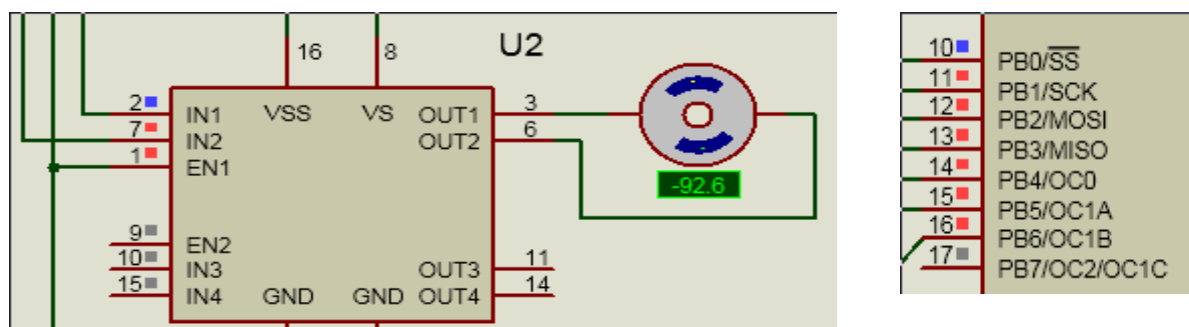


Рисунок 3.7 – Стан моделі після активації режиму реверсу

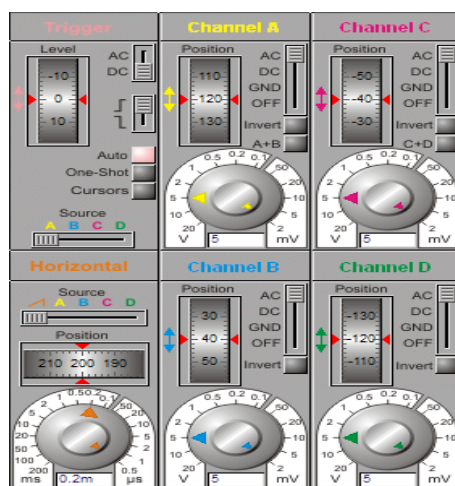


Рисунок 3.8 – Налаштування осцилографа

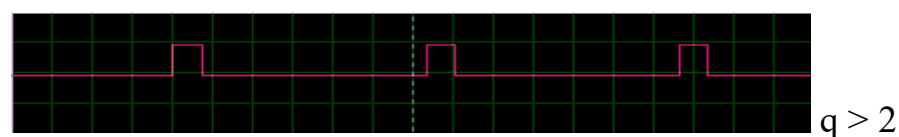
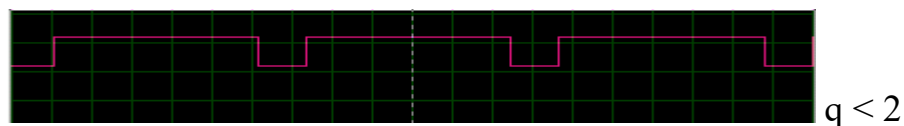
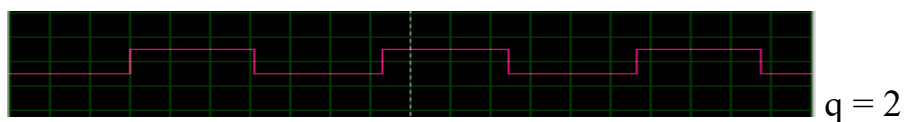


Рисунок 3.9 – ШІМ-сигнал, що подається на двигун
(шпаруватість $q = 2$, $q < 2$, $q > 2$)

На рисунку 3.10 наведено стан усієї системи після запуску сценарію. Шпаруватість ШІМ-сигналу дорівнює 2, але двигун не обертається, тому що $IN1 = IN2 = 0$.

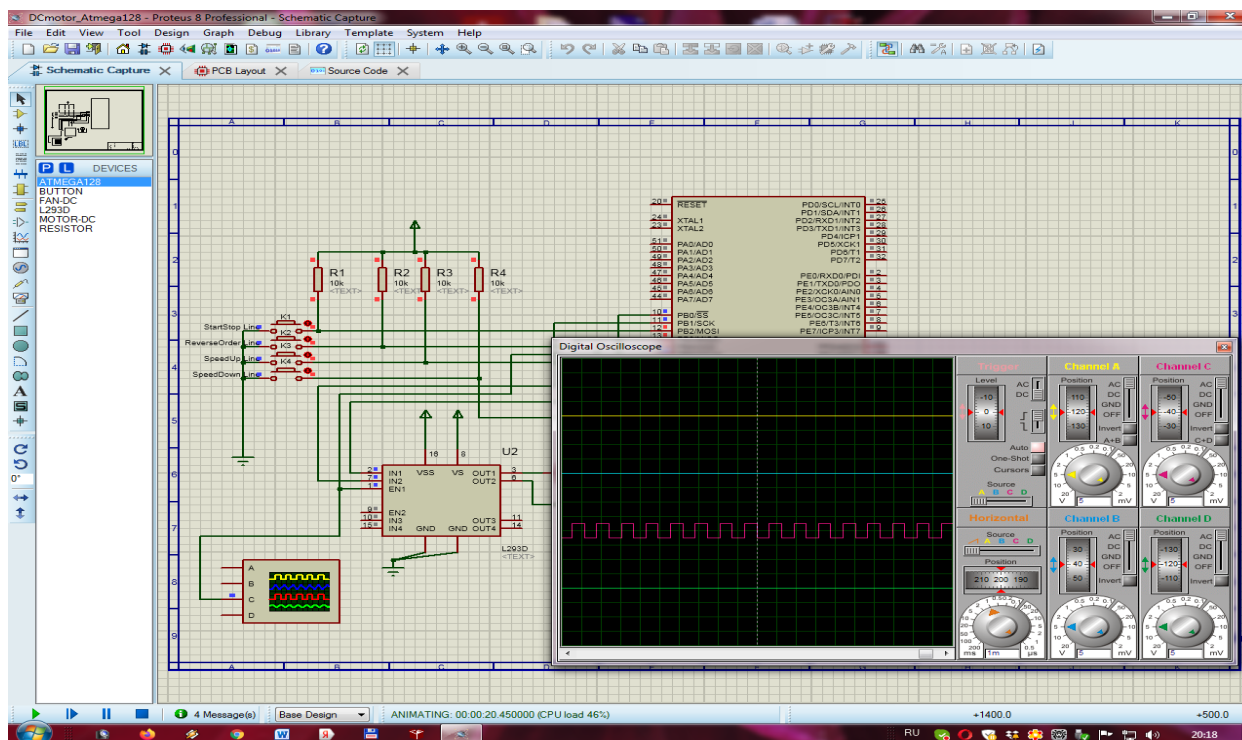


Рисунок 3.10 – Стан усієї системи після запуску сценарію

3.2.3 Опис окремих елементів моделі

3.2.3.1 Мікроконтролер

Для контролю та виконання всіх функцій, які покладено на модель, було обрано мікроконтролер сім'ї AVR – ATmega 128.

На рисунку 3.11 наведено зовнішній вигляд та позначення виводів цього мікроконтролера.

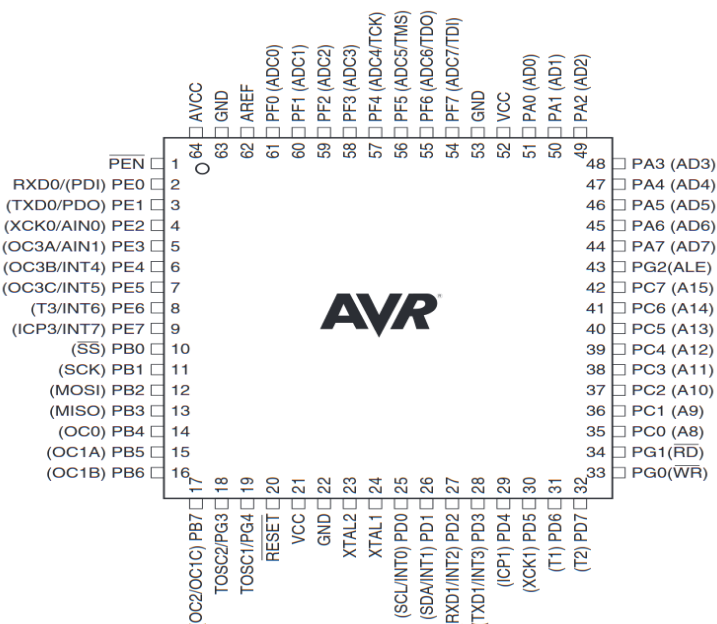


Рисунок 3.11 – Зовнішній вигляд та позначення виводів мікроконтролера

Він є малопотужним 8-розрядним КМОН-мікроконтролером, який має розширену RISC-архітектуру. Його основні характеристики та опис виводів розглянуто у [2].

3.2.3.2 Модуль таймера

Для керування двигуном постійного струму в моделі використовується модуль таймера мікроконтролера, який працює в режимі широтно-імпульсної модуляції.

Регулювання швидкості обертання двигуна забезпечується зміною шпаруватості керуючих імпульсів, що у свою чергу викликає зміну постійної

складової імпульсного сигналу (рисунок 3.3). Чим менше шпаруватість, а відповідно більше величина відношення «тривалість імпульсу/період», тим більша напруга надходить на двигун, який буде обертатись із більшою швидкістю.

3.2.3.3 Двигун

В моделі використовується електродвигун постійного струму. Регулювання швидкості обертання двигуна зазвичай забезпечується за допомогою формування керуючих імпульсів у вигляді сигналів з широтно-імпульсною модуляцією. Подібний підхід дозволяє регулювати середню величину потужності, що надходить на двигун. В даному випадку, чим більше величина відношення «тривалість імпульсу/період» (менше шпаруватість), тим більша потужність надходить на двигун.

Частота сигналу з широтно-імпульсною модуляцією повинна перевищувати 20 кГц, що дозволяє виключити звукові ефекти, пов'язані з формуванням звукових сигналів самим двигуном при зміні магнітного поля, яке в ньому утворюється [2].

Для формування ШІМ-сигналів в моделі використовується модуль таймера (T/C1) мікроконтролера.

3.2.3.4 Драйвер ШІМ

Двигун постійного струму підключається до мікроконтролера за допомогою відповідної мостової схеми – драйвера, яку зображено на рисунку 3.12.

Подібну схему мають драйвери двигунів, які використовуються для перетворення керуючих сигналів малої потужності від мікроконтролера у сигнали, потужність яких достатня для керування двигунами.

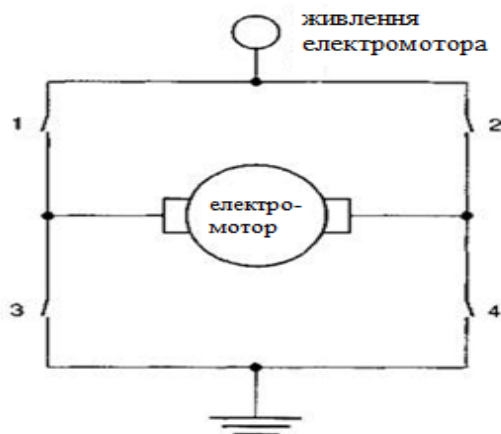


Рисунок 3.12 – Мостова схема підключення двигуна постійного струму до мікроконтролера

Окрім підсилення потужності керуючих сигналів драйвери керують зміною напрямку обертання двигунів. Існує декілька схем драйверів, які відрізняються як потужністю, так і елементною базою, на якій вони виконані.

В моделі застосовано драйвер, який виконано у вигляді готової до роботи мікросхеми – L293D. Її зовнішній вигляд показано на рисунку 3.13.



Рисунок 3.13 – Зовнішній вигляд мікросхеми L293D

Мікросхема L293D включає два драйвери та може керувати двома електродвигунами відповідної потужності. Також мікросхема забезпечує розділене живлення для самої мікросхеми та для керованих нею двигунів, що дозволяє підключати електродвигуни з більшою напругою живлення, чим у мікросхеми. Розділення живлення мікросхеми і електродвигунів необхідне також для зменшення завад, які викликані стрибками напруги, що пов'язані з роботою двигунів.

Обидва драйвери, що входять у склад мікросхеми мають ідентичні принципи роботи, тому нижче розглянуто принцип роботи лише одного з них, спрощений вид якого наведено на рисунку 3.14.

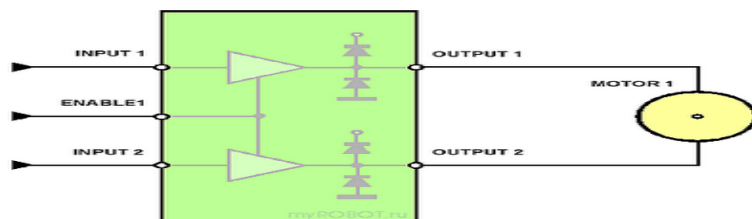


Рисунок 3.14 – Спрощений вид драйвера мікросхеми L293D

До виходів OUTPUT1 та OUTPUT2 підключається двигун постійного струму MOTOR1. Вхід ENABLE1, який відповідає за ввімкнення драйвера, підключають до додатного полюсу джерела живлення: +5V. Якщо при цьому відповідні керуючі сигнали на входи INPUT1 та INPUT2 не подаються, то двигун обертатися не буде.

Якщо вхід INPUT1 з'єднати з додатним полюсом джерела живлення, а вхід INPUT2 – з від'ємним, то двигун почне обертатися за годинниковою стрілкою.

Якщо з'єднати вхід INPUT1 з від'ємним полюсом джерела струму, а вхід INPUT2 – з додатним, то двигун почне обертатися в іншу сторону.

Якщо на керуючі входи INPUT1 та INPUT2 подати сигнали одного рівня, тобто під'єднати обидва входи до додатного полюсу джерела живлення або до від'ємного, то двигун обертатися не буде.

Якщо прибрати керуючий сигнал зі входу ENABLE1, то за будь-яких варіантах сигналів на входах INPUT1 та INPUT2 двигун також обертатися не буде. На вхід ENABLE1 подається імпульсний ШІМ-сигнал, який формується модулем таймера мікроконтролера. Змінюючи шпаруватість цього сигналу ми змінюємо постійну складову імпульсного сигналу, що в свою чергу змінює швидкість обертання двигуна. На рисунку 3.15 показано нумерацію та позначення виводів мікросхеми L293D.

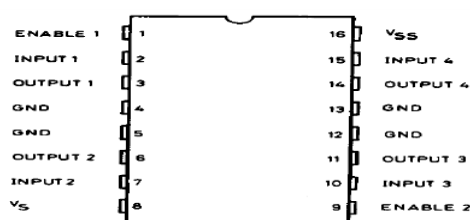


Рисунок 3.15 – Нумерація та позначення виводів мікросхеми L293D

Нижче описано призначення виводів мікросхеми:

- входи ENABLE1 та ENABLE2 відповідають за вмикання відповідного драйвера, що входить у склад мікросхеми. На ці входи подається ШІМ-сигнал;
- входи INPUT1 та INPUT2 керують напрямом обертання двигуна, який підключено до виходів OUTPUT1 та OUTPUT2;
- входи INPUT3 та INPUT4 керують напрямом обертання другого двигуна, який підключено до виходів OUTPUT3 та OUTPUT4;
- контакт Vs відповідає за живлення електродвигунів. Його з'єднують з додатним полюсом джерела живлення двигунів або просто з додатним полюсом джерела живлення мікросхеми, якщо схема і двигуни живляться від одного джерела;
- контакт Vss з'єднують з додатним полюсом джерела живлення. Цей контакт забезпечує живлення самої мікросхеми;
- чотири контакти GND з'єднують з «землею» (спільним дротом – від'ємним полюсом джерела живлення). Крім того, за допомогою цих контактів зазвичай забезпечують тепловідвід від мікросхеми, тому їх краще всього паяти на достатньо широку контактну поверхню.

Дана мікросхема має наступні технічні характеристики:

- напруга живлення двигунів Vs: 4,5...36 В;
- напруга живлення мікросхеми Vss: +5 В;
- допустимий струм навантаження на кожен канал: 600 мА;
- максимальний струм на виході на кожен канал: 1,2 А;
- логічний нуль вхідної напруги: до 1,5 В;
- логічна одиниця вхідної напруги: 2,3...7 В;
- швидкість перемикачів: до 5 кГц;
- мікросхема має захист від перегріву.

Інший спосіб керування двигунами постійного струму засновано на використанні мостових схем типу L298N [1; 2]. Це двоканальний пристрій, який працює від ТТЛШ-рівнів з напругою до 46 В та струмом до 2 А на кожен канал.

Для підвищення до 5А значення вихідного струму, що надходить безпосередньо на обмотку двигуна постійного струму, на вихід схеми підключають додаткові пари транзисторів, шунтуючи їх захисними діодами [1].

Захисні діоди

Як видно з рисунку 3.14 схема драйверів містить захисні діоди. Ці діоди призначено для захисту транзисторних ключів у складі драйверів від додатних і від’ємних паразитних імпульсів досить високої амплітуди, які з’являються на обмотках двигуна при комутації обмоток. Додатні імпульси виникають при запиранні (вимиканні) ключів, а від’ємні – при включенні. Механізм виникнення цих імпульсів описано у [1].

3.3 Розрахунок ШІМ-модуля та програмування його мовою Асемблер

На рисунку 3.16 наведено формування таймером мікроконтролера ШІМ-сигналу в режимі Phase and Frequency Correct PWM [2].

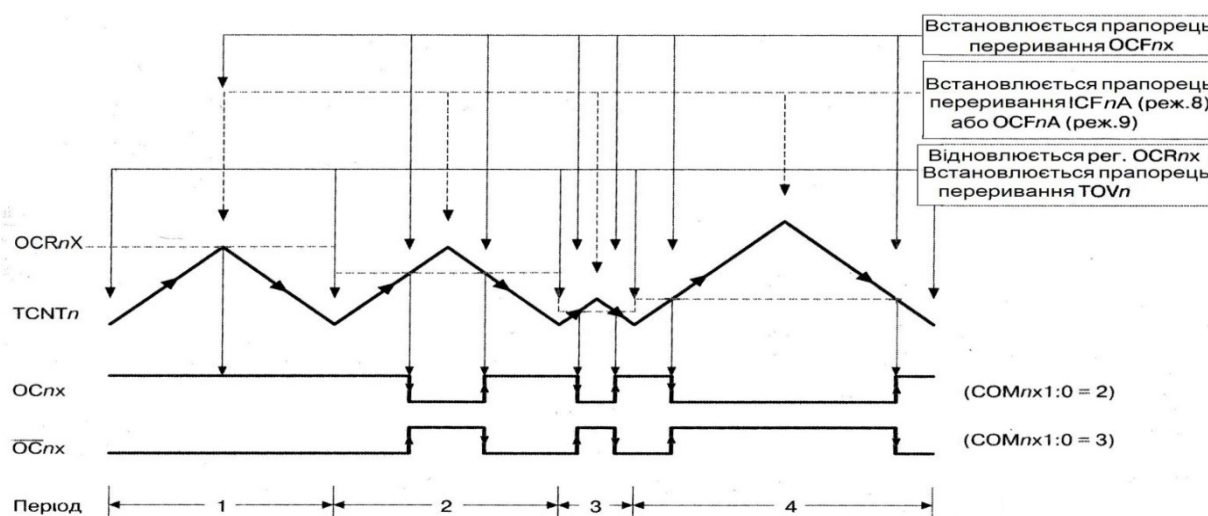


Рисунок 3.16 – Формування таймером ШІМ-сигналу в режимі Phase and Frequency Correct PWM

Величина частоти ШІМ-сигналу f_{PB5} визначається виразом [2]:

$$f_{OCnx} = f_{clkI/O} / 2 \cdot N \cdot TOP, \quad (3.1)$$

де N – коефіцієнт ділення попереднього дільника таймера; TOP – модуль лічби; $f_{clkI/O}$ – частота генератора тактових імпульсів підсистеми введення/виведення мікроконтролера.

Згідно таблиці 3.1 обираємо номер режиму роботи таймера – 8.

Таблиця 3.1 – Режими роботи 16-розрядних таймерів/лічильників

Номер режиму	WGMn3	WGMn2	WGMn1	WGMn0	Режим роботи таймера/лічильника T_n	Модуль лічби (TOP)	Оновлення регістрів TOV_n	Момент встановлення прапорця TOV_n
0	0	0	0	0	Normal	\$FFFF	Негайно	\$FFFF
1	0	0	0	1	Phase correct PWM, 8-розрядний	\$00FF	При TOP	\$0000
2	0	0	1	0	Phase correct PWM, 9-розрядний	\$01FF	При TOP	\$0000
3	0	0	1	1	Phase correct PWM, 10-розрядний	\$03FF	При TOP	\$0000
4	0	1	0	0	CTC (скидання за збігом)	OCRnA	Негайно	\$FFFF
5	0	1	0	1	Fast PWM, 8-розрядний	\$00FF	При TOP	При TOP
6	0	1	1	0	Fast PWM, 9-розрядний	\$01FF	При TOP	При TOP
7	0	1	1	1	Fast PWM, 10-розрядний	\$03FF	При TOP	При TOP
8	1	0	0	0	Phase and Frequency Correct PWM	ICRn	\$0000	\$0000
9	1	0	0	1	Phase and Frequency Correct PWM	OCRnA	\$0000	\$0000
10	1	0	1	0	Phase correct PWM	ICRn	При TOP	\$0000
11	1	0	1	1	Phase correct PWM	OCRnA	При TOP	\$0000
12	1	1	0	0	CTC (скидання за збігом)	ICRn	Негайно	\$FFFF
13	1	1	0	1	Зарезервовано	—	—	—
14	1	1	1	0	Fast PWM	ICRn	При TOP	При TOP
15	1	1	1	1	Fast PWM	OCRnA	При TOP	При TOP

Примітка. $n = 1, 3, 4$ або 5 .

В цьому режимі значення модуля лічби TOP визначається вмістом регістра ICR1. Згідно формули 3.1 за $N = 8$ та значенні f_{PB5} , яке задано в завданні та дорівнює 25кГц, розраховуємо значення $ICR1 = TOP$:

$$TOP = ICR1 = \frac{f_{CLKI/O}}{2 \cdot N \cdot f_{PB5}} = \frac{16 \cdot 10^6}{2 \cdot 8 \cdot 25 \cdot 10^3} = 40 = 00101000B = \$0028.$$

$$\text{Період ШІМ-сигналу: } T_{PB5} = \frac{1}{f_{PB5}} = \frac{1}{25000} = 40 \text{ мкс.}$$

За шпаруватості ШІМ-сигналу $Q_{PB5} = 2$ потрібна тривалість імпульсу

$$t_{\text{импPB5}} = \frac{T_{PB5}}{Q} = \frac{40 \text{ мкс}}{2} = 20 \text{ мкс.}$$

Згідно з рисунком 3.16

$$t_{\text{импPB5}} = T_{PB5} - 2 \Delta t, \quad (3.2)$$

$$\Delta t = \frac{T_{PB5} - t_{\text{импPB5}}}{2} = \frac{40 - 20}{2} = 10 \text{ мкс.} \quad (3.3)$$

Відповідно до роботи таймера в режимі Phase and Frequency Correct PWM (рисунок 3.16)

$$\Delta t = T_{\text{clkI/O}} \cdot N \cdot OCR1A, \quad (3.4)$$

де OCR1A – регістр порівняння каналу А таймера 1.

Із виразу (3.4)

$$OCR1A = \frac{\Delta t}{T_{\text{CLKI/O}} \cdot N} = \frac{10 \cdot 10^{-6}}{\frac{1}{16} \cdot 10^{-6} \cdot 8} = 20 = 00010100B = \$0014.$$

Тоді шпаруватість:

$$\begin{aligned} Q &= \frac{T_{PB5}}{T_{PB5} - 2 \Delta t} = \frac{T_{PB5}}{T_{PB5} - 2 T_{\text{CLKI/O}} \cdot N \cdot OCR1A} = \frac{1}{1 - \frac{2 T_{\text{CLKI/O}} \cdot N \cdot OCR1A}{T_{PB5}}} = \\ &= \frac{1}{1 - 2 \cdot \frac{1}{f_{\text{CLKI/O}}} \cdot N \cdot OCR1A \cdot f_{PB5}} \end{aligned} \quad (3.5)$$

За останньою формулою при $OCR1A = 20$ розрахуємо значення шпаруватості Q , яке повинно дорівнювати 2:

$$Q = \frac{1}{1 - 2 \cdot \frac{1}{f_{CLKI/O}} \cdot N \cdot OCR1A \cdot f_{PB5}} = \frac{1}{1 - 2 \cdot \frac{1}{16 \cdot 10^6} \cdot 8 \cdot 20 \cdot 25 \cdot 10^3} = 2.$$

Отриманий результат говорить про те, що попередні розрахунки зроблено вірно.

Тепер обчислимо значення Q у разі збільшення OCR1A на 5 та у разі зменшення на 5:

$$Q_{(OCR1A=25)} = \frac{1}{1 - 2 \cdot \frac{1}{16 \cdot 10^6} \cdot 8 \cdot 25 \cdot 25 \cdot 10^3} = 2,667,$$

$$Q_{(OCR1A=15)} = \frac{1}{1 - 2 \cdot \frac{1}{16 \cdot 10^6} \cdot 8 \cdot 15 \cdot 25 \cdot 10^3} = 1,6.$$

3.4 Схеми алгоритму роботи пристрою

3.4.1 Схеми алгоритму роботи пристрою, що відповідає завданню

На рисунку 3.17 наведено схеми алгоритму роботи пристрою, який відповідає завданню (підрозд. 3.1).

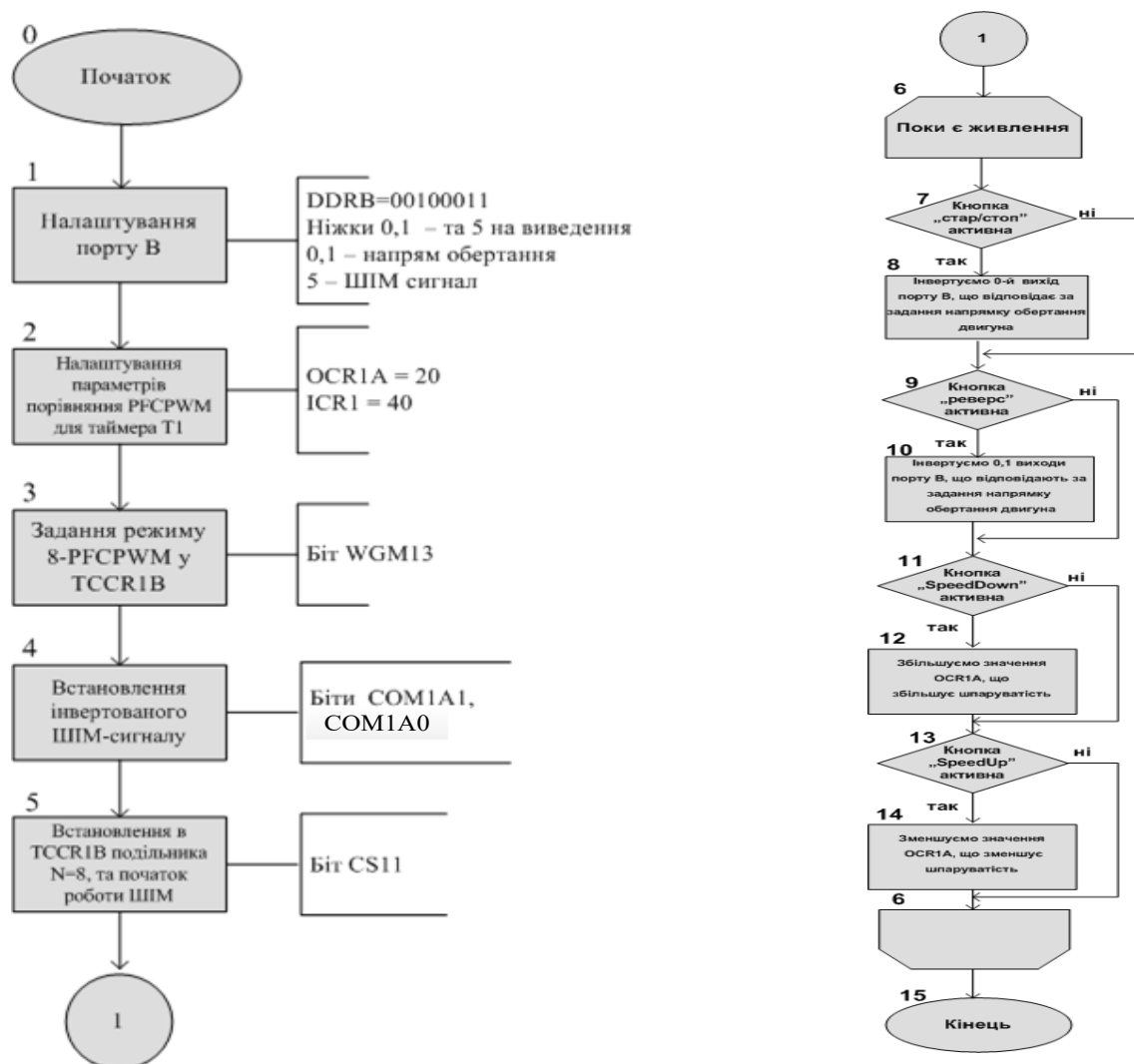


Рисунок 3.17 – Схема алгоритму роботи, що відповідає завданню

3.4.2 Особливості схем алгоритмів, які відповідають вихідним даним на виконання проекту згідно з відповідним варіантом

В завданнях інших варіантів існують суттєві відмінності, які впливають на особливості виконання роботи:

- замість таймера T/C1 використовується таймер – T/C3;
- ШІМ-сигнал знімається не з виводу з виводу PB5 порту В, а з виводу PE3 порту Е;

- замість режиму Phase and Frequency Correct PWM («З корекцією фази та частоти») використовуються режими: Phase Correct PWM («З корекцією фази»), або Fast PWM («Швидкодіючий ШІМ»).

Окрім того є відмінності у числових даних для розрахунків. Ці відмінності треба відобразити у схемі алгоритму роботи.

3.5 Розробка окремих фрагментів програми мовою Асемблер

3.5.1 Зупинка таймера

Для зупинки таймера треба записати в регістр керування TCCR1B (рисунок 3.18), адреса якого згідно із таблицею 3.2 дорівнює \$004E, наведене нижче керуюче слово KC1 = 00000000B = \$00:

7р.	6р.	5р.	4р.	3р.	2р.	1р.	0р.
ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10
0	0	0	0	0	0	0	0 (\$00 = KC1)

	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10	Все моделі
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
а)									
	7	6	5	4	3	2	1	0	
	ICNC3	ICES3	—	WGM33	WGM32	CS32	CS31	CS30	ATmega64x/128x ATmega162x ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
б)									
	7	6	5	4	3	2	1	0	
	ICNC4	ICES4	—	WGM43	WGM42	CS42	CS41	CS40	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
в)									
	7	6	5	4	3	2	1	0	
	ICNC5	ICES5	—	WGM53	WGM52	CS52	CS51	CS50	ATmega640x/1280x/1281x ATmega2560x/2561x
Читання (R)/Запис (W)	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Початкове значення	0	0	0	0	0	0	0	0	
г)									

Рисунок 3.18 — Формат регістрів TCCR1B (а), TCCR3B (б), TCCR4B (в), TCCR5B (г)

Тоді програма зупинки таймера має вигляд:

```
LDI R18, $00; R18 ← KC1 = $00;
LDI R27, $00; R27 ← $00
LDI R26, $4E; R26 ← $4E } X(R27, R26) ← $004E;
ST X, R18; TCCR1B ← R18 = $00, зупинка таймера.
```

3.5.2 Завантаження регістра TCCR1A

Формат регістра TCCR1A наведено на рисунку 3.19.

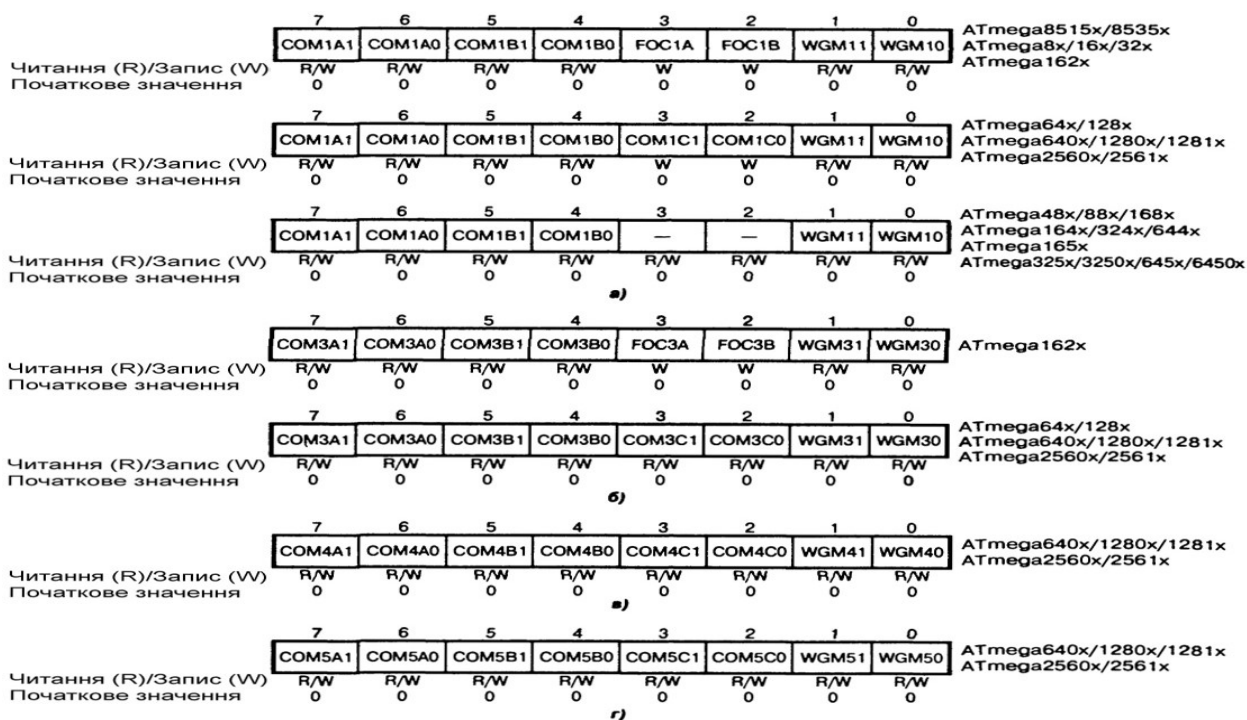


Рисунок 3.19 – Формат регістрів TCCR1A (а), TCCR3A (б), TCCR4A (в), TCCR5A (г)

Для нашої задачі для формування інвертованого ШІМ-сигналу згідно таблиці 3.3 необхідно встановити біти $COM1A1 = COM1A0 = 1$.

Для програмування режиму роботи 8 згідно таблиці 3.1 необхідно запрограмувати біти: $WGM11 = 0$; $WGM10 = 0$.

Інші біти регістра TCCR1A у нашому прикладі не використовуються. Тому запишемо в них нулі.

Таблиця 3.3 – Поведінка виводу OCnA/OCnB/OCnC в режимі Phase and Frequency Correct PWM

COMnx1	COMnx0	Опис
0	0	Таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC
0	1	WGMn3 = «0»: таймер/лічильник Tn відключено від виводу OCnA/OCnB/OCnC; WGMn3 = «1»: стан виводу OCnA змінюється на протилежний
1	0	Скидається в "0" при прямій лічбі і встановлюється в "1" при зворотній лічбі (неінвертований ШІМ-сигнал)
1	1	Встановлюється в "1" при прямій лічбі і скидається в "0" при зворотній лічбі (інвертований ШІМ-сигнал)

Примітка. $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Згідно рисунку 3.19, а формат регістра керуючого слова (KC2), яке завантажується у регістр TCCR1A МК-ра AT Mega 128 має вигляд:

7р.	6р.	5р.	4р.	3р.	2р.	1р.	0р.	KC2
COM1A1	COM1A0	COB1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	
1	1	0	0	0	0	0	0 B=\$C0	

Згідно таблиці 3.2 адреса регістра TCCR1A = \$004F.

Тоді програма завантаження регістра TCCR1A має вигляд:

```
LDI R17, $C0; R17← KC2 = $C0;
LDI R29, $00; R27 ← $00
LDI R28, $4F; $26 ← $4F } Y(R29, R28) ← $004F;
ST Y, R17; TCCR1A←R17 = $C0.
```

3.5.3 Програмування ліній PB5, PB1, PB0 на виведення

Для програмування ліній PB5, PB1, PB0 на виведення необхідно встановити 5-й, 1-й та 0-й розряди регістра DDRB в одиницю. Інші розряди регістра залишити без змін. Згідно таблиці 3.4 адреса регістра: \$0037.

Таблиця 3.4 – Адреси регістрів портів введення/виведення

Порт	Регістр	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x	ATmega162x	ATmega64x, ATmega128x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega165x, ATmega325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x, ATmega1280x/2560x
A	PORTA	\$1B (\$3B)	—	\$1B (\$3B)			—	\$02 (\$22)					
	DDRA	\$1A (\$3A)	—	\$1A (\$3A)			—	\$01 (\$21)					
	PINA	\$19 (\$39)	—	\$19 (\$39)			—	\$00 (\$20)					
B	PORTB	\$18 (\$38)					\$05 (\$25)						
	DDRB	\$17 (\$37)					\$04 (\$24)						
	PINB	\$16 (\$36)					\$03 (\$23)						
C	PORTC	\$15 (\$35)					\$08 (\$28)						
	DDRC	\$14 (\$34)					\$07 (\$27)						
	PINC	\$13 (\$33)					\$06 (\$26)						
D	PORTD	\$12 (\$32)					\$0B (\$2B)						
	DDRD	\$11 (\$31)					\$0A (\$2A)						
	PIND	\$10 (\$30)					\$09 (\$29)						
E	PORTE	\$07 (\$27)	—	—	\$07 (\$27)	\$03 (\$23)	—	—	\$0E (\$2E)				
	DDRE	\$06 (\$26)	—	—	\$06 (\$26)	\$02 (\$22)	—	—	\$0D (\$2D)				
	PINE	\$05 (\$25)	—	—	\$05 (\$25)	\$01 (\$21)	—	—	\$0C (\$2C)				

Порт	Регістр	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x	ATmega162x	ATmega64x, ATmega128x	ATmega48x/88x/168x	ATmega164x/324x/644x	ATmega165x, ATmega325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x, ATmega1280x/2560x
F	PORTF	\$03 (\$23)		—	—	—	—	(\$62)	—	—			
	DDRF	\$02 (\$22)		—	—	—	—	(\$61)	—	—			
	PINF	\$01 (\$21)		—	—	—	—	\$00 (\$20)	—	—			
G	PORTG	—	—	—	—	—	(\$65)	—	—	\$14 (\$34)			
	DDRG	—	—	—	—	—	(\$64)	—	—	\$13 (\$33)			
	PING	—	—	—	—	—	(\$63)	—	—	\$12 (\$32)			
H	PORTH	—	—	—	—	—	—	—	—	—	(\$DA)	—	(\$102)
	DDRH	—	—	—	—	—	—	—	—	—	(\$D9)	—	(\$101)
	PINH	—	—	—	—	—	—	—	—	—	(\$D8)	—	(\$100)
J	PORTJ	—	—	—	—	—	—	—	—	—	(\$DD)	—	(\$105)
	DDRJ	—	—	—	—	—	—	—	—	—	(\$DC)	—	(\$104)
	PINJ	—	—	—	—	—	—	—	—	—	(\$DB)	—	(\$103)
K	PORTK	—	—	—	—	—	—	—	—	—	—	—	(\$108)
	DDRK	—	—	—	—	—	—	—	—	—	—	—	(\$107)
	PINK	—	—	—	—	—	—	—	—	—	—	—	(\$106)
L	PORTL	—	—	—	—	—	—	—	—	—	—	—	(\$10B)
	DDRL	—	—	—	—	—	—	—	—	—	—	—	(\$10A)
	PINL	—	—	—	—	—	—	—	—	—	—	—	(\$109)

Тоді програма налаштування ліній PB5, PB1, PB0 на виведення має вигляд:

LDI R31, \$00; } Z(R31, R30) ← \$0037

LDI R30, \$37; R30 ← \$37 ;

LD R19, Z; R19← DDRB

ORI R19, \$23; R19←00100011B ∨ R19; R19.5←1, R19.1←1, R19.0←1,

інші біти без змін;

ST Z, R19; DDRB $\leftarrow$ R19; DDRB.5 $\leftarrow$ 1, DDRB.1 $\leftarrow$ 1, DDRB.0 $\leftarrow$ 1.

3.5.4 Завантаження регістра ICR1

Згідно таблиці 3.2 16-розрядний регістр ICR1 має адресу:

\$0047 (СБ) : \$0046 (МБ).

В ICR1 треба завантажити: TOP = 40 = 00101000B = \$0028 (підрозд. 3.3).

Тоді програма завантаження регістра ICR1 має вигляд:

```
LDI R23, $00; R23 $\leftarrow$  $00;
LDI R27, $00; R27  $\leftarrow$  $00;
LDI R26, $47; R26  $\leftarrow$  $47 } X(R27, R26)  $\leftarrow$  $0047;
ST X, R23; СБ ICR1 $\leftarrow$ R23 = $00.

LDI R24, $28; R24 $\leftarrow$  $28
LDI R29, $00; R29  $\leftarrow$  $00
LDI R28, $46; R28  $\leftarrow$  $46 } Y(R29, R28)  $\leftarrow$  $0046;
ST Y, R24; МБ ICR1 $\leftarrow$ R24 = $28.
```

3.5.5 Завантаження регістра OCR1A

Згідно таблиці 3.2, 16-розрядний регістр OCR1A має адресу:

\$004B (СБ): \$004A (МБ).

В OCR1A треба завантажити:

20 = 00010100B = \$0014 (див. вище).

Тоді програма завантаження регістра OCR1A має вигляд:

```
LDI R24, $00; R24 $\leftarrow$  $00;
LDI R31, $00; R31  $\leftarrow$  $00;
LDI R30, $4B; R30  $\leftarrow$  $4B } Z(R31, R30)  $\leftarrow$  $004B;
ST Z, R24; СБ OCR1A $\leftarrow$ R24 = $00.

LDI R25, $14; R25  $\leftarrow$  $14;
```

```
LDI R31, $00; R31 ← $00;  
LDI R30, $4A; R30 ← $4A } Z(R31,R30) ← $004A;  
ST Z, R25; МБ OCR1A ← R25 = $14.
```

3.5.6 Програмування $K_{\text{д\ddot{u}л}} = N = 8$, режиму роботи номер 8 та запуск таймера Т/С1

Вище при завантаженні регістра TCCR1A було записано WGM11 = 0 та WGN10 = 0, що разом з двома бітами WGM12 = 0 та WGM13 = 1 регістра TCCRB програмують режим роботи № 8 (таблиця 3.1).

Для програмування Кділ = 8 (таблиця 3.5) та запуску таймера також використовують регістр TCCR1B (рисунок 3.2), в якій треба завантажити керуюче слово: КС3 = \$12:

7 p.	6p.	5p.	4p.	3p.	2p.	1p.	0p.
ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
0	0	0	1	0	0	1	0 (КСЗ = 12)

(режим 8) (Кділ = 8)

Таблиця 3.5 – Вибір джерела тактового сигналу таймерів/лічильників Tn

CSn2	CSn1	CSn0	Джерело тактового сигналу	
			ТЗ в моделях ATmega162x	Інші
0	0	0	Таймер/лічильник	Таймер/лічильник зупинений
0	0	1	$f_{clkI/0}$	$f_{clkI/0}$
0	1	0	$f_{clkI/0}/8$	$f_{clkI/0}/8$
0	1	1	$f_{clkI/0}/64$	$f_{clkI/0}/64$
1	0	0	$f_{clkI/0}/256$	$f_{clkI/0}/256$
1	0	1	$f_{clkI/0}/1024$	$f_{clkI/0}/1024$
1	1	0	$f_{clkI/0}/16$	Вивід Тп, лічба виконується за спадаючим фронтом

1	1	1	$f_{\text{clkI/O}}/32$	Вивід T_n , лічба виконується за наростаючим фронтом
---	---	---	------------------------	--

Примітка. $n = 1, 3, 4$ або 5 .

Адреса регістра TCCR1B згідно таблиці 3.2 дорівнює \$004E.

Тоді програма програмування Кділ = N = 8, режиму роботи номер 8 та запуск таймера T/C1 має вигляд:

```
LDI R17, $12; R17 ← $12;
LDI R29, $00; R29 ← $00;
LDI R28, $4E; R28 ← $4E;
ST Y, R17; TCCR1B ← R17 = $12.
```

3.6 Керуюча програма мовою C

3.6.1 Керуюча програма мовою C, яка відповідає завданню

```
1.  #include <inttypes.h>
2.  #include <avr/io.h>
3.  #include <avr/interrupt.h>
4.  #include <avr/sleep.h>
5.  #include <util/delay.h>
6.  #define F_CPU 16000000L
7.  int main(void) // 0
8.  {
9.  // 1: Init port B
10. DDRB = 0b00100011;
11. PORTB = 0b00000000;
12. // 2: Init 16-bit timer 1 values
13. OCR1A = 20;
14. ICR1 = 40;
```

```

15. int delay = 1; //змінна delay, яка використовується при зміні
    //шпаруватості
16. // Init Phase and frequency correct PWM
17. TCCR1B = _BV(WGM13); // 3
18. TCCR1A = 0; // 4
19. TCCR1A |= _BV(COM1A0); // 4
20. TCCR1A |= _BV(COM1A1);
21. TCCR1B |= _BV(CS11); // 5
22. //об'явлення змінних-прапорців
23. char flag_start_stop=0,flag_reverse=0,flag_speedup=0,flag_speeddown=0;
24. while(1) // 6
25. {
26. // 7: Start–Stop button action
27. if(!(PINB&4))
28. { flag_start_stop = 1; _delay_ms(10); }
29. if(( flag_start_stop==1 )&&(PINB&4))
30. {PORTB^=1; flag_start_stop = 0; } // 8
31. // 9: Reverse button action
32. if(!(PINB&8))
33. { flag_reverse = 1; _delay_ms(10); }
34. if(( flag_reverse==1 )&&(PINB&8))
35. {PORTB^=3; flag_reverse = 0; } // 10
36. // 11: Speed – button action
37. if(!(PINB&64))
38. { flag_speeddown=1; _delay_ms(10); }
39. if(( flag_speeddown==1 )&&(PINB&64))
40. { if (OCR1A!=40) OCR1A+=delay; flag_speeddown = 0; } // 12
41. // 13: Speed + button action
42. if(!(PINB&16))

```

```

43. { flag_speedup = 1; _delay_ms(10); }
44. if(( flag_speedup==1 )&&(PINB&16))
45. {
46. if (OCR1A!=0)OCR1A-=delay; // 14
47. flag_speedup = 0;
48. }
49. } // 6: End while
50. } // 15: End of proram

```

3.6.2 Компіляція робочої програми, отримання hex-файлу та його завантаження у пам'ять комп'ютера

У [3] описано послідовність створення робочого проекту в Atmel Studio, компіляцію програми та створення hex-файлу.

У [3] також описано послідовність створення hex-файлу та його завантаження в пам'ять мікроконтролера.

3.6.3 Особливості керуючої програми, яка відповідає вихідним даним на виконання роботи згідно з відповідним варіантом

На рисунку 3.17 наведено схему алгоритму роботи пристрою, який відповідає завданню (підрозд. 3.4).

В завданнях інших варіантів існують суттєві відмінності, які впливають на особливості виконання роботи:

- замість таймера T/C1 використовується таймер – T/C3;
- ШІМ-сигнал знімається не з виводу з виводу PB5 порту В, а з виводу PE3 порту Е;

– замість режиму Phase and Frequency Correct PWM («З корекцією фази та частоти») використовуються режими: Phase Correct PWM («З корекцією фази»), або Fast PWM («Швидкодіючий ШІМ»).

Окрім того є відмінності у числових даних для розрахунків. Ці відмінності треба відобразити у керуючій програмі.

Для цього нижче наведено деякі додаткові теоретичні відомості.

3.6.3.1 Режим Fast PWM («Швидкодіючий ШІМ»)

Режим Fast PWM («Швидкодіючий ШІМ») дозволяє генерувати високочастотний сигнал із широтно-імпульсною модуляцією.

Лічильний регістр у цьому режимі функціонує як лічильник, що підсумовує, інкремент якого здійснюється кожним імпульсом тактового сигналу f_{clkTn} . Стан лічильника змінюється від 0000 до максимального значення, після чого лічильний регістр скидається і цикл повторюється. Максимальне значення лічильника (роздільна здатність ШІМ-сигналу) є або фіксованим значенням, або визначається вмістом окремих розрядів керуючих регістрів таймерів/лічильників (таблиця 3.1).

Рекомендується обирати режим 14, тоді модуль лічби (TOP) визначається регістром ICR1, або ICR3 (залежить від номеру таймера в завданні).

Часові діаграми для випадку, коли модуль лічби визначається вмістом регістра ICRn, показано на рисунку 3.20.

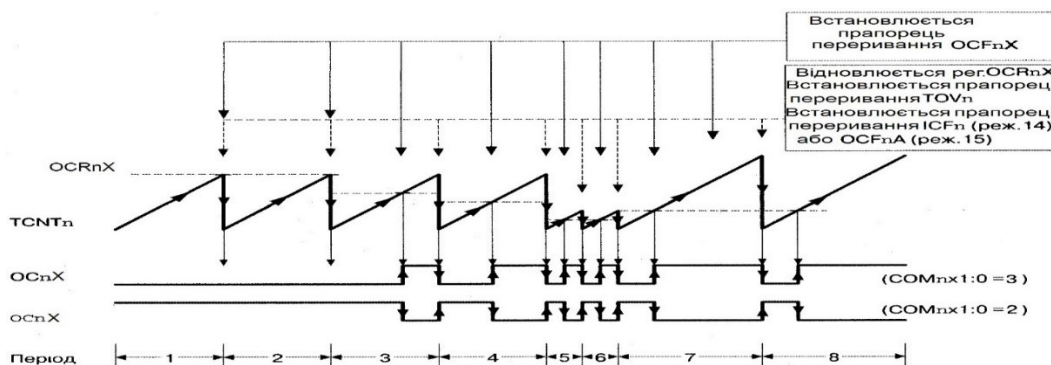


Рисунок 3.20 – Формування ШІМ-сигналу в режимі Fast PWM

Частота сигналу, тривалість імпульсу та період сигналу, які генеруються, визначаються виразами:

$$f_{OCnx} = f_{clkI/0} / N \cdot (1 + TOP), \quad (3.6)$$

$$t_{imp} = T_{clkI/0} \cdot N \cdot (OCRnx + 1), \text{ неінвертований ШІМ-сигнал} \quad (3.7)$$

$$t_{imp} = T_{OCnx} - T_{clkI/0} \cdot N \cdot (OCRnx + 1), \text{ інвертований ШІМ-сигнал} \quad (3.8)$$

$$T_{OCnx} = T_{clkI/0} \cdot N \cdot (1 + TOP), \quad (3.9)$$

де N – коефіцієнт ділення попереднього дільника; TOP – модуль лічби; $OCRnx$ – вміст регістра $OCRnx$; $f_{clkI/0}$, $T_{clkI/0}$ – відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення; $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Значення: $ICR1 = TOP$, періоду ШІМ-сигналу, тривалості імпульсу розраховуємо за $N = 8$ згідно прикладу, наведеному вище у підрозд. 3.3.

Згідно з рисунком 3.20 для *інвертованого* ШІМ-сигналу $t_{imp} = T_{OCnx} - \Delta t$.

Звідки $\Delta t = T_{OCnx} - t_{imp}$.

Згідно роботи таймера у режимі Fast PWM

$$\Delta t = T_{CLK I/0} \cdot N \cdot (OCnx + 1) \quad (3.10)$$

Відповідно

$$OCnx + 1 = \Delta t / (T_{CLK I/0} \cdot N). \quad (3.11)$$

Останній вираз дозволяє розрахувати значення $OCnx$.

Для *неінвертованого* ШІМ-сигналу значення t_{imp} визначається виразом (3.7), з якого можна розрахувати значення $OCnx$.

3.6.3.2 Режим Phase Correct PWM («ШІМ з корекцією фази»)

Режим Phase Corect PWM («ШІМ з корекцією фази»), як і режим Fast PWM призначено для генерації сигналів із широтно-імпульсною модуляцією. Однак у цьому режимі лічильний регістр функціонує як реверсивний лічильник, стан якого спочатку змінюється від \$0000 до максимального значення, а потім назад до \$0000.

Відповідно максимальна частота сигналу у цьому режимі в два рази менше максимальної частоти сигналу у режимі Fast PWM.

Максимальне значення лічильника (роздільна здатність ШІМ-сигналу) є або фіксованим значенням, або визначається вмістом окремих розрядів керуючих регістрів таймерів/лічильників (таблиця 3.1).

Рекомендується обирати режим 10, тоді модуль лічби (TOP) визначається регістром ICR1, або ICR3 (залежить від номеру таймера в завданні).

Часові діаграми в режимі Phase Correct PWM, показано на рисунку 3.21.

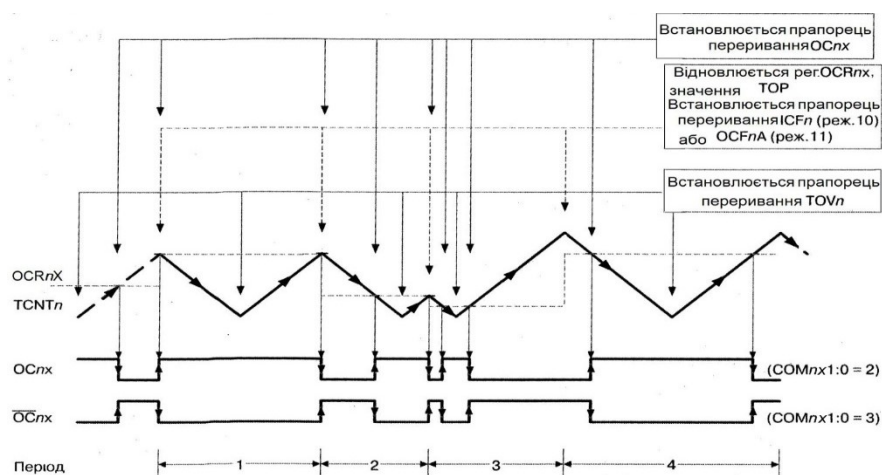


Рисунок 3.21 – Формування ШІМ-сигналу в режимі Phase Correct PWM

Частота, тривалість імпульсу та період сигналу, що генерується, визначаються виразами:

$$f_{OCnx} = f_{clkI/0} / (2 \cdot N \cdot TOP), \quad (3.12)$$

$$t_{imp} = T_{clkI/0} \cdot 2 \cdot N \cdot OCR_{nx}, \text{ неінвертований ШІМ-сигнал} \quad (3.13)$$

$$t_{imp} = T_{OCn} - (T_{clkI/0} \cdot 2 \cdot N \cdot OCR_{nx}), \text{ інвертований ШІМ-сигнал} \quad (3.14)$$

$$T_{OCnx} = T_{clkI/0} \cdot 2 \cdot N \cdot TOP, \quad (3.15)$$

де N – коефіцієнт ділення попереднього дільника; TOP – модуль лічби; OCR_{nx} – вміст регістра OCR_{nx} ; $f_{clkI/0}$, $T_{clkI/0}$ – відповідно частота та період генератора тактових імпульсів підсистеми введення/виведення; $n = 1, 3, 4$ або 5 ; $x = A, B$ або C .

Згідно з рисунком 3.21 для *інвертованого* ШІМ-сигналу – $t_{\text{ІМП}} = T_{\text{OCnx}} - 2\Delta t$.

Звідки

$$\Delta t = (T_{\text{OCnx}} - t_{\text{ІМП}})/2. \quad (3.16)$$

Згідно роботи таймера у режимі Phase Correct PWM

$$\Delta t = T_{\text{CLK I/O}} \cdot N \cdot \text{OCnx}. \quad (3.17)$$

Відповідно

$$\text{OCnx} = \Delta t / (T_{\text{CLK I/O}} \cdot N). \quad (3.18)$$

Останній вираз дозволяє розрахувати значення OCnx.

Для *неінвертованого* ШІМ-сигналу значення $t_{\text{ІМП}}$ визначається виразом (3.13), з якого можна розрахувати значення OCnx.

3.6.3.3 Режим Phase and Frequency Correct PWM («ШІМ з корекцією фази та частоти»)

Розрахунок ШІМ-модуля та програмування його мовою Асемблер розглянуто вище у підрозд. 3.3.

3.7 Моделювання пристрою в пакеті PROTEUS

3.7.1 Моделювання пристрою, який розроблено згідно завданню на КП

Для початку моделювання необхідно запустити модель на виконання (рисунок 3.22).

На екран осцилографа виводиться послідовність прямокутних імпульсів зі шпаруватістю 2, що відповідає наведеному вище розрахунку. Цей сигнал формується першим таймером мікроконтролера та з лінії PB5 поступає на вхід EN1 драйвера (U2). Двигун не обертається, тому що лінії IN1 та IN2 драйвера з виходів PB0 та PB2 подаються низькі рівні сигналів – логічні нулі.

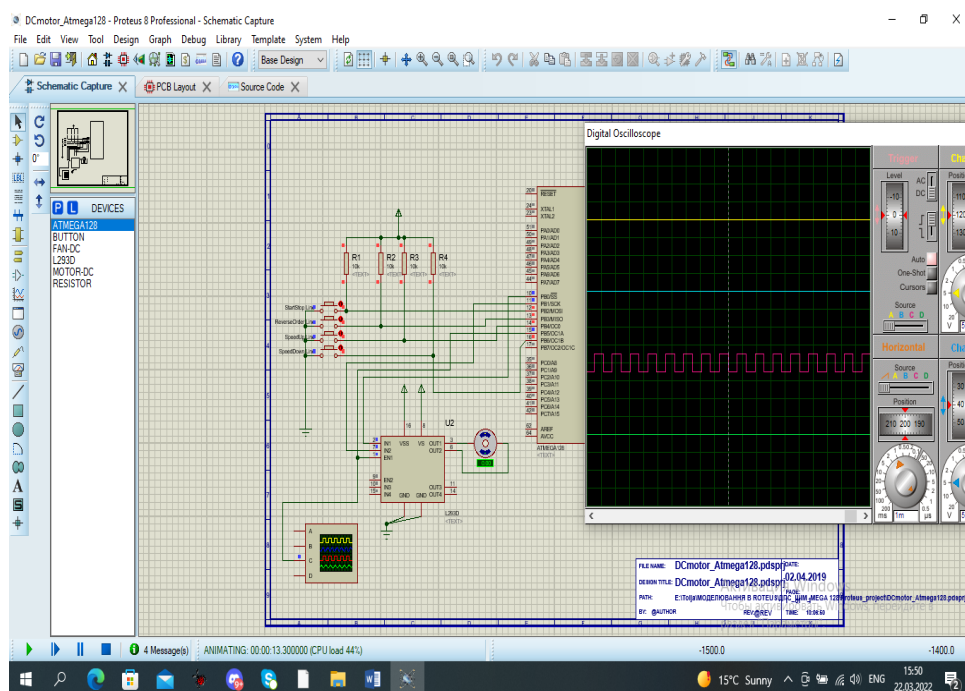


Рисунок 3.22 – Стан моделі після початку моделювання

Для запуску двигуна короткочасно натискаємо та відпускаємо кнопку StartStopLine. Двигун починає обертатися за годинниковою стрілкою (рисунок 3.23), тому що на лінію IN1 від мікроконтролера подається логічна одиниця, а на лінію IN2 – логічний нуль.

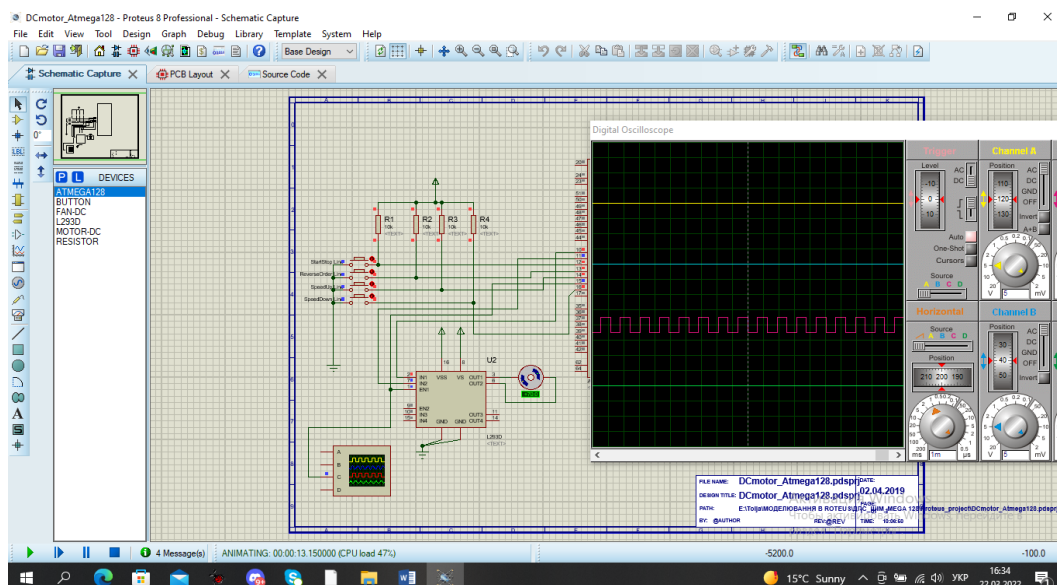


Рисунок 3.23 – Стан моделі після натискання та відпускання кнопки StartStopLine

Для зупинки двигуна треба ще раз натиснути та відпустити кнопку StartStopLine. Через інерційність двигуна він зупиняється не миттєво.

Для зміни напрямку обертання двигуна натискаємо та відпускаємо кнопку ReverseOrderLine. Дигун починає обертатися проти годинникової стрілки (рисунок 3.24), тому що на лінію IN1 від мікроконтролера подається логічний нуль, а на лінію IN2 – логічний одиниця.

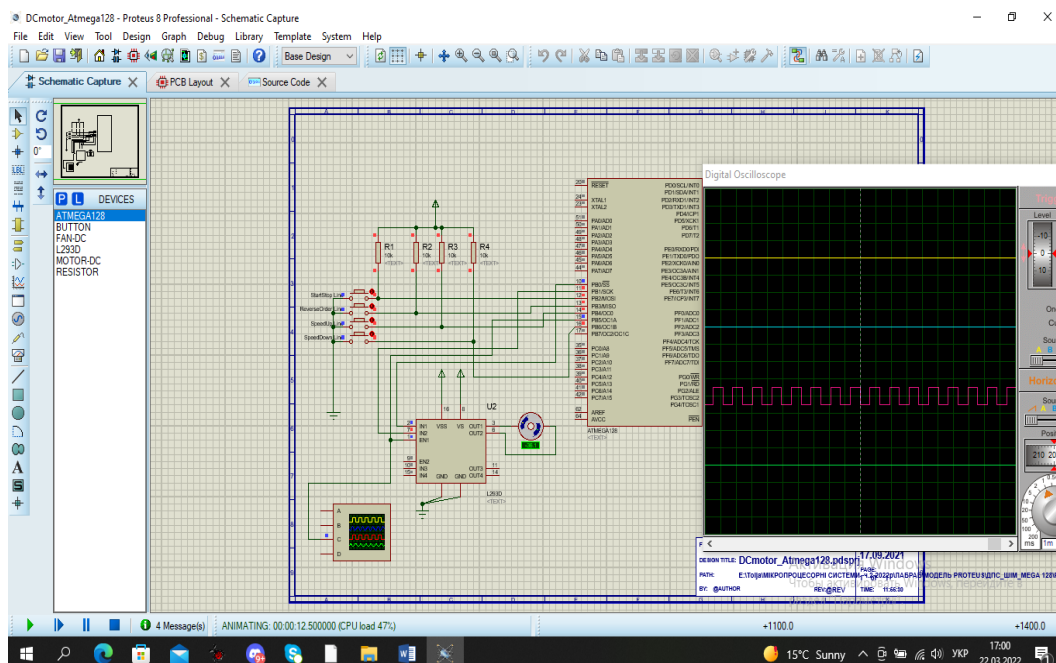


Рисунок 3.24 – Стан моделі після натискання та відпускання кнопки ReverseOrderLine

Для збільшення швидкості обертання двигуна треба декілька разів короткочасно натискати та відпускати кнопку SpeedUpLine. Шпаруватість керуючих імпульсів зменшується (рисунок 3.25), тому двигун починає обертатися швидше.

Для зменшення швидкості обертання двигуна треба декілька разів короткочасно натискати та відпускати кнопку SpeedDownLine. Шпаруватість керуючих імпульсів збільшується (рисунок 3.26), тому двигун починає обертатися повільніше.

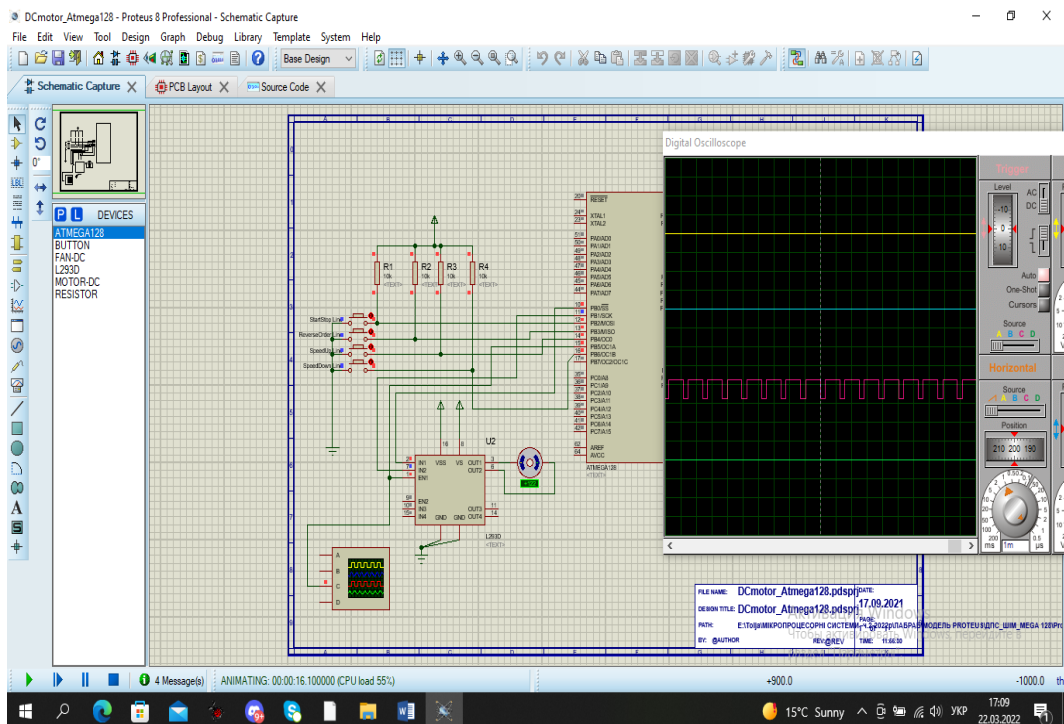


Рисунок 3.25 — Стан моделі після декількох натискань та відпускань кнопки SpeedUpLine

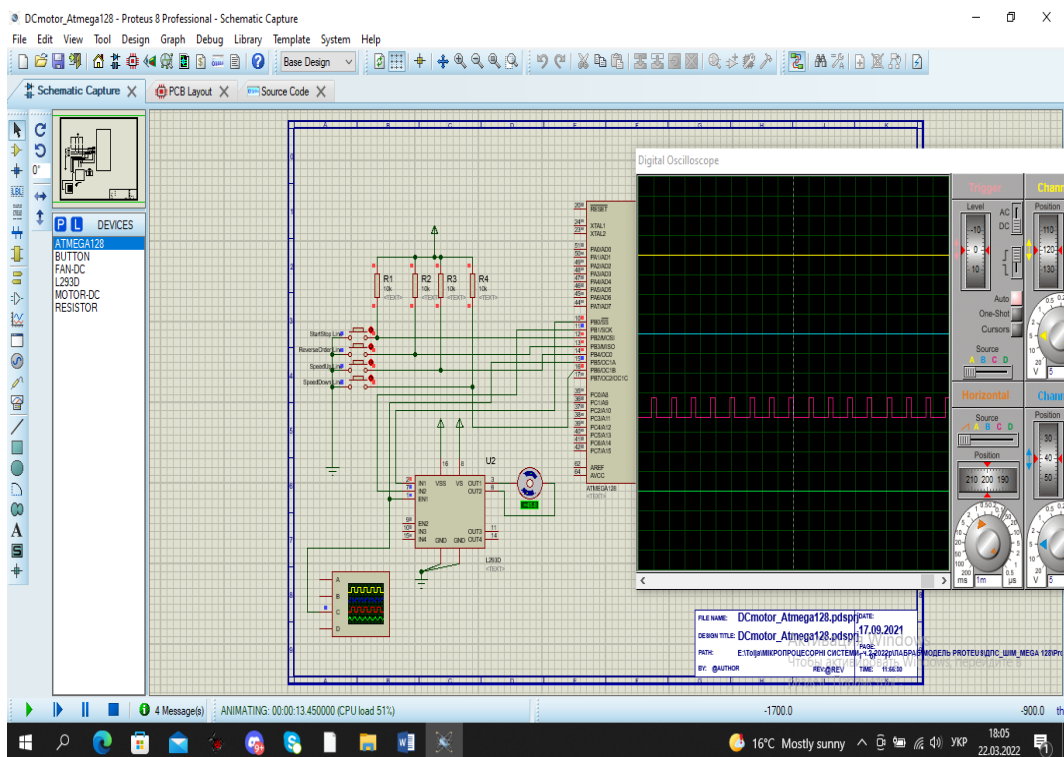


Рисунок 3.26 — Стан моделі після декількох натискань та відпускань кнопки SpeedDownLine

3.7.2 Моделювання пристрою, який відповідає вихідним даним на виконання роботи згідно з відповідним варіантом завдання на КП

Нижче наведено схему моделі пристрою, який відповідає вихідним даним на виконання роботи згідно з відповідним варіантом завдання на КП (рисунок 3.27).

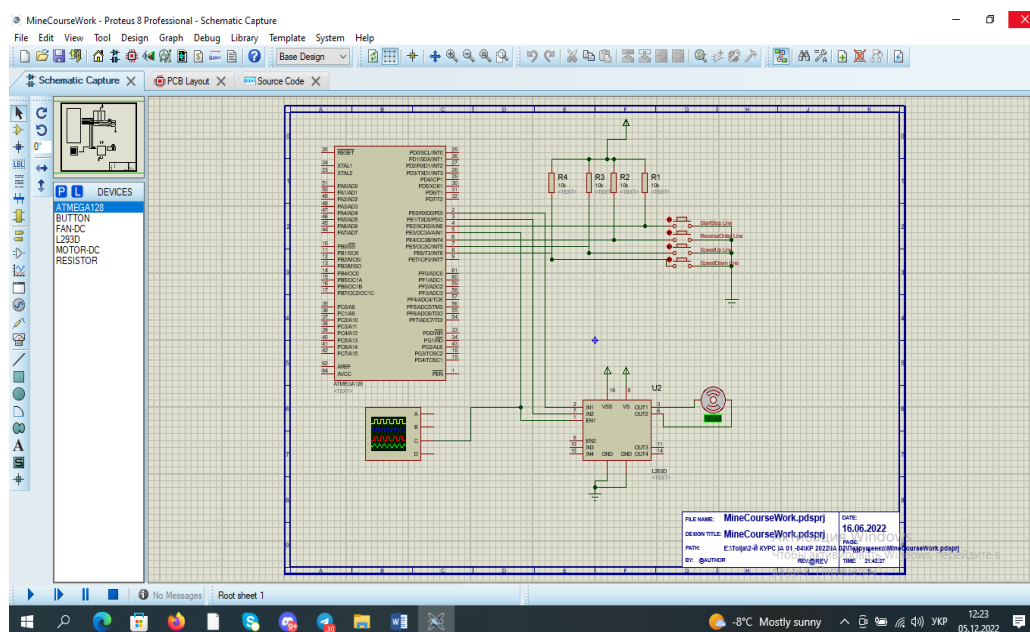


Рисунок 3.27 – Схема робочої моделі пристрою, який розроблено згідно відповідного варіанту завдання на КП

Ця модель відповідає пристрою, в якому використовується: мікроконтролер AT mega 128, таймер – T/C3, лінія порту мікроконтролера, на якій формується ШІМ-сигнал, – PE3; шпаруватість ШІМ-сигналу – $Q_{\text{шім}} = 4$; режим роботи модуля ШІМ – Phase Correct PWM, частота вихідного ШІМ-сигналу – $f_{\text{шім}} = 15 \text{ кГц}$, частота тактового сигналу підсистеми введення/виведення: $f_{\text{CLK I/O}} = 16 \text{ МГц}$, вид вихідного ШІМ-сигналу – неінвертований.

Нижче наведено стан моделі після запуску (рисунок 3.28).

З рисунку видно, що пристрій формує ШІМ-сигнал зі шпаруватістю – $Q_{\text{шім}} = 4$, що відповідає завданню.

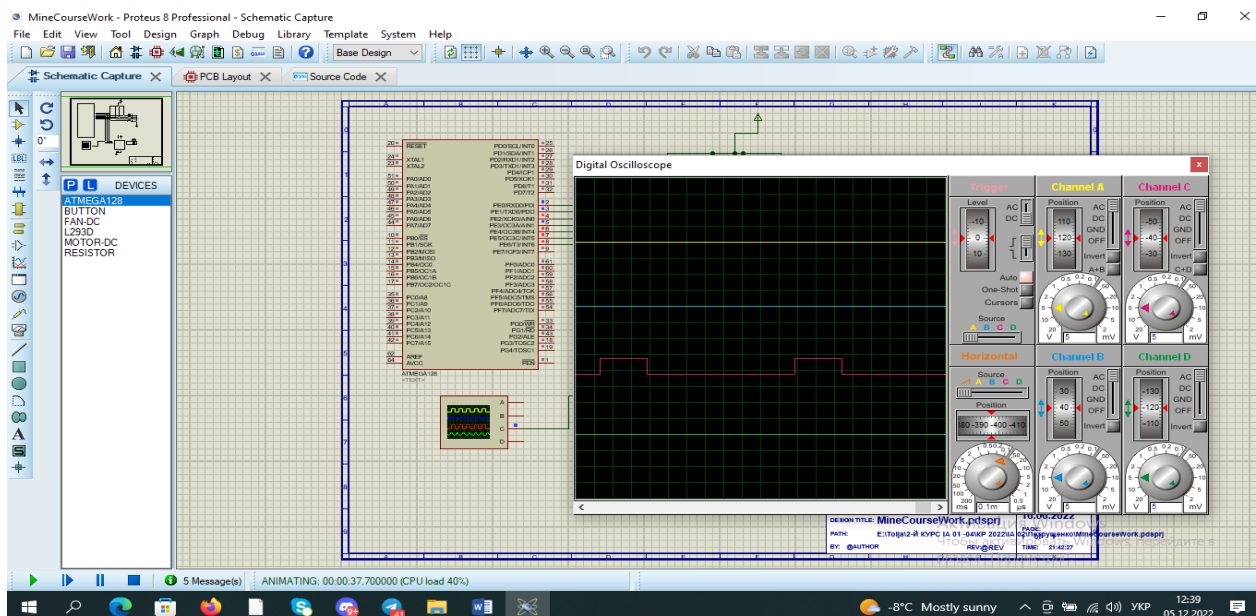


Рисунок 3.28 — Стан моделі після запуску

3.8 Розробка та опис структурної схеми пристрою

Нижче на рисунку 3.29 наведено схему електричну структурну пристрою керування двигуном постійного струму (ДПС).

Основним вузлом структури є мікроконтролер, який через вхідний паралельний порт отримує сигнали керування двигуном від чотирьох кнопок:

- запуск/зупинка двигуна;
- зміна напрямку обертання двигуна;
- збільшення швидкості обертання;
- зменшення швидкості обертання.

Через вихідний паралельний порт МК видає сигнали керування, які запускають чи зупиняють двигун та змінюють напрямок його обертання.

Для зміни швидкості обертання двигун використовується модуль таймера, який формує імпульсний ШІМ-сигнал зі змінною шпаруватістю.

Для збільшення потужності сигналу, який подається в обмотку збудження двигуна використовується драйвер ШІМ (ДРШІМ).

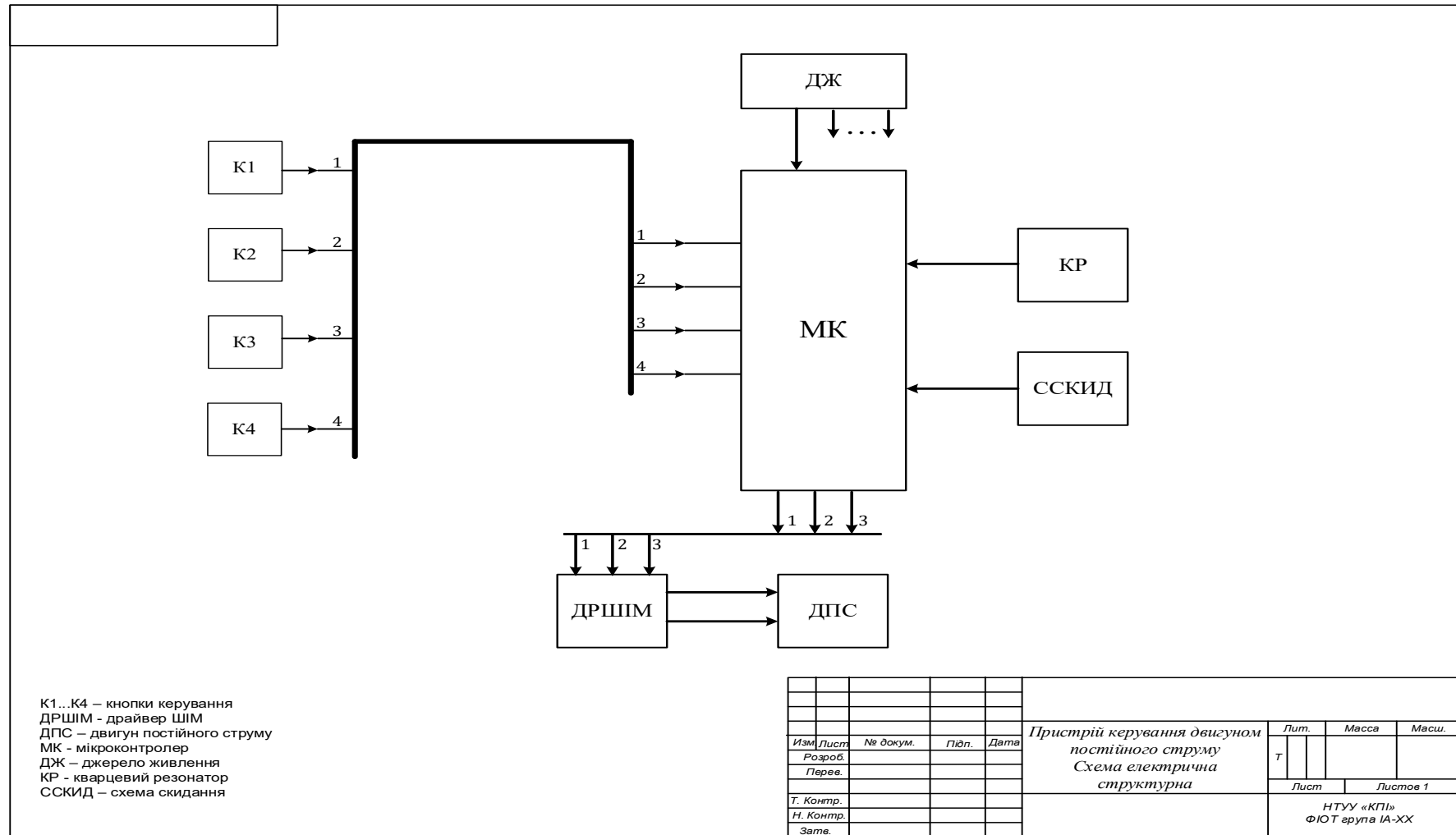


Рисунок 3.29 – Схема електрична структурна пристрою

Кварцовий резонатор (КР) визначає частоту високочастотного генератора, схема якого вбудована у МК.

Схема скидання (ССКИД) формує сигнал «RESET» (Скидання), який запускає виконання робочої програми спочатку. Цей сигнал формується автоматично при включенні живлення, або від зовнішньої кнопки.

3.9 Розробка текстових документів до роботи

Пояснювальна записка та інші текстові документи до курсового проєкту розробляються Word згідно переліку, який наведено у підрозділі 1.3 методички до виконання КП.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Базова

1. Мікропроцесорні та мікроконтролерні системи : підручник. У 2 ч. Ч. 1. Мікропроцесорні системи [Електронний ресурс] / А. О. Новацький. – Електронні текстові дані (1 файл: 43,8 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019.
2. Електроніка та мікропроцесорна техніка : підручник. У 2 ч. Ч. 2. Мікропроцесорні системи [Електронний ресурс]/А. О. Новацький. – Електронні текстові дані (1 файл: 20,2 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2023.
3. Мікропроцесорні та мікроконтролерні системи: Частина 2. Проектування мікропроцесорних систем: Лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. освітньої програми «Інтегровані інформаційні системи» спеціальності 126 «Інформаційні системи та технології» / А.О. Новацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 22,38 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021.
4. Проектування вбудованих систем: Лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. освітньої програми «Інтегровані інформаційні системи» спеціальності 126 «Інформаційні системи та технології» / А.О. Новацький, В.М. Шимкович; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 34,22 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022.
5. Комп'ютерна електроніка [Електронний ресурс] : підручник для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізації «Інтегровані інформаційні системи» / А.О. Новацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 80.9 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018.

Допоміжна

6. Проектування та програмування мікропроцесорних систем і мереж: Проектування мережі 1–WIRE: Навчальний посібник для студентів спеціальностей 7.05020101, 8.05020101 «Комп'ютеризовані системи управління та автоматика» кафедри автоматики та управління в технічних системах / Автор: А.О. Новацький– К: НТУУ „КПІ”, 2014.
7. Навчальний посібник з дисципліни «Проектування мікропроцесорних систем та мереж», розділ «Проектування CAN-мережі» для студентів

спеціальності 8.050201.01 «Комп'ютеризовані системи управління та автоматика» кафедри Автоматики та управління у технічних системах / Автор: А.О. Новацький – К: НТУУ „КПІ”, 2016.

8. Сван Т. Освоение Turbo Assembler/Т. Сван. – 2-е изд. – Киев : СПб. : Диалектика, 1996.
9. Гилмор Ч. Введение в микропроцессорную технику / Пер. с англ. – М. : Энергоатомиздат, 1984.
10. Официальное описание микроконтроллеров XMEGA
http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/avr_xmega/start.htm

Додаток А

Перелік тем курсових проєктів для реальних пристроїв

1. Пристрій управління двигуном постійного струму.
2. Пристрій управління уніполярним кроковим двигуном.
3. Пристрій управління біполярним кроковим двигуном.
4. Пристрій відображення інформації із використанням 7 – сегментних світлодіодних індикаторів.
5. Пристрій відображення інформації із використанням 7 – сегментних рідкокристалічних індикаторів.
6. Пристрій відображення текстової інформації.
7. Пристрій контролю: температури повітря; вологості; атмосферного тиску; освітленості; CO та CO₂ у приміщенні; кута повороту валу; частоти обертання валу; швидкості та напрямлення вітру; температури та вологості ґрунту; пульсу людини; артеріального тиску людини; температури тіла людини; ЕКГ – людини.
8. Пристрій перетворення частота – аналог.
9. Кодовий замок.
10. Цифровий частотомір.
11. Цифровий вольтметр.
12. Годинник реального часу.
13. Цифровий генератор синусоїдальної напруги.
14. Пристрій охоронної сигналізації.
15. Пристрій для заряджання акумуляторів.
16. Пристрій керування пральною машиною.
17. Пристрій вимірювання температури та вологості.
18. Пристрій виведення інформації на рідкокристалічний дисплей.
19. Пристрій керування водозабірним вузлом.
20. Пристрій керування кабіною ліфта.

21. Пристрій керування підсистемою поверху ліфта.
22. Пристрій керування підсистемою силової частини ліфта.
23. Пристрій керування кондиціонером.
24. Пристрій введення інформації у мікроконтролер від цифрових давачів.
25. Пристрій зв'язку мікроконтролера з модемом.
26. Кодер коду Хемінга.
27. Декодер коду Хемінга.
28. Кодер циклічного коду.
29. Декодер циклічного коду.
30. Пристрій зв'язку за інфрачервоним каналом.
31. Пристрій протипожежної сигналізації.
32. Пристрій сигналізації перевищення рівня шуму у приміщенні.
33. Пристрій керування обладнанням житлового приміщення.
34. Пристрій керування мікрокліматом у теплиці.
35. Пристрій керування мікрокліматом у автомобілі.
36. Пристрій керування замками та склопідйомниками в автомобілі.
37. Пристрій керування подушками безпеки в автомобілі.
38. Пристрій контролю метеорологічних параметрів.
39. Пристрій керування фарами легкового автомобіля.
40. Пристрій керування рухом на перехресті.
41. Пристрій керування автозаправною станцією.
42. Пристрій керування колісним роботом.
43. Пристрій керування інкубатором.
44. Пристрій керування паровим котлом.
45. Пристрій керування мікрокліматом у виробничому приміщенні.
46. Пристрій керування НВЧ-піччю.
47. Пристрій керування випічкою печива.

Додаток Б

Приклад оформлення титульного аркуша

Міністерство освіти і науки України

Національний технічний університет України

„Київський політехнічний інститут ім. Ігоря Сікорського”

Кафедра інформаційних систем та технологій

Пристрій обміну інформацією

Курсовий проєкт

З дисципліни «ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА»

Керівник доц. Новацький А.О.

„Допущений до захисту”

Виконавець

ст. Петренко О.Е.

зал. книжка №

(Особистий підпис керівника)

« ____ » _____ 2026 р.

Захищений з оцінкою

(Особистий підпис виконавця)

« ____ » _____ 2026 р.

(оцінка)

« ____ » _____ 2026 р.

Члени комісії :

(Особистий підпис)

(Розшифровка підпису)

(Особистий підпис)

(Розшифровка підпису)

Київ 2026р.

Додаток В

Приклад оформлення завдання на курсовий проєкт

Національний технічний університет України

„Київський політехнічний інститут ім. Ігоря Сікорського”

Кафедра інформаційних систем та технологій

Дисципліна «Електроніка та мікропроцесорна техніка»

Курс 2 Група ІА-21 Семестр 4

ЗАВДАННЯ

на курсовий проєкт студента

Петренко Олега Едуардовича

(прізвище, ім'я, по батькові)

- 1.Тема проєкту Пристрій обміну інформацією
- 2.Термін здачі студентом закінченого проєкту 31.05.2026р.
- 3.Вихідні дані до проєкту Швидкість передачі: $V_{\text{пд}}=2400$ бод; тактова частота кварцового резонатора: $f_{\text{CLK}} = 8\text{МГц}$; для обміну використовувати модуль USART; режим передачі – асинхронний; розмір слова даних – 8 розрядів; використовувати перевірку на парність; кількість стоп-бітів – 1; байт для передачі знаходиться у регістрі R22 = 10110101b; символ для прийому від віртуального терміналу – A
- 4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) Призначення та область застосування пристрою, технічна характеристика, вибір та опис структурної схеми, розрахунки, які підтверджують працездатність пристрою, розробка схеми алгоритму роботи, керуючої програми та схеми моделювання
- 5.Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) схема електрична структурна
- 6.Дата видачі завдання 14.02.2026р.

Продовження додатку В

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсового проєкту	Термін виконання етапів роботи	Примітки
1	Отримання завдання на курсовий проєкт		
	та узгодження вихідних даних	25.02.2026р	
2	Огляд існуючих рішень та розробка		
	структури пристрою	15.03.2026р.	
3	Виконання основних розрахунків	06.04.2026р	
4	Розробка схеми алгоритму та керуючої програми	15.04.2026р	
5	Моделювання пристрою	27.04.2026р.	
6	Оформлення текстової документації	15.05.2026р.	
7	Представлення до захисту	31.05.2026р.	

Студент _____

(підпис)

Керівник _____

(підпис)

Анатолій НОВАЦЬКИЙ

(ім'я та прізвище)

«25» _____ 02 _____ 2026 р.

Додаток Г

Приклад оформлення змісту до ПЗ курсового проєкту

ЗМІСТ

ВСТУП	2
1 Призначення та область використання	3
2 Технічна характеристика пристрою	4
3 Огляд існуючих рішень, обґрунтування вибору структури та опис роботи пристрою за структурною схемою	6
3.1 Огляд існуючих рішень та обґрунтування вибору структури пристрою	8
3.2 Опис роботи пристрою за структурною схемою	10
4 Обґрунтування вибору та опис роботи окремих вузлів пристрою	13
4.1 Мікроконтролер	15
4.2 Пристрій вибірки-зберігання	16
4.3 Аналого-цифровий перетворювач	17
4.4 Цифро-аналоговий перетворювач	18
4.5 Фільтр нижніх частот	17
5 Розробка та опис принципової схеми	20
6 Розрахунки, що підтверджують дієздатність пристрою	20
6.1 Нормуючі перетворювачі	20
6.2 Фільтр нижніх частот	21
6.3 Пристрій вибірки-зберігання	22
6.4 Аналого-цифровий перетворювач	23
6.5 Цифро-аналоговий перетворювач	24
7 Розробка схеми алгоритму та керуючої програми	25
8 Розробка та опис роботи моделі пристрою	26

ВИСНОВОК	29
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
Додаток А Результати моделювання пристрою	31
Додаток Б Лістинг керуючої програми	34

Додаток Д

Приклад оформлення текстового документу

1 ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВОГО ПРОЄКТУ

1.1 Мета і завдання курсового проєкту

Мета курсового проєкту – закріпити теоретичні положення курсу «ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА», навчитися застосовувати отримані знання для практичних розрахунків, придбати навички проєктування окремих модулів мікропроцесорних систем. Виконання курсового проєкту є підготовкою до бакалаврських проєктів.

Робота над курсовим проєктом повинна бути завершена протягом шістнадцяти тижнів від дня видачі завдання.

Студент захищає проєкт у встановлені кафедрою терміни перед комісією, до складу якої входять викладачі кафедри. При захисті студент повинен продемонструвати необхідні знання, здатність орієнтуватися в теоретичному курсі «ЕЛЕКТРОНІКА ТА МІКРОПРОЦЕСОРНА ТЕХНІКА» і суміжних дисциплінах, вміння логічно мислити, повинен переконливо обґрунтовувати свою точку зору.

Захищена робота зберігається на кафедрі протягом двох років з моменту захисту.

1.2 Основна тематика курсового проєктування

Темою курсової проєкту може бути в залежності від завдання, отриманого від керівника на початку семестру, розробка одного з двох типових пристроїв:

- проєктування пристрою обміну інформацією між модулем УСАПП та мікроконтролером;
- проєктування пристрою керування двигуном постійного струму.